

BAB II

LANDASAN TEORI

II.1. Konsep Dasar Game

II.1.1. Pengertian Game

Game berasal dari bahasa Inggris yang memiliki arti dasar permainan.

Permainan dalam hal ini merujuk pada (*Intellectual Playability*). *Game* juga bisa diartikan sebagai arena keputusan dan aksi permainannya. Permainan juga dapat disebut sesuatu yang dimainkan dengan aturan tertentu sehingga ada yang menang dan ada yang kalah, biasanya dalam konteks tidak serius atau dengan tujuan *refreshing* (hiburan).

Menurut Ernest Adams *Fundamentals* dalam bukunya *Game Design*, “2.nd Edition” (2010, 2), *Game* adalah salah satu jenis aktifitas bermain, yang didalamnya dilakukan dalam konteks berpura-pura namun terlihat seperti realitas, yang mana pemainnya memiliki tujuan untuk mendapatkan satu kemenangan serta dilakukan dengan sesuai dengan aturan permainan yang dibuat.

Game berasal dari keinginan untuk bermain (*playing*) dan kemampuan untuk berpura-pura (*pretending*) yang dimiliki manusia. Bermain merupakan aktifitas manusia yang termasuk golongan *nonessential* dan biasanya untuk tujuan hiburan yang berhubungan dengan kehidupan sosial. Berpura-pura adalah kemampuan mental untuk menciptakan sebuah realitas khayalan (*notional reality*) dimana orang yang berpura-pura tersebut dan menciptakan, mengubah, atau meninggalkan realitas tersebut sesuai keinginannya.

II.1.2. Elemen dalam Game

Menurut Samuel Henry (2005), elemen-elemen *game* adalah sangat penting, bahkan tanpa elemen-elemen ini maka nilai dari *game* kita bisa berkurang. Bahkan untuk skala komersil yang besar. Elemen –elemen *game* ini mutlak harus ada untuk *game* skala kecil atau maupun freeware maka elemen ini terkadang tidak semuanya ada di dalam sebuah *game*.

Elemen-elemen penting dalam sebuah *game* adalah bermain (*play*), berpura-pura (*pretend*), tujuan (*goal*), dan peraturan (*rules*) (Ernest Adams, 2010:6-9). Adapun definisi dari elemen-elemen tersebut adalah sebagai berikut:

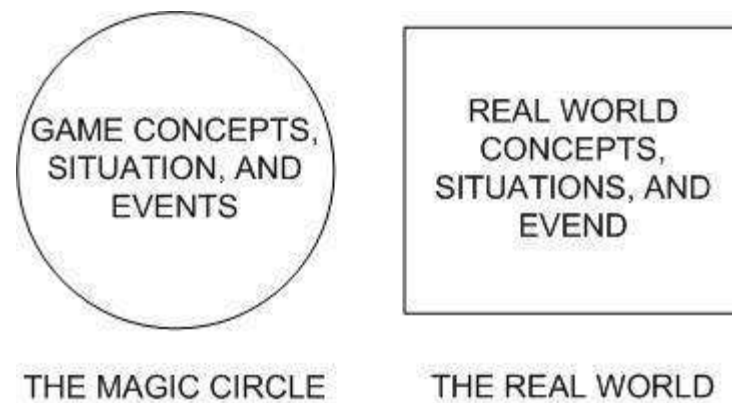
1. Bermain (*Play*)

Bermain adalah salah satu bentuk hiburan yang membutuhkan keikutsertaan (komunikasi dua arah), dimana buku, *film*, dan teater adalah bentuk hiburan yang berupa pertunjukan (komunikasi dua arah). Ketika membaca buku, penulis yang menghibur pembaca, tetapi ketika bermain, pemainlah yang menghibur dirinya sendiri. Sebuah buku tidak dapat berubah meskipun sudah sering dibaca, namun ketika bermain, pemain dapat membuat keputusan yang mempengaruhi kejadian dalam permainan tersebut.

2. Berpura-pura (*Pretending*)

Berpura-pura adalah kegiatan untuk menciptakan sebuah realitas khayalan di dalam pikiran. Nama lain dari realitas yang diciptakan dari berpura-pura adalah *magic circle*. Istilah ini pertama kali digunakan oleh sejarawan Belanda bernama Johan Huizinga dalam *Homo Ludens* bukunya” (Huizinga, berjudul 1971” berhubungan dengan dunia imajinasi dalam sebuah drama atau novel fiksi. Dalam

game, *magic circle* mengacu pada batas yang memisahkan ide dan aktifitas yang berarti yang berada dalam *game* dari ide dan aktifitas yang berarti dalam dunia nyata. Dengan kata lain batas antara dunia nyata dan khayalan (*make believe*).



Gambar II.1. Konsep The Magic Circle dalam sebuah game (Sumber: Ernest Adams; 2010)

3. Tujuan (*Goal*)

Sebuah *game* harus memiliki satu atau lebih tujuan. Tujuan dari *game* tersebut oleh peraturan dan bentuk *game* itu sendiri karena pembuat *game* dapat menentukannya sesuai dengan keinginannya. Dalam mencapai tujuan dalam *game*, harus ada tantangan. Meskipun kesulitan dari tantangan itu tergantung dari persepsi masing-masing pemain.

4. Peraturan (*Rules*)

Peraturan adalah definisi dan instruksi dimana pemain harus mengikutinya selama *game* berlangsung. Peraturan memiliki beberapa fungsi yaitu membuat objek *game* dan arti dari setiap aktifitas dan kejadian yang berbeda yang berlangsung di dalam *circle magic*. Peraturan juga membuat pemain mengetahui

aktifitas apa yang diperbolehkan dan juga mengevaluasi setiap tindakan yang akan membuat pemain mencapai tujuan dari *game* tersebut.

II.1.3. Jenis *Game*

1. Berdasarkan Jenis (*genre*) Permainannya

Jenis (*Genre*) *game* adalah format atau gaya dari sebuah *game*. Menurut Adams, Ernest (2010, 70-71), *game* memiliki jenis (*genre*) yang luas antara lain:

a. *Actions Games*

Action games meliputi tantangan fisik, teka-teki (*puzzle*), balapan, dan beberapa konflik lainnya. Dapat juga meliputi masalah ekonomi sederhana, seperti mengumpulkan benda-benda.

b. *Strategy Games*

Strategy Games melibatkan masalah strategi, taktik, dan logika.

Contoh *game* jenis ini adalah *Age of Empire*, *War Craft*, dan sebagainya.

c. *Role-playing Games*

Game jenis ini melibatkan masalah taktik, logika, dan eksplorasi atau penjelajahan. Dan juga kadang meliputi teka-teki dan masalah ekonomi karena pada *game* ini biasanya melibatkan pengumpulan barang-barang rampasan dan menjualnya untuk mendapatkan senjata yang lebih baik. Contoh dari *game* ini adalah *Final Fantasy*, *Ragnarok*, *Lord of The Rings*, dan sebagainya.

d. *Real-world Simulations*

Game ini meliputi permainan olahraga dan simulasi kendaraan termasuk kendaraan militer. *Games* ini kebanyakan melibatkan masalah fisik dan taktik,

tetapi tidak mengandung unsur eksplorasi, ekonomi dan konseptual. Contohnya seperti adalah *game Championship Manager*.

e. *Construction and management games*

Game ini seperti *game Roller Coster Tycoon* dan *The Sims*. Pada dasarnya adalah masalah ekonomi dan konseptual. *Game* ini jarang yang melibatkan konflik dan eksplorasi, dan hampir tidak pernah meliputi tantangan fisik.

f. *Adventure Games*

Game jenis ini mengutamakan masalah eksplorasi dan pemecahan teka-teki. Namun terkadang meliputi masalah konseptual, dan tantangan fisik meskipun sangat jarang.

g. *Puzzle Games*

Game ini ditujukan untuk memecahkan suatu masalah tertentu. Hampir semua tantangan disini menyangkut masalah logika yang biasanya dibatasi oleh waktu.

2. Berdasarkan Jenis Platform yang Digunakan

Game juga dapat dibedakan berdasarkan *platform* (Mesin *Game*) yang digunakan, menurut Adam, Ernest (2010 , 77-82), adapun jenis *platform* yang digunakan antara lain:

a. *Home Game Consoles*

b. *Personal Computers*

c. *Handheld Game Machines*

d. *Mobile Phones* dan *Wireless Devices*

e. Perangkat lainnya seperti *arcade*, mesin judi, dan alat simulasi pesawat.

II.1.4. Definisi Real-world Simulations

Real-world Simulations adalah *genre game* termasuk di dalamnya *Sport Games* (Permainan Olahraga) dan *Vehicle Simulations* (Simulasi Kendaraan), yang melibatkan tantangan fisik bersifat aksi dan taktis tetapi tidak mengandung unsur eksplorasi (penjelajahan), ekonomi, atau konseptual (Ernest Adams, 2010:71).

II.1.5. Definisi Sport Games

Permainan olahraga mensimulasikan beberapa aspek dari olahraga atletik nyata atau imajiner, apakah itu bermain di pertandingan, mengelola sebuah tim atau karir, atau keduanya. Permainan dalam pertandingan menggunakan tantangan fisik dan strategis (Ernest Adams, 2010:482).

Kebanyakan permainan olahraga berkonsentrasi pada simulasi pertandingan yang sebenarnya, tetapi banyak juga yang memasukkan sejumlah fungsi manajemen serta tantangan dalam mengelola tim atau karir seorang atlet. Beberapa permainan olahraga hanya menerapkan aspek olahraga dan tidak memungkinkan pemain untuk mengontrol atlet individu dalam pertandingan. Kadang-kadang disebut *Game Manager*, yang sangat populer di Eropa.



**Gambar II.2. Contoh tampilan grafis Sport Game “Balap Karung”
(Sumber: Denny Listyo;2013)Mu’adzin**

1. Struktur Permainan

Mode *gameplay* utama dalam permainan olahraga adalah pertandingan yang mensimulasikan olahraga itu sendiri seperti yang sebagaimana permainan aslinya. Pemain biasanya dapat menghentikan sebentar permainan (*pause*), yang biasanya membawa sebuah menu memungkinkan mereka untuk menggantikan atlet, mengubah tampilan kamera, melakukan jenis lain dari tugas pelatih, dan kadang-kadang menyesuaikan AI. Pemain juga dapat menyimpan permainan untuk nanti atau meninggalkannya.

2. *Gameplay and Rules* (Pola dan Aturan Permainan)

Tantangan dan tindakan dalam permainan olahraga adalah tindakan atlet dalam olahraga yang sebenarnya tetapi tindakan atlet diatur menggunakan perangkat pengendali dari mesin *game*. Apapun yang atlet mencoba untuk capai

dalam pertandingan adalah tujuan pemain. Aturan dalam permainan olahraga, sebagian besar menggunakan aturan olahraga yang sebenarnya.

II.2. Pengertian *Game Engine*

Game Engine adalah sistem perangkat lunak yang dirancang untuk menciptakan dan pengembangan *video game*. Ada banyak *game engine* yang dirancang untuk bekerja pada konsol permainan video dan sistem operasi *desktop* seperti Microsoft Windows, Linux, dan Mac OS X. fungsionalitas inti yang biasanya disediakan oleh *game engine* mencakup mesin render untuk merender grafis 2D atau 3D, mesin fisika atau tabrakan, suara, script, animasi, kecerdasan buatan, jaringan, streaming, manajemen memori, *threading*, dukungan lokalisasi, dan adegan grafik.

II.2.1. *Unity Engine*

Unity Game Engine merupakan *software* yang digunakan untuk membuat *video game* 3D atau konten yang interaktif lainnya seperti, visual arsitektur dan real-time 3D animasi. *Unity Game Engine* tidak hanya merupakan sebuah *game engine*, tapi juga merupakan sebuah *editor*. *Unity Game Engine* mirip dengan *game engine* lainnya seperti, *Director*, *Blender game engine*, *Virtools*, *Torque Game Builder* atau *Gamestudio*.

Kelebihan dari *Unity Game Engine* yaitu *multi platform*, *Unity Game Engine* dapat dioperasikan pada *platform* Windows dan Mac Os dan dapat menghasilkan *game* untuk Windows, Mac, Linux, Wii, iPad, iPhone, dan *platform*

Android. Unity Game Engine juga dapat menghasilkan *game browser*, untuk menjalankan pada *web browser* kita memerlukan sebuah *plugin* yaitu *Unity web player plugin*.

II.2.2. Komponen-komponen *Unity Engine*

1. *Scene*

Sebuah *game* yang dibuat menggunakan *Unity Engine* terdiri dari satu atau lebih *scene*. Dalam *game engine* lain, *scene* bisa juga disebut level (Philip Chu, 2009, 5).

2. *Game Object*

Sebuah *scene* terdiri dari *game object* yang juga disebut entitas (entities) dalam *game engine* lain. Sebuah *game object* memiliki nama, posisi, dan orientasi di dalam *scene* dan *attributes* atau *behaviour* tergantung tipe dari objek yang direpresentasikannya.

Game object juga dapat memiliki hubungan *parent-child* dengan objek lainnya. Posisi dan orientasi dari masing-masing *game object* berhubungan dengan *parent*-nya. Hal ini sering disebut sebagai *scene graph* dalam dunia grafis 3D (Philip Chu, 2009, 6).

3. *Classes*

Unity Engine berorientasi objek, artinya setiap objek memiliki sebuah *class*, dan masing-masing *class* dapat berasal dari *base class* atau *parent class*. Sebagai sistem berorientasi objek, *Unity* memiliki sebuah hierarki *class*.

Game object berasal dari *object class*, begitu juga dengan *class* lainnya.



**Gambar II.3. Hierarki class dari sebuah objek
(Sumber: Philip Chu; 2009:6)**

4. Components

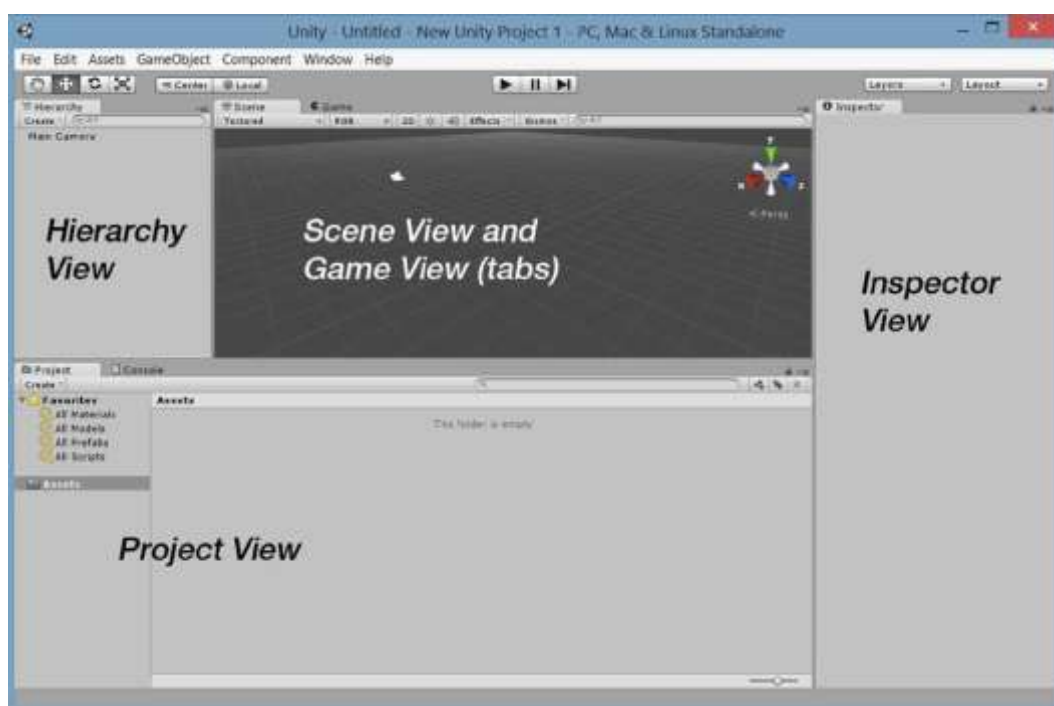
Dalam *Unity Engine*, setiap node dari *scene graph* merupakan satu *class*, *game object*, dan untuk menentukan *behaviour*-nya, *components* harus ditambahkan sesuai dengan *behaviour* yang diinginkan.



**Gambar II.4. Hierarki class dari sebuah component yang berisi
behaviour (Sumber: Philip Chu; 2009: 8)**

II.2.3. Editor Unity Engine

Game dibuat sebagai *project* dalam *Unity Engine*. Editor menampilkan satu *scene Unity* dalam satu waktu. Editor *Unity* terdiri dari 4 tampilan (*view*) utama, yaitu *Scene View*, *Game View*, *Hierarchy View*, dan *Inspector View* (biasanya disebut sebagai *Inspector*) dan beberapa fitur penting seperti *Playback Button*, *Layer Button*, dan lain-lain.



**Gambar II.5. Tampilan GUI dari Editor Unity
(Sumber: Sue Blackman; 2013:7)**

II.2.4. Scripting

Scripting (penulisan kode program) dalam *Unity Engine* menggunakan *Mono*, sebuah versi *open-source* dari *.NET*. *Mono* mendukung banyak bahasa pemrograman, tetapi *Unity* hanya mendukung bahasa *C#*, *Boo*, dan *Javascript*.

II.2.5. Lingkungan Bahasa Pemrograman *Microsoft Visual C#*

Microsoft Visual C#, yang selanjutnya disebut sebagai C# merupakan bagian dari bahasa keluarga Microsoft yang berjalan pada *framework* .NET. C# bebas dari masalah kompatibilitas dan dilengkapi dengan berbagai fitur baru, menarik, dan tentu saja menjanjikan (Adi Nugroho, 2010, 1).

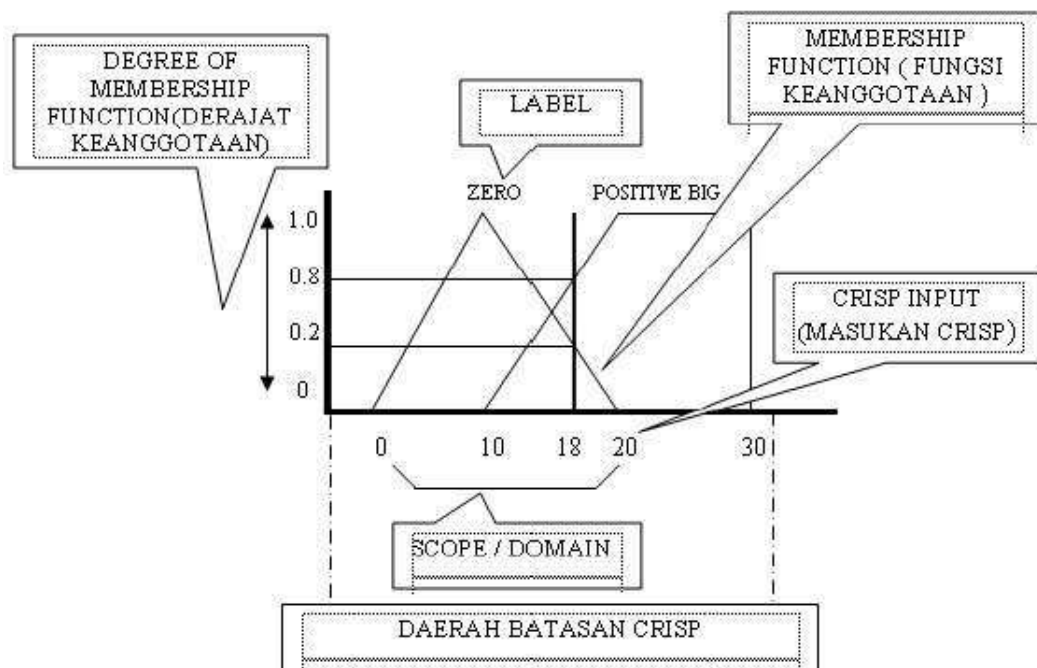
C# merupakan bahasa pemrograman yang berorientasi objek dan memiliki banyak kesamaan dengan bahasa C++, *Java*, dan *Visual Basic* (VB). C# merupakan kombinasi antara efisiensi pemrograman C++, kesederhanaan *Java*, dan penyederhanaan dari bahasa VB.

II.3. Pengertian *Fuzzy Logic*

Profesor Lotfi A. Zadeh adalah guru besar pada *University of California* yang merupakan pencetus sekaligus yang memasarkan ide tentang cara mekanisme pengolahan atau manajemen ketidakpastian yang kemudian dikenal dengan logika *fuzzy*. Dalam penyajiannya variabel-variabel yang akan digunakan harus cukup menggambarkan ke-*fuzzy*-an tetapi di lain pihak persamaan-persamaan yang dihasilkan dari variabel-variabel itu haruslah cukup sederhana sehingga komputasinya menjadi cukup mudah. Karena itu Profesor Lotfi A Zadeh kemudian memperoleh ide untuk menyajikannya keanggotaan”*membershipfunction*() dari masing-masing variabelnya (Sudradjat, 2008: 27).

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik input data kedalam nilai keanggotaanya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1.

1. Derajat Keanggotaan (*membership function*) adalah : derajat dimana nilai *crisp* dengan fungsi keanggotaan (dari 0 sampai 1), juga mengacu sebagai tingkat keanggotaan, nilai kebenaran, atau masukan *fuzzy*.
2. Label adalah nama deskriptif yang digunakan untuk mengidentifikasi sebuah fungsi keanggotaan.
3. Fungsi Keanggotaan adalah mendefinisikan *fuzzy set* dengan memetakan masukan *crisp* dari domainnya ke derajat keanggotaan.



Gambar II.6 Konsep dasar logika *fuzzy*

(Sumber: Sudradjat, 2008:28)

4. Masukan *Crisp* adalah masukan yang tegas dan tertentu.
5. Lingkup/*Domain* adalah lebar fungsi keanggotaan. Jangkauan konsep, biasanya bilangan, tempat dimana fungsi keanggotaan dipetakan.
6. Daerah Batasan *Crisp* adalah jangkauan seluruh nilai yang dapat diaplikasikan pada variabel sistem.

Logika *Fuzzy* adalah AI yang digunakan untuk menangani ketidakpastian masalah-masalah yang memiliki banyak jawaban. Untuk mempresentasikan masalah yang mengandung ketidakpastian ke dalam suatu bahasa formal yang dipahami komputer. Pada dasarnya sudah terbawa sejak lahir, seperti pertanyaan tentang “seberapa secepatbesarapa?dan?,sebagainyaseberapa”. keci

Logika *Fuzzy Logic* dapat dikatakan sebagai logika baru yang lama, sebab ilmu tentang logika modern dan metodel baru ditemukan pada tahun 1965, padahal sebenarnya konsep tentang logika *fuzzy* itu sendiri sudah ada sejak lama.

Beberapa alasan digunakannya logika *fuzzy* :

1. Konsep logika *fuzzy* mudah dimengerti.
2. Logika *fuzzy* sangat fleksibel.
3. Logika *fuzzy* memiliki toleansi terhadap data yang kurang tepat.
4. Logika *fuzzy* mampu memodelkan fungsi nonlinier yang kompleks.
5. Logika *fuzzy* didasari pada bahasa alami.

II.4. Pengertian Multimedia

Multimedia dapat diartikan sebagai penggunaan beberapa media yang berbeda untuk menggabungkan dan menyampaikan informasi dalam bentuk text, audio, grafik, animasi, dan video.

Beberapa definisi menurut beberapa ahli:

1. Kombinasi dari komputer dan video (Rosch, 1996)
2. Kombinasi dari tiga elemen: suara, gambar dan teks (McComick, 1996)
3. Kombinasi dari paling sedikit dua media input atau output. Media ini dapat berupa audio (suara, musik), animasi, video, teks, grafik dan gambar (Turban dan kawan-kawan, 2002)
4. Alat yang dapat menciptakan presentasi yang dinamis dan interaktif yang mengkombinasikan teks, grafik, animasi, audio dan video (Robin dan Linda, 2001).