

BAB II LANDASAN TEORI

2.1 Game

Game merupakan suatu jenis model permainan atau pertandingan. *Game* dapat diartikan sebagai aktivitas terstruktur atau semi terstruktur, yang biasanya dilakukan untuk hiburan dan sebagai alat pembelajaran. Kata *Game* berasal dari bahasa Inggris yang artinya adalah permainan. Permainan dalam hal ini merujuk pada pengertian kelincahan intelektual (*Intellectual Playability Game*) yang juga bisa diartikan sebagai keputusan dalam aksi pemainnya untuk mencapai tujuan [8]. Dengan kata lain, *game* merupakan salah satu media yang dapat digunakan dalam menyampaikan suatu tujuan, dapat berupa hiburan ataupun pendidikan.

2.1.1 Jenis-Jenis Permainan

Game dikategorikan dalam beberapa jenis (*genre*) diantaranya [9]:

a. *Action Game*

Game yang mengutamakan gerak. Permainan jenis ini membutuhkan ketangkasan/ respon yang cepat dari pemain. Permainan aksi (*Action games*) mengajak pemain untuk menggunakan refleks, akurasi serta waktu yang tepat untuk menyelesaikan sebuah tantangan. Merupakan genre dasar dari sebuah permainan dan digunakan di hampir semua permainan yang ada. Dalam permainan aksi, biasanya terdapat pertempuran. Terdapat banyak sub-genre dari permainan aksi, seperti *Fighting games* dan *First-person shooter*.

b. *Strategy Game*

Asal-usul genre ini berasal dari *board games* biasanya memiliki banyak aturan yang merangsang strategi dan perencanaan. *Real-time strategy* (RTS), yaitu genre permainan video yang mengandung unsur strategi. Permainan video RTS memerlukan kecakapan pemain untuk memimpin suatu kelompok, mengelola sumber daya, dan melancarkan serangan untuk memperluas wilayah kekuasaan.

Contoh permainan video jenis ini adalah *serial Age of Empires, Civilization, Rise of Nations, dan Warcraft. Turn-based strategy (TBS)*, yaitu jenis permainan video strategi yang pemainnya saling menunggu giliran untuk bergerak.

c. *Role-Playing*

Game Permainan ini biasanya memiliki ciri khas dengan memiliki alur cerita yang kuat dan meningkatkan pengalaman pemain. Permainan peran (*Role-playing games*), merupakan permainan video yang menempatkan pemain sebagai tokoh dalam permainan tersebut untuk memecahkan suatu misteri dengan menyelesaikan berbagai macam *puzzle* dan *quest*. Jenis ini biasanya memiliki alur cerita yang kompleks. Contoh permainan video dalam ragam ini adalah serial *Final Fantasy, Star Ocean, dan Suikoden*.

d. *Sport Game*

Pertandingan olahraga seperti di dunia nyata yang dibawa ke dalam sebuah *game*. Permainan olahraga (*Sports games*), yaitu jenis permainan video yang menuntut keterampilan pemain untuk melakukan pertandingan olahraga secara virtual, seperti pertandingan sepak bola, basket, dan sebagainya. Contoh permainan video dalam ragam ini adalah serial *Pro Evolution Soccer* dan *Madden NFL*.

e. *Vehicle Simulation Game*

Game ini mencoba menciptakan bagaimana rasa mengemudi atau menerbangkan suatu kendaraan sebagai media simulasi. *Game* simulasi kendaraan adalah genre *video game* yang mencoba untuk memberikan pengalaman pemain dengan interpretasi realistis operasi berbagai jenis kendaraan. Kendaraan yang dimaksud termasuk mobil, pesawat, perahu, pesawat ruang angkasa, kendaraan militer dan berbagai kendaraan lain. Tantangan utama game ini adalah untuk menguasai dan mengemudi kendaraan dari perspektif pilot atau *driver*. Kebanyakan *game* menambahkan tantangan lain seperti balap atau kendaraan tempur saingan.

Game sering ditambahkan pula efek *real*, dengan beberapa *game* termasuk fisika dan tantangan seperti manajemen bahan bakar yang lebih realistis.

f. *Construction and Management Simulation Game*

Jenis *game* ini bertujuan untuk membuat sesuatu dalam konteks proses yang berkelanjutan. Semakin baik pula hasil yang diperoleh. Konstruksi dan simulasi manajemen (CMS) adalah jenis simulasi permainan di mana pemain membangun, memperluas atau mengelola komunitas fiksi atau proyek dengan sumber daya yang terbatas. Strategi *video game* kadang-kadang menggabungkan aspek CMS ke dalam perekonomian. Pemain harus mengelola sumber daya sekaligus memperluas proyek mereka. Permainan CMS murni berbeda dari *game* strategi. Tujuan pemain adalah tidak untuk mengalahkan musuh, tetapi untuk membangun sesuatu dalam konteks yang berkelanjutan. Permainan dalam kategori ini kadang-kadang juga disebut "permainan manajemen". SimCity merupakan contoh awal dari keberhasilan besar genre *game* ini.

g. *Adventure Game*

Game petualangan dengan sebuah cerita interaktif tentang karakter yang dikendalikan oleh pemain. Permainan petualangan aksi (*Action-adventure games*), yaitu jenis permainan video yang menempatkan pemain sebagai tokoh dalam permainan tersebut, mirip dengan permainan peran, tetapi ditambah dengan unsur-unsur aksi, misalnya perkelahian dan tembak-menembak. Contoh permainan video dalam ragam ini adalah serial *Prince of Persia* dan *Devil May Cry*.

h. *Artificial Life*

Jenis ini melibatkan proses pemodelan biologis dan seringkali untuk mensimulasikan siklus kehidupan makhluk hidup. *Game* simulasi kehidupan sekitar, memelihara dan menumbuhkan populasi. Pemain diberikan kekuatan untuk mengontrol kehidupan makhluk otonom atau orang-orang. *Artificial Life* berhubungan dengan penelitian ilmu komputer di kehidupan buatan.

i. *Puzzle Game*

Permainan mengenai pemecahan teka-teki. Permainan jenis ini menarik secara visual dan menyenangkan untuk dimainkan.

j. *Education Game*

Permainan yang mengandung konten pembelajaran terhadap suatu bidang tertentu.

2.1.2 Unsur-Unsur Permainan

Game terdiri dari berbagai unsur dasar yang dipadukan sehingga menciptakan suatu pengalaman yang menarik, adapun komponen-komponen *game* antara lain [9]:

1. *Fitur*

Fitur merupakan suatu hal yang menggambarkan *game* kedalam bentuk yang dapat dilihat maupun dirasakan.

2. *Gameplay*

Gameplay merupakan cara kerja suatu *game*, dimana fitur-fitur yang ada akan membentuk suatu *gameplay*.

3. *Interface*

Interface merupakan semesta bentuk tampilan yang ada pada sebuah *game*. *Interface* yang baik akan membuat pemainnya betah dan tidak bosan dalam memainkan *game*.

4. *Rules* (Aturan)

Rules merupakan sekumpulan aturan yang ada dalam memainkan sebuah *game*.

5. *Desain level*

Desain level merupakan tingkat kesulitan yang menggaambarkan jalan cerita pada sebuah *game*.

2.2 Game Edukasi

Game edukasi merupakan salah satu dari jenis *game* digital yang dirancang untuk pengayaan pendidikan (mendukung pengajaran dan pembelajaran) menggunakan teknologi multimedia interaktif.

Menurut Hurd dan Jenuings, perancang yang haruslah memenuhi kriteria dari *game* edukasi, yaitu [2]:

1. Nilai Keseluruhan (*Overall Value*)

Nilai keseluruhan dari suatu *game* terpusat pada desain dan panjang durasi *game*. Aplikasi ini dibangun dengan desain yang menarik dan interaktif. Untuk penentuan panjang durasi.

2. Dapat Digunakan (*Usability*)

Mudah digunakan dan diakses adalah point penting bagi pembuat *game*. Aplikasi ini merancang system dengan *interface yang user friendly* sehingga user dengan mudah dapat mengakses aplikasi.

3. Keakuratan (*Accuracy*)

Keakuratan diartikan sebagai bagaimana kesuksesan model/gambaran sebuah *game* dapat dituangkan ke dalam percobaan atau perancangannya. Perancangan aplikasi ini harus sesuai dengan model *game* pada tahap perancangan.

4. Kesesuaian (*Appropriateness*)

Kesesuaian dapat diartikan bagaimana isi dan desain *game* dapat diadaptasikan terhadap keperluan user dengan baik. Aplikasi ini menyediakan menu dan fitur yang diperlukan *user* untuk membantu pemahaman *user* dalam menggunakan aplikasi.

5. Relevan (*Relevance*)

Relevan artinya dapat mengaplikasikan isi *game* ke target *user*. Agar dapat relevan terhadap *user*, sitem harus membimbing mereka dalam pencapaian tujuan pembelajaran. Karena aplikasi ini ditunjukkan untuk anak-anak, maka desain antarmuka harus sesuai dengan nuansa anak-anak, yaitu menampilkan warna-warna yang ceria.

6. Objektivitas (*Objectives*)

Objektivitas menentukan tujuan *user* dan kriteria dari kesuksesan atau kegagalan. Dalam aplikasi ini objektivitas adalah usaha untuk mempelajari hasil dari permainan.

7. Umpan Balik (*Feedback*)

Untuk membantu pemahaman user bahwa permainan (*performance*) mereka sesuai dengan objek *game* atau tidak, *feedback* harus disediakan. Aplikasi ini menyajikan animasi dan efek suara yang mengindikasikan kesuksesan atau kegagalan pemain.

2.3 Algoritma A-Star (A*)

Algoritma a-star merupakan algoritma yang digunakan untuk pencarian langkah seperti pada berbagai macam *game puzzle* serta *pathfinding* yang merupakan pencarian jalan suatu simbol untuk sampai ke tujuan. Algoritma a-star dapat menyelesaikan permasalahan secara optimal dengan membangkitkan simpul paling mendekati nilai optimum untuk menemukan jalan terpendek menuju tujuan. [5].

Menurut Anita Desiani, algoritma a-star merupakan algoritma pengembangan dari *best first search* dengan memodifikasi fungsi heuristik [1]. Perbedaan algoritma a-star dengan *best first search* terletak pada notasi standar fungsi heuristik yang digunakan oleh kedua algoritma, *best first search* hanya menggunakan biaya perkiraan ($h(n)$) tetapi algoritma a-star selain menggunakan biaya perkiraan ditambah biaya yang diperlukan atau biaya sebenarnya ($g(n)$). Dengan kata lain, modifikasi fungsi heuristik inilah yang menjadikan algoritma a-star lebih baik dari pada algoritma yang lainnya dalam menghasilkan solusi yang optimal [5]. Selain itu dengan menerapkan suatu heuristik, algoritma ini membuang langkah-langkah yang tidak perlu dengan pertimbangan bahwa langkah-langkah yang dibuang sudah pasti merupakan langkah yang tidak akan mencapai solusi yang diinginkan.

Secara sistematis, fungsi evaluasi dari suatu node n dituliskan sebagai berikut: $f(n) = g(n) + h(n)$

yang berarti :

$f(n)$ = fungsi evaluasi $g(n)$ = biaya (cost) yang sudah dikeluarkan dari node asal hingga node n $h(n)$ = estimasi biaya untuk sampai pada tujuan mulai dari n

Dengan perhitungan seperti ini, algoritma a-star menjadi optimal. Namun fungsi heuristik masih merupakan perkiraan/estimasi, sama sekali tidak ada rumus khususnya. Artinya, setiap kasus memiliki fungsi heuristik yang berbeda-beda.

2.4 Metode Heuristik

Kata *heuristic* berasal dari sebuah kata kerja bahasa Yunani, *heuriskein*, yang berarti mencari atau memasukkan. Dalam dunia pemrograman, sebagian orang menggunakan kata *heuristic* sebagai lawan kata dari algoritmik, dimana kata *heuristic* ini diartikan sebagai suatu proses yang mungkin dapat menyelesaikan suatu masalah tetapi tidak menjamin bahwa selalu solusi yang dicari selalu dapat ditemukan. Didalam mempelajari metode-metode pencarian ini, kata *heuristic* diartikan sebagai suatu fungsi yang memberikan suatu nilai berupa biaya perkiraan (estimasi) dari suatu solusi.

Heuristic adalah sebuah teknik yang mengembangkan efisiensi dalam proses pencarian. Fungsi *heuristic* ini digunakan untuk mengevaluasi keadaan-keadaan problema untuk mendapatkan solusi yang diinginkan. Teknik pencarian *heuristic* merupakan suatu strategi untuk melakukan proses pencarian secara selektif [10].

Heuristik adalah penilai yang memberi harga pada tiap simpul yang memandu a-star mendapatkan solusi yang diinginkan. Dengan heuristik yang benar, maka a-star pasti akan mendapatkan solusi yang dicari Dengan kata lain, heuristik adalah fungsi optimasi yang menjadikan algoritma a-star lebih baik dari pada algoritma lainnya [11].

Pada algoritma a-star, ada fungsi heuristik yang memprediksikan langkah terpendek dari node asal ke node tujuan. Pada *game the tabperi*, fungsi heuristik yang digunakan berupa jarak dari koordinat asal potongan puzzle hingga koordinat tujuan potongan puzzle yang dimaksudkan untuk mengurangi pencarian yang tidak perlu.

2.5 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyiapkan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Android merupakan generasi baru *platform mobile* yang memberikan kesempatan kepada pengembang untuk melakukan pengembangan sesuai dengan yang diharapkan. Sistem operasi yang mendasari Android merupakan lisensi dibawah naungan GNU, *General Public Licence Version 2* (GPLv2) yang biasa dikenaldengan istilah *copyleft*. Istilah *copyleft* ini merupakan lisensi yang setiap perbaikan oleh pihak ketiga harus terus jatuh dibawah lisensi terms pengembangan memiliki beberapa pilihan dalam membuat aplikasi yang berbasis Android. Namun kebanyakan pengembang menggunakan Eclipse sebagai IDE untuk merancang aplikasi mereka [6]. Adapun versi android adalah sebagai berikut:

Tabel 2. 1 Jenis-jenis *Android*

No	Versi Android	Tahun <i>Release</i>
1.	<i>Android 1.0 Apple Pie</i>	23 September 2008
2.	<i>Android 1.1 Banana Bread</i>	9 Februari 2009
3.	<i>Android 1.5 Cupcake</i>	30 April 2009

4.	<i>Android 1.6 Donut</i>	15 September 2009
5.	<i>Android 2.0/2.1 Eclair</i>	26 Oktober 2009
6.	<i>Android 2.2 Froyo</i>	20 Mei 2010
7.	<i>Android 2.3 Gingerbread</i>	6 Desember 2010
8.	<i>Android 3.0 Honeycomb</i>	22 February 2011
9.	<i>Android 4.0 Ice Cream Sandwich</i>	19 Oktober 2011
10.	<i>Android 4.1 Jelly bean</i>	Juli 2012
11.	<i>Android 4.4 KitKat</i>	3 september 2013
12.	<i>Android 5.0 Lollipop</i>	25 Juni 2014
13.	<i>Android 6.0 Marshmallow</i>	28 Mei 2015
14.	<i>Android 7.0 Nougat</i>	23 Agustus 2016
15.	<i>Android 8.0 Oreo</i>	21 Agustus 2017
16.	<i>Android 9.0 Pie</i>	6 Agustus 2018
17.	<i>Android 10.0 Q</i>	7 Agustus 2019

2.6 Unity

Unity adalah sebuah game engine yang memberikan keuntungan yang besar dibandingkan game engine lainnya yang terdaftar di pasaran saat ini. *Unity* memberikan kapabilitas *drag-anddrop* pada alur kerja visualnya serta mendukung pemrograman pada bahasa C#, yang mana Bahasa tersebut sangat terkenal. *Unity* sudah mendukung pengembangan grafis 3D dan 2D, juga menyediakan seperangkat peralatan untuk dua jenis grafis ini yang selalu berkembang, menjadi semakin mudah digunakan pada setiap pembaharuan. *Unity* juga dibuat khusus untuk mendukung pengembang menggunakan plugin dari *software* pihak ketiga. *Unity* juga menyediakan toko asset (*Asset Store*) sendiri yang menyediakan berbagai *plugin* yang diperlukan untuk pengembang *game*, dari pengembang, oleh pengembang dan untuk pengembang.

Plugin sendiri terdiri dari beberapa hal, diantaranya adalah aset seperti model 3D, sprite 2D, texture, material, efek suara, music, script, efek partikel, dan masih banyak lagi. Semuanya tersedia pada *Unity Asset Store*, mulai dari yang gratis hingga yang berbayar. Disamping itu, sistem berbasis *tile (tile-based system)* untuk pembuatan *game* telah digunakan secara luas, sehingga menjadi salah satu dari standar pendekatan yang digunakan untuk membuat *game* pada sebagian banyak teknologi *game design*. Sistem ini tidak hanya untuk *game* 2D, melainkan *game* 3D juga dapat menggunakan sistem yang sama. *Tile-based system* ini terkenal karena memberikan solusi untuk masalah-masalah yang berbeda yang mana begitu kompleks untuk diselesaikan dengan cara yang lain.

Unity tidak memberikan dukungan secara langsung untuk membuat *game* menggunakan *tile based system*, sementara *plugin* yang terkait dengan fungsionalitas yang memadai dibatasi oleh pembayaran, dan *plugin* yang *gratis* kurang akan fungsionalitas yang baik, jadi pengembang memiliki kesulitan dalam hal akses kepada *tile-based system plugin* yang baik. Studi ini dilakukan untuk membuat *tile-based system* yang lebih kuat dalam hal fungsionalitas serta

dapat diakses oleh pengguna *Unity* terutama dalam bidang edukasi dan pembelajaran [12].



Gambar 2. 1 Logo *Unity*

2.7 UML (*Unified Modelling Language*)

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. Unified Modeling Language (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut bisa berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun [13].

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 [14][12].

Adapun Jenis-jenis Unified Modeling Language (UML) :

2.7.1 Uctivity Diagram

Activity Diagram secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun *use case*. *Activity* diagram dapat juga digunakan untuk memodelkan action yang akan dilakukan saat sebuah operasi dieksekusi, dan memodelkan hasil dari *action* tersebut.

Pada dasarnya *Activity* diagram sering digunakan oleh flowchart. Diagram ini berhubungan dengan diagram Statechart. Diagram Statechart berfokus pada obyek yang dalam suatu proses (proses menjadi suatu obyek), diagram *Activity* berfokus pada aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktifitasaktifitas tersebut bergantung satu sama lain [12].

2.7.2 Sequence Diagram

Sequence Diagram secara grafis menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada sekuensi sebuah *use case* atau operasi.

Sequence diagram merupakan salah satu diagram *Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan; message (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyekobyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut [12].

2.8 Bahasa Pemrograman C#

Microsoft memiliki banyak sekali bahasa pemrograman seperti Visual Basic.NET, Visual Basic, Visual C++, Visual C#.NET, serta visual lain-lain yang dapat anda cari hingga kita bingung memilih mana yang terbaik. Untuk saya pribadi,

mungkin akan lebih menekuni C#.NET ini ketimbang bahasa pemrograman lain seperti Visual Basic maupun C++.

C# merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka .NET Framework. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasabahasa pemrograman lainnya seperti Java, Delphi, Visual Basic, dan lain-lain) dengan beberapa penyederhanaan. Menurut standar ECMA-334 *C# Language Specification*, nama C# terdiri atas sebuah huruf Latin C (U+0043) yang diikuti oleh tanda pagar yang menandakan angka # (U+0023). Tanda pagar # yang digunakan memang bukan tanda kres dalam seni musik (U+266F *keyboard* standar) [15].

2.9 Tabel Periodik Unsur Kimia

Tabel periodik adalah tampilan unsur-unsur kimia dalam bentuk tabel. Unsur-unsur tersebut disusun berdasarkan nomor atom (jumlah proton dalam inti atom), konfigurasi elektron, dan keberulangan sifat kimia. Tabel juga terbagi menjadi empat blok: blok -s, -p, -d, dan -f. Secara umum, dalam satu periode (baris), di sebelah kiri bersifat logam, dan di sebelah kanan bersifat non-logam [16].

Baris pada tabel disebut periode, sedangkan kolom disebut golongan. Enam golongan (kolom) mempunyai nama selain nomor: contoh, unsur golongan 17 adalah halogen, dan golongan 18 adalah gas mulia. Tabel periodik dapat digunakan untuk menurunkan hubungan antara sifat-sifat unsur, dan memperkirakan sifat unsur baru yang belum ditemukan atau disintesis. Tabel periodik memberikan kerangka kerja untuk melakukan analisis perilaku kimia, dan banyak digunakan dalam bidang kimia dan ilmu lainnya.

