

BAB II

TEORI PENUNJANG

2.1 Permainan Komputer (Computer Game)

Permainan komputer adalah permainan video yang dimainkan pada komputer pribadi, dan bukan pada konsol permainan. Permainan ini telah berevolusi dari sistem grafis sederhana menjadi sangat kompleks dan mutakhir. Saat ini permainan komputer telah berubah menjadi permainan online (*online game*) sehingga para pemainnya dapat berinteraksi meskipun satu dengan lainnya saling berjauhan. Permainan online adalah jenis permainan video atau permainan komputer dengan menggunakan jaringan internet sebagai media interaksinya. Permainan ini terdiri dari dua unsur utama, yaitu server dan client. Yang dimaksud dengan server adalah perangkat penyedia layanan gaming sehingga *client – client* dapat terhubung dan saling berkomunikasi ketika memainkan permainan. Suatu server pada prinsipnya hanya melakukan administrasi permainan dan menghubungkan *client-client*. Adapun *client* adalah pengguna permainan dan memakai kemampuan *server*. Contoh permainan online adalah Ragnarok Online, DoTA, World of Warcraft [3].

Terdapat dua jenis strategi permainan yang dapat digunakan, yakni *pure strategy* dimana setiap pemain mempergunakan strategi tunggal dan *mixed strategy*. *Mixed strategy* memiliki ciri setiap pemain menggunakan strategi campuran yang berbeda. *Pure strategy* digunakan untuk jenis permainan yang hasil optimalnya mempunyai *saddle point* (titik keseimbangan antara nilai permainan kedua pemain). Sedangkan *mixed strategy* digunakan untuk mencari solusi optimal dari kasus yang tidak memiliki *saddle point* [2].

2.1.1 Jenis-jenis Permainan

Gim dapat dikategorikan dalam beberapa genre. Andrew Rollings and Ernest Adams menjelaskan kategori *game* sebagai berikut (Rolling dkk, 2003) [2]:

a. *Action Game*

Gim 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 *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*

Gim 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 gim. 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

Gim ini mencoba menciptakan bagaimana rasa mengemudi atau menerbangkan suatu kendaraan sebagai media simulasi. Gim 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 gim ini adalah untuk menguasai dan mengemudi kendaraan dari perspektif pilot atau *driver*. Kebanyakan gim menambahkan tantangan lain seperti balap atau kendaraan tempur saingan. Gim sering ditambahkan pula efek *real*, dengan beberapa gim termasuk fisika dan tantangan seperti manajemen bahan bakar yang lebih realistis.

f. Construction and Management Simulation Game

Jenis gim 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 gim 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 gim ini.

g. Adventure Game

Gim 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. Gim 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.

2.2 Gim Edukasi

Menurut John von Neumann and Oskar Morgenstern tahun 1944, gim adalah: "Permainan terdiri atas sekumpulan peraturan yang membangun situasi bersaing dari dua sampai beberapa orang atau kelompok dengan memilih strategi yang dibangun untuk memaksimalkan kemenangan sendiri atau pun untuk meminimalkan kemenangan lawan. Peraturan-peraturan menentukan kemungkinan tindakan untuk setiap pemain, sejumlah keterangan diterima setiap pemain sebagai kemajuan bermain, dan sejumlah kemenangan atau kekalahan dalam berbagai situasi". Gim edukasi adalah gim digital yang dirancang untuk pengayaan pendidikan (mendukung pengajaran dan pembelajaran), menggunakan teknologi multimedia interaktif Menurut Hurd dan Jenuings, perancang yang baik haruslah memenuhi kriteria dari *education game* itu sendiri. Berikut ini adalah beberapa kriteria dari sebuah *education game* [4], yaitu:

1. Nilai Keseluruhan (*Overall Value*)

Nilai keseluruhan dari suatu gim terpusat pada desain dan panjang durasi gim. Aplikasi ini dibangun dengan desain yang menarik dan interaktif. Untuk penentuan panjang durasi, aplikasi ini menggunakan fitur timer.

2. Dapat Digunakan (*Usability*)

Mudah digunakan dan diakses adalah poin penting bagi pembuat gim. Aplikasi ini merancang sistem dengan *interface* yang *user friendly* sehingga *user* dengan mudah dapat mengakses aplikasi.

3. Keakuratan (*Accuracy*)

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

4. Kesesuaian (*Appropriateness*)

Kesesuaian dapat diartikan bagaimana isi dan desain gim 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 gim ke target *user*. Agar dapat relevan terhadap *user*, sistem harus membimbing mereka dalam pencapaian tujuan pembelajaran. Karena aplikasi ini ditujukan untuk anak-anak, maka desain antarmuka harus sesuai dengan nuansa anak-anak, yaitu menampilkan warna-warna yang ceria.

6. Tujuan (*Objectives*)

Tujuan 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 gim atau tidak, *feedback* harus disediakan. Aplikasi ini menyajikan animasi dan efek suara yang mengindikasikan kesuksesan atau kegagalan permainan.

2.3 Algoritma Dijkstra

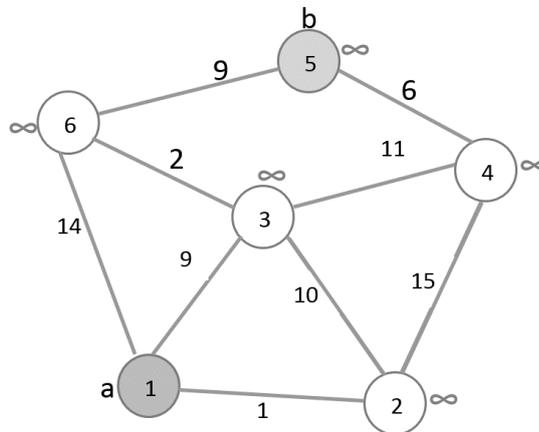
Sudah banyak algoritma yang bisa digunakan untuk tujuan pencarian rute terpendek. Tidak bisa di pungkiri Dijkstra masih menjadi salah satu yang populer dari sekian banyak algoritma tersebut. Algoritma Dijkstra, dinamakan sesuai dengan nama penemunya, seorang ilmuwan komputer berkebangsaan Belanda

yang bernama Edsger Dijkstra, adalah algoritma yang digunakan untuk mencari lintasan terpendek pada sebuah graf berarah. Ian Millington (2006) pada bukunya yang berjudul *Artificial Intelligence for Games* menyebut algoritma Dijkstra sebagai versi ringkas algoritma a star. Keringkasan yang dimaksud adalah bahwa keduanya memiliki kesamaan dalam proses pencarian rute terdekat, hanya saja pada algoritma Dijkstra fungsi heuristik tidak diperhitungkan sebagaimana yang dilakukan oleh A star.

Cara kerja algoritma Dijkstra memakai strategi *greedy*, dimana pada setiap langkah dipilih sisi dengan bobot paling kecil yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih.

Dijkstra merupakan salah satu varian bentuk algoritma populer dalam pemecahan persoalan terkait masalah optimasi pencarian lintasan terpendek. Sebuah lintasan yang mempunyai panjang minimum dari verteks a ke z dalam *graph* berbobot. Bobot tersebut adalah bilangan positif jadi tidak dapat dilalui oleh node negatif. Namun jika terjadi demikian, maka penyelesaian yang diberikan adalah *infinity* (tak hingga). Pada algoritma Dijkstra, node digunakan karena algoritma Dijkstra menggunakan graf berarah untuk penentuan rute listasan terpendek.

Algoritma Dijkstra bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Misalkan titik menggambarkan gedung dan garis menggambarkan jalan, maka algoritma Dijkstra melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik [2].



Gambar II.1 Contoh Ketersambungan Titik dalam Algoritma Dijkstra

Pertama-tama tentukan titik mana yang akan menjadi node awal, lalu beri bobot jarak pada node pertama ke node terdekat satu per satu, Dijkstra akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap. Inilah urutan logika dari algoritma Dijkstra [2]:

1. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga terhadap node lain (belum terisi)
2. Set semua node “Belum terjamah” dan set node awal sebagai “Node keberangkatan”
3. Dari node keberangkatan, pertimbangkan node tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan. Sebagai contoh, jika titik keberangkatan A ke B memiliki bobot jarak 6 dan dari B ke node C berjarak 2, maka jarak ke C melewati B menjadi $6+2=8$. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.
4. Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah terjamah sebagai “Node terjamah”. Node terjamah tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.

5. Set “Node belum terjamah” dengan jarak terkecil (dari node keberangkatan) sebagai “Node Keberangkatan” selanjutnya dan lanjutkan dengan kembali ke step 3 [2].

2.4 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 License Version 2 (GPLv2)* yang biasa dikenal dengan istilah *copyleft*. Istilah *copyleft* ini merupakan lisensi yang setiap perbaikan oleh pihak ketiga harus terus jatuh dibawah lisensi terms pengembang 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 II-1 Jenis-jenis Android

| No | Versi Android | Tahun <i>Release</i> |
|----|--------------------------------|----------------------|
| 1. | <i>Android 1.5 Cupcake</i> | 30 April 2009 |
| 2. | <i>Android 1.6 Donut</i> | 15 September 2009 |
| 3. | <i>Android 2.0/2.1 Eclair</i> | 26 Oktober 2009 |
| 4. | <i>Android 2.2 Froyo</i> | 20 Mei 2010 |
| 5. | <i>Android 2.3 Gingerbread</i> | 6 Desember 2010 |

| | | |
|----|---------------------------------------|-------------------|
| 6. | <i>Android 3.0 Honeycomb</i> | 22 February 2011 |
| 7. | <i>Android 4.0 Ice Cream Sandwich</i> | 19 Oktober 2011 |
| 8. | <i>Android 4.1 Jelly bean</i> | Juli 2012 |
| 9. | <i>Android 4.4 KitKat</i> | 3 september 2013 |
| 10 | <i>Android 5.0 Lollipop</i> | 25 Juni 2014 |
| 11 | <i>Android 6.0 Marshmallow</i> | 28 Mei tahun 2015 |
| 12 | <i>Android 7.0 Nougat</i> | 23 Agustus 2016 |
| 13 | <i>Android 8.0 Oreo</i> | 21 Agustus 2017 |

2.5 Unity

Unity merupakan perangkat lunak untuk pengembangan gim *multiplatform*. Editor pada Unity dibuat dengan user interface yang sederhana dan mudah dipahami oleh *game developer* pemula sekalipun. Grafis pada *Unity* dibuat dengan grafis tingkat tinggi untuk OpenGL dan DirectX. Unity mendukung semua format file, terutamanya format umum. Unity cocok dengan versi 64-bit. Dapat beroperasi pada Mac OS X dan Windows dan dapat menghasilkan gim untuk Mac, Windows, Wii, iPhone, iPad maupun Android.

Unity merupakan salah satu *game engine* paling terkemuka dewasa ini. Unity adalah sebuah *software development* yang terintegrasi untuk menciptakan video gim atau konten lainnya seperti visualisasi arsitektur atau real-time animasi baik yang bernuansa 2D maupun 3D. *Unity* dapat digunakan pada Microsoft Windows dan Mac OS X. Permainan yang dihasilkan dapat dijalankan secara *multiplatform*. *Unity* juga dapat menghasilkan permainan untuk *browser* dengan menggunakan plugin Unity Web Player.

Unity secara lebih rinci dapat dimanfaatkan untuk pengembangan *3D video game*, real time animasi 3D dan visualisasi arsitektur maupun konten interaktif serupa lainnya. Editor Unity dapat menggunakan *plugin* untuk *web player* dan menghasilkan *game browser* yang didukung oleh Windows maupun Mac. *Plugin web player* dapat juga dipakai untuk widgets Mac. Unity juga mendukung *console* terbaru seperti PlayStation 3 dan Xbox 360. Tahun 2009 *Unity Technology* menjadi 5 perusahaan gim terbesar di dunia setelah tahun sebelumnya di tahun 2006, menjadi juara dua pada Apple Design Awards. Hingga puncaknya di tahun 2010 *Unity* berhasil memperoleh Technology Innovation Award yang diberikan oleh Wall Street Journal.

Unity Technology yang merupakan pengembang resmi Unity juga menyediakan asset store. Asset store terdiri dari berbagai konten, model, prefab, script, sound dan kebutuhan lain yang diperlukan dalam pembangunan sebuah permainan. Editor Unity dapat menyimpan metadata. Editor Unity dapat diperbaharui dengan sesegera mungkin seperti file yang telah dimodifikasi. Server aset Unity juga berjalan pada Mac, Windows, Linux dan juga berjalan pada PostgreSQL, database server opensource.

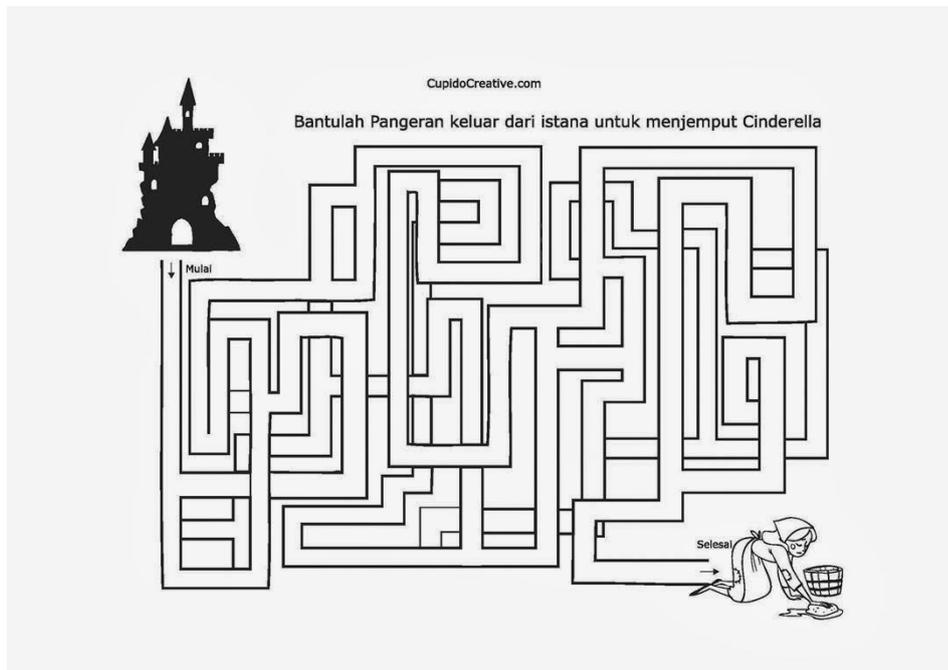
Unity memiliki dua lisensi yakni *Unity* dan *Unity Pro*. Versi *Unity* tersedia dalam bentuk gratis, sedang versi *Unity Pro* didistribusikan secara berbayar. Versi *Unity Pro* memiliki berbagai fitur bawaan seperti efek post processing, render, efek tekstur serta berbagai fitur spesial yang tidak tersedia di versi gratis. *Unity* dan *Unity Pro* menyediakan berbagai tutorial, konten, completed project, wiki, dukungan melalui forum dan pembaruan ke depannya [2].



Gambar II.2 Logo Unity

2.6 Labirin (*The Maze*)

Maze atau labirin adalah sebuah *puzzle* dalam bentuk percabangan jalan yang kompleks dan memiliki banyak jalan buntu. Tujuan permainan ini adalah pemain harus menemukan jalan keluar dari sebuah pintu masuk ke satu atau lebih pintu keluar. Bisa juga kondisi pemain menang yaitu ketika dia mencapai suatu titik di dalam *maze* tersebut. *Maze* dalam dunia nyata banyak dibuat di taman atau ruangan-ruangan dengan pembatas berupa pagar tanaman, tembok atau pagar. Ukurannya bervariasi, tergantung ukuran ruangan atau taman tersebut [5].



Gambar II.3 Labirin