BAB 2
TINJAUAN PUSTAKA

2.1. Profil Tempat Kerja Praktek

Berikut ini adalah profil dari tempat kerja praktek

2.1.1. Sejarah Instansi

Universitas Padjadjaran berdiri pada hari Rabu 11 September 1957 dikukuhkan berdasarkan PP No. 37 Tahun 1957 tertanggal 18 September 1957 (LN RI No. 91 Tahun 1957). Adapun nama "Padjadjaran" diambil dari nama Kerajaan Sunda, yaitu Kerajaan Padjadjaran yang dipimpin oleh Prabu Siliwangi atau Prabu Dewantaprana Sri Baduga Maharaja di Pakuan Padjadjaran (1473-1513 M). Nama ini adalah nama yang paling terkenal dan dikenang oleh rakyat Jawa Barat, karena kemashuran sosoknya di antara raja-raja yang ada di tatar Sunda ketika itu.

Pada saat berdirinya, Unpad terdiri dari 4 fakultas: Fakultas Hukum dan Pengetahuan Masyarakat, Fakultas Ekonomi (keduanya berawal dari Yayasan Universitas Merdeka di Bandung), Fakultas Keguruan & Ilmu Pendidikan (FKIP, penjelmaan dari PTPG di Bandung), dan Fakultas Kedokteran.



Gambar 2. 1 Lambang Universitas Padjadjaran

Bentuk dan Arti Lambang Universitas Padjadjaran :

Bentuk dan arti lambang Universitas Padjadjaran ditetapkan berdasarkan SK Rektor Nomor: 903/H6.1/KEP/HK/2010 dan terdaftar di Kementerian Hukum dan Hak Asasi Manusia Nomor: 045978 tanggal 5 Februari 2010.

A. Lambang terdiri atas 6 bagian, yakni:

Obor ialah ilmu yang merupakan suluh penerangan kehidupan, yang membawa cahaya bahagia menuju keluhuran budi.

Kujang ialah senjata pusaka orang Sunda sebagai lambang kekuatan dan keberanian untuk melindungi hak dan kebenaran, yang menjadi tugas ilmu pengetahuan.

Sayap (terlukis daun paku) ialah sindiran terhadap kota Pakuan yang mengandung kata "paku" yang merupakan Pusat Kebesaran dari Keprabuan "Padjadjaran" dan daun paku ini merupakan sayap berarti senantiasa berusaha mencapai taraf atau mutu yang lebih tinggi.

Roda ialah kemasyarakatan, yang seperti halnya dengan roda senantiasa bergerak berputar dalam proses perubahan dan yang menjadi corak Universitas Padjadjaran.

Bunga Teratai yang sedang mekar berarti jaya abadi, suci sepanjang masa, lambang kebudayaan dan pendidikan.

Pentagon, perisai yang berbentuk segi lima ialah Pancasila, asas negara yang menjadi pedoman bagi segala usaha serta kegiatan universitas demi kepentingan negara dan bangsa.

B. Makna Warna Lambang :

Kuning berarti kejayaan dan keluhuran budi (dasar)

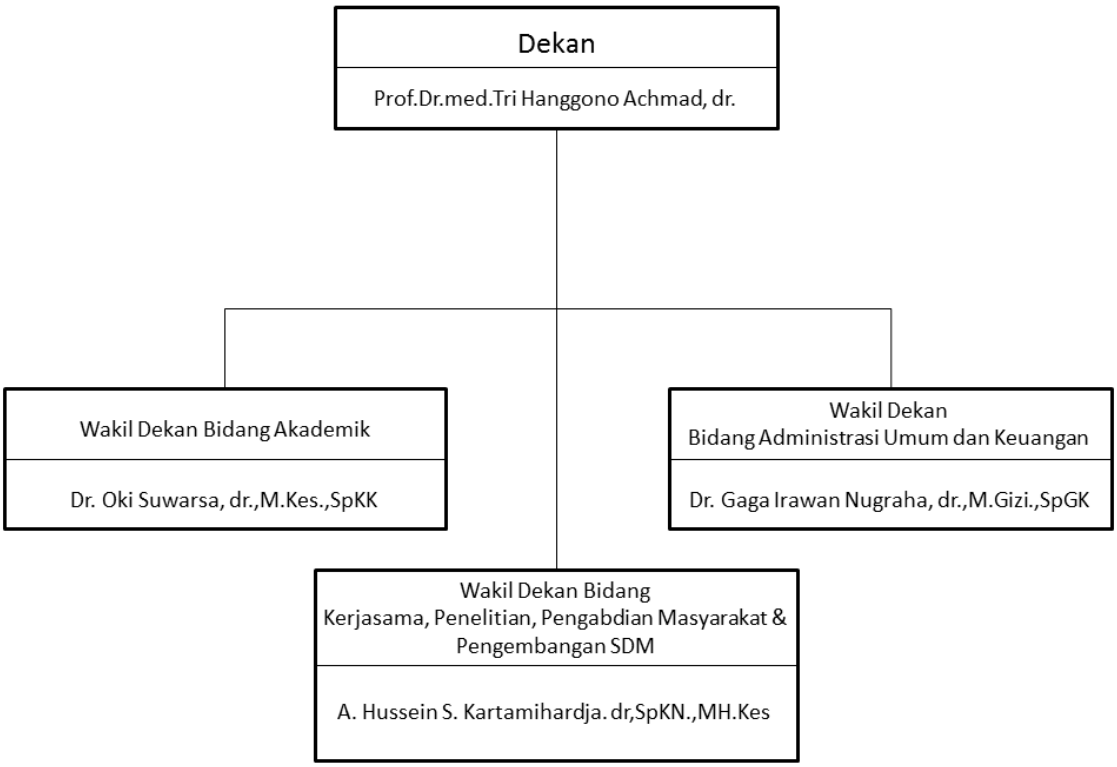
Merah berarti kebersihan (nyala obor dan roda).

Putih berarti kejujuran dan kesucian (kujang dan teratai).

Hitam berarti keteguhan, kekuatan, dan ketabahan hati (sayap).

2.1.2. Struktur Organisasi

Fakultas Kedokteran Universitas Padjadjaran memiliki struktur organisasi yang terbagi menjadi beberapa bagian, diantaranya sebagai berikut :



Gambar 2. 2 Struktur Organisasi Fakultas Kedokteran Universitas Padjadjaran

Fungsi dari setiap bagian yang ada dalam struktur organisasi Universitas Padjadjaran

1. Dekan

   Memimpin penyelenggaraan pendidikan, penelitian, dan pengabdian masyarakat, membinatenaga kependidikan, mahasiswa, dan tenaga administrasi fakultas serta memimpinpenyelenggaraan kegiatan di bidang keuangan umum dan administrasi.

2. Wakil Dekan Bidang Akademik

   Memimpin, mengoordinasikan, menyusun rencana, memberi tugas, mengarahkan, membina dan mengevaluasi pelaksanaan kegiatan bidang akademik serta merumuskan

kebijaksanaan dan memberi layanan dibidangnya berdasarkan ketentuan yang berlaku untuk kelancaran pelaksanaan tugas.

3. Wakil Dekan Bidang Administrasi Umum dan Keuangan

   Membantu Dekan dalam memimpin pelaksanaan kegiatan di bidang administrasi keuangan, perencanaan, kepegawaian/ketenagaan, administrasi umum dan perencanaan. Mengidentifikasi berbagai perencanaan strategi dan kebutuhan di bidang administrasi, keuangan, Sumber Daya Manusia, sarana-prasarana, dan pelayanan hukum.

4. Wakil Dekan Bidang Kerjasama, Penelitian, Pengabdian Kepada Masyarakat, dan Pengembangan Sumber Daya Manusia

   Melaksanakan penilaian prestasi dan proses penyelenggaraan kegiatan dan menyusun laporan bidang akademik. yang mempunyai tugas membantu Rektor dalam memimpin pelaksanaan pendidikan dan pengajaran, penelitian dan pengabdian kepada masyarakat. Mengatur dan mengelola semua sumber daya manusia dan sarana-prasarana agar efektif dan tepat sasaran dalam mencapai tujuan.

2.2. Landasan Teori

Berikut ini adalah landasan teori dari *game Autonomic Drugames*

2.2.1. Konsep Dasar *Game*

Berikut ini adalah penjelasan dari konsep dasar pada *game*

2.2.1.1. Pengertian *Game*

Game diambil dari bahasa Inggris yang diterjemahkan yang artinya permainan. Di dalam era perkembangan teknologi yang pesat ini, seperti halnya permainan kartu, catur dan lain-lainnya dapat ditemui melalui dunia virtual atau yang biasa kita mainkan di dalam komputer. Dengan perkembangan teknologi sekarang ini munculah berbagai sarana permainan, misalnya Playstation 3, Xbox 360, Nintendo Wii, PSP, Nintendo DS, maupun PC dari yang berbasis individu ataupun multiplayer.

Menurut Katie Salen dan Eric Zimmerman dalam buku Rules of play : Game Design Fundamental *"suatu sistem tempat pemain melakukan konflik bohongan, yang ditentukan oleh aturan, dan yang memberi hasil terukur"*[2]

Game itu sendiri mempunyai tujuan tidak hanya sekedar untuk kesenangan, *game* juga mempunyai tujuan yang lain yaitu:

1. *Game as game*

   Jenis *game* ini adalah jenis *game* yang paling umum ditemui. Tujuan utama dari *game* jenis ini adalah untuk kesenangan.

2. *Game as media*

   Jenis *game as media* punya tujuan utama yang berbeda dari jenis *game as game*, Tujuan utama dari *game as media* adalah untuk menyampaikan pesan tertentu, memanfaatkan *game* sebagai medium pesan. Isi dan cara penyampaian pesan ini tergantung pada keinginan desainer *game* dan target yang dikejar. Contoh umum dari *game as media* adalah *game* yang termasuk dalam *game* khusus, seperti *advergame*, *edugame*, dan *exergame*. Jenis *game* ini bisa diimplementasikan pada bidang-bidang yang lebih luas, seperti marketing, pendidikan, dan lain-lain.

3. *Game Beyond Game*

   Game beyond game, atau bisa disebut juga dengan istilah *gamification*. *Gamification* adalah penerapan konsep atau cara berpikir *game* design ke dalam lingkup non-*game*. Bentuk *gamification* sendiri pun biasanya tidak berupa *game*, misalnya sistem *reward* dari *point* yang dikumpulkan dengan berbelanja, atau lainnya. Sama seperti *game* khusus, *gamification* bisa diterapkan dalam beberapa bidang yang lebih umum seperti seni, pendidikan, jurnalisme, dan bisnis.

2.2.1.2. Karakteristik Game

Menurut Roger Caillois dalam bukunya Les jeux et les hommes (Games and Men)[3], game harus mempunyai karakteristik sebagai berikut:

1. *Fun*: sebuah aktivitas yang memberikan kesenangan dan tidak menyakitkan hati.
2. *Separate*: terpisah secara ruang dan waktu
3. *Uncertain*: hasil luaran dari aktivitas ini sulit untuk dilihat
4. *Non-produktive*: partisipasi tidak produktif
5. *Governed by rules*: memiliki aturan main yang berbeda dengan kehidupan nyata
6. *Fictitious*: diciptakan sebagai dimensi kehidupan yang berbeda dengan kenyataan.

2.2.1.3. Elemen Game

Didalam sebuah game terdapat beberapa elemen-elemen yang akan membentuk sebuah game itu sendiri yaitu :

a. *Rules* (aturan-aturan)

Sebuah *game* mengambil tempat atau seting di dunia buatan yang diatur oleh aturan-aturan (*rules*). *Rules* istilah yang menentukan aksi dan gerakan pemain dalam sebuah *game*. Pada *game* komputer, kebanyakan *rules* ini tersembunyi. Karena pemain berinteraksi dengan *game* hanya melalui suatu alat masukan atau yang biasanya disebut dengan *input device* dan mesin mengabaikan input yang tidak sesuai tanpa harus memberitahu aturan kepada pemain.

b. *Victory condition* (kondisi menang) dan *Lose condition* (kondisi kalah).

Karena ada kondisi kemenangan, maka *game* juga memiliki kondisi kekalahan (*lose condition*).

c. *Setting* (seting)

Sebuah *game* mengambil tempat pada suatu dunia atau setting tertentu. Contohnya pada permainan bola yang berseting pada lapangan dan batasan-batasan tertentu.

d. *Interaction model* (model interaksi)

Yaitu cara pemain berinteraksi dengan *game* dan melakukan aksi untuk menghadapi tantangan dari *game* tersebut. Model interaksi yang biasa dipakai pada *game* komputer ada dua jenis. Yang pertama, jika pemain mengendalikan tokoh tunggal yang mempresentasikan dirinya dalam *game* dan tokoh tersebut dapat mempengaruhi dunia sekitarnya, maka tokoh tersebut disebut sebagai *avatar*-nya. Sedangkan yang kedua adalah jika pemain memiliki kemampuan untuk melihat berbagai bagian dari *gameworld* (dunia permainan) dan melakukan aksi pada banyak tempat maka pemain tersebut sebagai *omnipresent*. Namun hal ini dapat diterapkan pada tokoh atau unit yang menjadi miliknya, dengan memberikan perintah kepada mereka satu-persatu. Salah satu contoh *game* yang termasuk jenis ini adalah catur.

e. *Perspective* (sudut pandang)

Yaitu menjelaskan bagaimana pemain melihat *gameworld* dari suatu *game* pada layar.

f. *Role* (peran)

Yaitu tokoh yang dimainkan oleh pemain dalam suatu *game*. Dengan adanya peran ini maka pemain akan lebih mudah untuk memahami apa yang sebenarnya ingin dicapai dan aturan apa yang yang dimainkan. Sebagai contoh, pada permainan monopoli kita berperan sebagai pialang perumahan. Pada *game* Championship Manager kita berperan sebagai manager sebuah tim sepak bola. Pada seri *game* Sierra Online's Police Quest pemain berperan sebagai polisi dan memiliki aturan-aturan seperti polisi sungguhan, seperti tidak boleh menembak sembarang sasaran, harus menaati peraturan tentang kapan diperbolehkannya menembak.

g. Mode

Beberapa *game*, seperti catur, berlaku sama dari awal sampai akhir. Pemain selalu ingin mencapai atau menyelesaikan hal yang sama dengan cara yang berbeda pula. Namun ada juga *game* yang memiliki mode yang nyata, yaitu dimana *gameplay*-nya berubah dari satu mode ke mode lainnya. Contohnya seperti pada *game* perang, yaitu sebelum pemain turun ke medan perang biasanya pemain diberi briefing terlebih dahulu, lalu pindah ke mode pemilihan senjata, dan terakhir adalah perang itu sendiri.

h. *Structure* (struktur)

Hubungan antara mode dan aturan menentukan kapan dan mengapa *game* berubah secara bersama-sama membentuk suatu permainan.

i. *Realism* (realisme)

Game menggambarkan sebuah dunia, bahkan mungkin dunia yang khayal. Sebuah *game* yang menerapkan akal sehat dan logika pada aturan permainannya dapat dikatakan sebagai *game* yang realistis. Contoh dari *game* ini adalah Microsoft Flight Simulator, *game* ini mencoba mengimplementasikan bagiamana mensimulasikan pesawat terbang yang sesungguhnya.

j. *Story* (cerita)

Game computer merupakan perpaduan antara media pasif, pasif seperti televisi dan film dan media aktif, dongeng, non pasif seperti permainan poker dan domino. Beberapa *game* komputer, seperti tetris tidaklah mempunyai cerita. Lain halnya dengan seri Metal Gear Solid, dan Final Fantasy. Beberapa *game* memiliki alur cerita yang linear atau hanya satu jalan cerita dan bersifat non interaktif, namun ada juga yang bersifat interaktif dimana cerita akan berbeda atau bercabang tergantung pada pilihan atau tindakan yang diambil pemain. Sehingga cerita akhir (ending) yang didapat akan berbeda-beda pula, atau biasanya disebut dengan *multiple ending*.

2.2.1.4. Genre Game

Perkembangan *game* setiap tahunnya semakin banyak jenisnya, teknologi yang digunakan juga semakin canggih. dari ratusan atau bahkan ribuan *game* yang telah dibuat, *game* tersebut dapat dikelompokkan menjadi beberapa genre. Pada dasarnya, genre *game* yang beredar di pasaran dikelompokkan sebagai berikut :

a. Action *Game*

Adalah tipe *game* dengan fitur utama berupa banyaknya aksi di mana pemain harus memiliki keterampilan reaksi yang cepat untuk menghindari musuh atau menghindari rintangan. Pengembang *game* tipe ini perlu memastikan *game* yang dibuat dioptimasi sehingga pemain memiliki pengalaman bermain yang baik, yang tidak terganggu oleh delay proses yang lama.

b. Adventure *Game*

Adalah tipe *game* yang umumnya membuat pemain harus berjalan mengelilingi suatu tempat yang terkondisi, seperti sebuah istana, gua yang berkelok, dan planet yang jauh. Pemain melakukan navigasi suatu area, mencari pesan-pesan rahasia, memperoleh obyek yang memiliki kemampuan yang bervariasi, bertempur dengan musuh, dan lain-lain. Untuk membuat *game* ini, diperlukan perencanaan yang akurat sehingga memiliki alur cerita yang menarik bagi pemain.

c. Sport *Game*

Adalah tipe *game* olahraga yang berupa kompetisi antara dua pemain atau lebih, di mana pemain dapat berupa individual atau tim. Contoh *game* tipe ini antara lain sepakbola, bola basket, tenis, dan bilyard.

d. Role Playing *Game* (RPG)

Adalah tipe *game* yang seringkali berupa multi-player *game* di mana setiap pemain memiliki karakter dengan kemampuan, kekuatan, dan kelemahan yang spesifik. Para pemain saling berkompetisi, berinteraksi, dan bertempur satu sama lain. Tampilan grafis yang khas untuk setiap karakter pemain ditambah dengan storyline yang mendebarkan akan sangat menarik dan memberikan pengalaman yang berbeda di dalam bermain.

e. Platform *Game*

Adalah tipe *game* yang mengharuskan pemain mengarahkan suatu obyek dengan melalui berbagai tahap atau tingkatan area untuk menyerang musuh dan menghindar terhadap serangan. Tipe *game* ini sedikit serupa dengan action *game*, tetapi aksinya tidak secepat action *game*. Teknik *collision detection* sangat sering dimanfaatkan pada tipe *game* ini.

f. Puzzle *Game*

Adalah tipe *game* yang umumnya membuat pemain menggunakan kemampuan berpikirnya sebagai pengganti keterampilan reaksi yang cepat karena terdapat rahasia yang perlu dipecahkan. *Game* ini lebih bersifat statis dibanding action *game*. Pembuatan *game* tipe ini seringkali ditunjang dengan algoritma *artificial intelligence*.

g. Sandbox *Game*

Adalah tipe *game* yang umumnya ditujukan untuk menjelajahi suatu kota atau tempat dan bebas berinteraksi dengan objek disekitarnya. Mungkin game ini bisa di bilang mirip RPG tapi Sandbox sangat berbeda dengan RPG game, perbedaannya adalah tidak adanya peningkatan level karakternya.

h. *First Person Shooter* (FPS)

Adalah tipe *game* yang mengunakan sudut pandang orang pertama untuk membidik atau membunuh musuh, sehingga kita hanya melihat tangannya saja dan tidak melihat tubuh karakter yang dimainkan.

i. *Third Person Shooter* (TPS)

Adalah tipe *game* yang mirip dengan FPS, tapi bedanya tipe *game* ini mengunakan sudut pandang orang ke-3, sehingga kita bisa melihat seluruh tubuh karakter yang dimainkan.

j. Musik *Game*

Adalah tipe *game* yang menuntut pemainnya untuk menekan tombol sesuai dengan tombol yang ada di layar dengan diiringi dengan musik.

k. *Fighting Game*

Adalah tipe *game* yang intinya harus menjatuhkan lawan tandingnya, entah itu dengan pukulan, tendangan, *combo*, maupun dengan jurus special.

l. Strategi *Game*

Adalah tipe *game* yang mengharuskan pemainnya menggunakan taktik dan strategi untuk jeli dalam melihat setiap peluang, kelemahan musuh dan bijaksana dalam menggunakan sumber daya yang ada.

m. *Simulation Game*

Adalah tipe *game* yang memberikan pengalaman atau interaksi 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.

n. *Racing Game*

Adalah tipe *game* yang tujuannya adalah mencapai garis finish dari suatu *race*, dalam *game* ini biasanya pemain dapat memilih & membeli kendaraan, mendandani, mengupgrade mesin dsb.

2.2.2. Konsep Dasar Edukasi

Berikut ini adalah konsep dasar dari edukasi

2.2.2.1. Pengertian Edukasi

Edukasi adalah proses yang dilakukan oleh seseoarang 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.

Menurut Craven dan Hirnle *"Edukasi adalah penambahan pengetahuan dan kemampuan seseorang melalui teknik praktik belajar atau instruksi, dengan tujuan untuk mengingat fakta atau kondisi nyata, dengan cara memberi dorongan terhadap pengarahan diri (self direction), aktif memberikan informasi-informasi atau ide baru"*.[4]

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 ilmiyah. Dalam belajar ada hubungan erat antara penguat, baik yang positif atau nigatif 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 seseoarang harus selalu bisa diukur dan selalu pulu diberikan penguat dan stimulasi yang diciptakan sedimikian rupa sesuai dengan tujuan yang hendak dicapai.

Edukasi dalam pendekatan pendidikan lebih kepada pemeberian 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 kesepakatan-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*.

2.2.3. Konsep Dasar *Game* Edukasi

Berikut ini adalah konsep dasar dari *game* edukasi

2.2.3.1. Pengertian *Game* Edukasi

Menurut A. Ritzhaupt, H. Higgins dan B. Allred *game* edukasi adalah *"Game digital yang dirancang untuk pengayaan pendidikan (mendukung pengajaran dan pembelajaran), menggunakan teknologi multimedia interatif"*.[5]

Game edukasi sangat menarik untuk dikembangkan. Ada beberapa kelebihan dari *game* edukasi dibandingkan dengan metode edukasi konvensional. Salah satu kelebihan utama *game* edukasi adalah pada visualisasi dari permasalahan nyata. Massachussets Insitute of Technology (MIT) berhasil membuktikan bahwa *game* sangat berguna untuk meningkatkan logika dan pemahaman pemain terhadap suatu masalah melalui proyek *game* yang dinamai Scratch.

Game edukasi didesain untuk mensimulasikan permasalahan yang ada sehingga diperoleh esensi atau ilmu yang dapat digunakan untuk menyelesaikan permasalahan tersebut. *Game* dengan tujuan edukasi ini dapat digunakan sebagai salah satu media edukasi yang memiliki pola pembelajaran *learning by doing*. Berdasarkan pola yang dimiliki oleh *game* tersebut, pemain dituntut untuk belajar sehingga dapat menyelesaikan permasalahan yang ada. Status *game*, instruksi, dan tools yang disediakan oleh *game* akan membimbing pemain secara aktif untuk menggali informasi sehingga dapat memperkaya pengetahuan dan strategi saat bermain. *Game* edukasi adalah salah satu bagian dari permainan yang serius.

Berdasarkan hasil penelitian, bahwa *game* edukasi dapat menunjang proses pendidikan (Marsh, dkk, 2005; Clark, 2006). *Game* edukasi unggul dalam beberapa aspek jika dibandingkan dengan metode pembelajaran konvensional. Salah satu keunggulan yang signifikan adalah adanya animasi yang dapat meningkatkan daya ingat sehingga anak dapat menyimpan materi pelajaran dalam waktu yang lebih lama dibandingkan dengan metode pengajaran konvensional (Clark, 2006).

Secara umum, *game* edukasi telah banyak dikembangkan namun tidak banyak *game* edukasi untuk perguruan tinggi dan praktisi. *Game* edukasi yang dibuat biasanya ditujukan untuk anak-anak. Ada beberapa *game* pada segmen perguruan tinggi dan praktisi, misalnya bussiness *game*, namun kebanyakan *game* pada segmen tersebut mengangkat permasalahan untuk tingkat strategis. Tidak banyak ditemui *game* edukasi yang mengangkat permasalahan ditingkat operasional.

2.2.3.2. Kriteria Game Edukasi

Menurut R. Peterson, I. Verenikina, dan J. Herrington [6] kriteria *game* edukasi yang ideal dibagi menjadi enam bagian yaitu :

1. Rasa ingin tahu, fantasi dan kontrol

   Motivasi intrinsik, menyenangkan untuk digunakan. Pengguna memiliki kontrol terhadap permainan.
2. Tantangan

   Tantangan diberikan secara terus menerus tapi disesuaikan dengan tingkat pemahaman pemain.
3. Sosialiasi

   Pemain dapat berkolaborasi bersama dan saling berkompetisi.
4. *Pedagogi*

Menyebutkan secara eksplisit target umur dan menyesuaikan rancangan permainan sesuai umur tersebut.

5. Teknologi

   Menggunakan teknologi yang tersedia di sekolah dan masyarakat umum (tidak membutuhkan hardware yang terlalu tinggi)

6. Pengguna anak-anak dan yang berkebutuhan khusus. Memberikan petunjuk yang dibacakan selain petunjuk tertulis.

2.2.4. Konsep Dasar Analisis Sistem

Berikut ini adalah konsep dasar dari analisis system

2.2.4.1. OOP (Object Oriented Programming)

OOP (Object Oriented Programming) atau yang dikenal dengan Pemrograman Berorientasi Objek merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus ke dalam kelas-kelas atau objek-objek. Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh anggap kita memiliki sebuah departemen yang memiliki manager, sekretaris, petugas administrasi data dan lainnya. Misal manager tersebut ingin memperoleh data dari bag administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bagian administrasi untuk mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas adminiistrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri. Pemrograman orientasi-objek menekankan konsep berikut:[7]

a. Kelas (*Class*)

Kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh 'class of dog' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi objek. Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya. [7]

b. Objek (*Object*)

Membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer. Objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek. [7]

c. Abstraksi (*Abstract*)

Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan. [7]

d. Enkapsulasi (*Encapsulation*)

Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi ijin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut. [7]

e. Polimorfisme (*Polymorphism*)

Melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan. Metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tunggal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama. [7]

f. Inheritas (*Inheritance*)

Mengatur polimorfisme dan enkapsulasi dengan mengijinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada objek-objek ini dapat membagi (dan memperluas) perilaku mereka tanpa harus mengimplementasi ulang perilaku tersebut (bahasa berbasis-objek tidak selalu memiliki inheritas). [7]

2.2.4.2. UML (Unified Modelling Language)

UML (Unified Modeling Language) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem. Unified Modeling Language (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP.

UML merupakan dasar bagi perangkat (tool) desain berorientasi objek dari IBM. UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. 25 UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan

yang umum dalam industri peranti lunak dan pengembangan system. UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu : [8]

1. *Use Case Diagram*

Use-case diagram adalah gambaran graphical dari beberapa atau semua actor, use-case, dan interaksi diantara komponen-komponen tersebut yang memperkenalkan suatu sistem yang akan dibangun. Use-case diagram menjelaskan manfaat suatu sistem jika dilihat menurut pandangan orang yang berada di luar sistem. Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar.

Use-case diagram dapat digunakan selama proses analisis untuk menangkap requirement sistem dan untuk memahami bagaimana sistem seharusnya bekerja. Selama tahap desain, use-case diagram berperan untuk menetapkan perilaku (behavior) sistem saat diimplementasikan. Dalam sebuah model mungkin terdapat satu atau beberapa use-case diagram. Kebutuhan atau requirements system adalah fungsionalitas apa yang harus disediakan oleh sistem kemudian didokumentasikan pada model use-case yang menggambarkan fungsi sistem yang diharapkan (use-case), dan yang mengelilinginya (actor), serta hubungan antara actor dengan use-case (use-case diagram) itu sendiri. [8]

2. *Conceptual Diagram*

Sebuah diagram konseptual merupakan representasi visual dari cara di mana konsep-konsep abstrak terkait. Hal ini digunakan sebagai bantuan dalam memvisualisasikan proses atau sistem tingkat tinggi melalui serangkaian garis yang unik dan bagan. Diagram konseptual secara luas digunakan dalam segala bidang seperti bisnis, ilmu pengetahuan, dan manufaktur. [8]

3. *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. [8]

4. *Collaboration Diagram*

Collaboration diagram yaitu diagram yang mengelompokkan pesan pada kumpulan diagram sekuen menjadi sebuah diagram. Dalam diagram tersebut terdapat method yang dijalankan antara objek yang satu dan objek lainnya. Di diagram kolaborasi ini, objek harus melakukan sinkronisasi pesan dengan serangkaian pesan-pesan lainnya. Collaboration Diagram lebih menekankan kepada peran setiap objek dan bukan pada waktu penyampaian pesan. [8]

5. *State Diagram*

Diagram State adalah diagram untuk menggambarkan *behavior*, yaitu perubahan state di suatu class berdasarkan *event* dan pesan yang dikirimkan dan diterima oleh class tersebut. Setiap diagram state hanya boleh memiliki satu *start state (initial state)* dan boleh memiliki satu atau lebih dari satu *stop states (final state).* [8]

6. *Activity Diagram*

Activity diagram memiliki pengertian yaitu lebih fokus kepada menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dipakai pada business modeling untuk memperlihatkan urutan aktifitas proses bisnis. Memiliki struktur diagram yang mirip flowchart atau data flow diagram pada perancangan terstruktur. Memiliki pula manfaat yaitu apabila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan. Dan activity dibuat berdasarkan sebuah atau beberapa use case pada use case diagram. [8]

7. *Class Diagram*

Class diagram adalah sebuah class yang menggambarkan struktur dan penjelasan class, paket, dan objek serta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Class diagram juga menjelaskan hubungan antar class dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan. [8]

8. *Object Diagram*

Objek diagram adalah diagram yang memberikan gambaran struktur model sebuah sistem, dalam kurun waktu tertentu. Diagram objek yang berasal dari diagram

kelas sehingga diagram objek tergantung pada diagram kelas. Obyek Diagram, kadang-kadang disebut sebagai diagram Instance sangat mirip dengan diagram kelas. Seperti diagram kelas object diagram juga menunjukkan hubungan antara obyek, tetapi object diagram menggunakan contoh-contoh dunia nyata. Object diagram digunakan untuk menunjukkan bagaimana sistem akan terlihat seperti pada waktu tertentu. Karena ada data yang tersedia di objek object diagram sering digunakan untuk menjelaskan hubungan yang kompleks antara objek. [8]

9. *Component Diagram*

Component diagram adalah diagram UML yang menampilkan komponen dalam system dan hubungan antara mereka. Pada component View, akan difokuskan pada organisasi fisik system. Pertama, diputuskan bagaimana kelas-kelas akan diorganisasikan menjadi kode pustaka. Kemudia akan dilihat bagaimana perbedaan antara berkas eksekusi, berkas dynamic link library (DDL), dan berkas runtime lainnya dalam system. [8]

10. *Deployment Diagram*

Deployment Diagram adalah diagram yang menggambarkan detail bagaimana komponen di-sebar (di-deploy) kedalam infrastruktur sistem, dimana komponen akan terletak (pada mesin, node, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal. [8]

2.2.5. Perangkat Lunak Yang Digunakan

Berikut ini adalah perangkat lunak yang digunakan dalam pembuatan *game Autonomic Drugames*

2.2.5.1. HTML5 (Hyper Text Markup Language)

HTML5 adalah Generasi selanjutnya dari HTML sebelumnya. HTML5 diciptakan dalam pengembangan bahasa HTML untuk memperbaiki konten dan memperbarui teknologi multimedia yang sudah ada pada HTML 4 dan versi sebelumnya agar mudah dijalankan oleh browser dan mudah dimengerti. HTML5 merupakan kerjasama antara *World Wide Web Consortium* (W3C) dan *Web Hypertext Application Teknologi Working Group* (WHATWG). WHATWG bekerja dengan bentuk web dan aplikasi, dan W3C bekerja dengan XHTML 2.0.

Pada tahun 2006, mereka memutuskan untuk bekerja sama dan membuat versi baru dari HTML yaitu HTML5.[9]

2.2.5.2. Construct 2

Construct 2 adalah editor *game* berbasis HTML5 dikembangkan oleh *Scirra Limited*. Construct 2 ditujukan terutama untuk non-programmer untuk menciptakan game dengan *drag-and-drop* memanfaatkan editor visual. Construct2 adalah penerus Construct Classic. Construct 2 juga memiliki kerangka javascript untuk programmer yang digunakan bersama editor *drag-and-drop*.

2.2.5.3. Microsoft Visual Studio 2012

Microsoft Visual Studio merupakan sebuah perangkat lunak yang dapat digunakan untuk melakukan pengembangunan aplikasi, baik itu aplikasi bisnis atau aplikasi personal dalam bentuk aplikasi *console*, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.