

BAB II

TEORI PENUNJANG

1.1 Pengertian Game

Game atau permainan adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada yang menang dan ada yang kalah, biasanya dalam konteks tidak serius dengan tujuan refreshing. Pada zaman sekarang bermain game telah menjadi gaya hidup masyarakat modern. *Game* merupakan hiburan berbagai usia yang mampu membawa nuansa menyenangkan [5]

Teori permainan pertama kali dikemukakan oleh sekelompok ahli Matematika pada tahun 1944. John von Neumann and Oskar Morgenstern menyatakan bahwa 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 [9]

Berdasarkan pendapat seorang ahli, teori permainan adalah bagian dari ilmu pengetahuan yang berkaitan dengan pembuatan keputusan pada saat ada dua pihak atau lebih berada dalam kondisi bersaing. Pihak-pihak yang bersaing ini diasumsikan bersifat rasional dan cerdas. Hal ini diartikan bahwa masing-masing pihak akan melakukan strategi rasional demi tujuan memenangkan persaingan. Masing-masing pihak juga berusaha mengetahui strategi lawan. Selanjutnya pihak ini disebut pemain. (Dimiyati 1992).

Teori permainan merupakan teori yang menggunakan pendekatan matematis dalam merumuskan situasi persaingan dan konflik antara berbagai kepentingan. Teori ini dikembangkan untuk menganalisa proses pengambilan keputusan yaitu strategi optimum dari situasi-situasi persaingan yang berbeda-beda dan melibatkan dua atau lebih kepentingan [10].

Tujuan teori ini adalah untuk menganalisis proses pengambilan keputusan. Kegunaan dari teori permainan adalah metodologi yang disediakan

untuk menstruktur dan menganalisis masalah pemilihan strategi. Menggunakan teori permainan, maka langkah pertama adalah menentukan secara eksplisit pemain, strategi yang ada, dan juga menentukan preferensi serta reaksi dari setiap pemain.

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*.

1.1.1 Jenis-Jenis Game

1. *Action*

Action game adalah sebuah game yang menekankan reflek pemain, seperti koordinasi tangan dan mata karena pemain harus melakukan gerakangerakan yang cepat didalam game tersebut. Contoh jenis dari game ini adalah : Call of Duty series dan Battlefield series.

2. *Adventure*

Adventure game adalah game yang mengedepankan cerita dan petualangan yang penuh dengan penuh aksi hingga game tersebut selesai. Contoh game dengan tipe ini adalah Mario Bros.

3. *Driving*

Driving game adalah game yang membuat pemainnya seolah dapat mengemudikan kendaraan seperti mobil, pesawat, atau kapal laut. Biasanya game ini membuat pemain untuk berlomba menjadi yang tercepat dari pemain lainnya, karena itu game driving sering juga disebut game racing. Contoh game tipe driving adalah Need For Speed dan Dirt.

4. *Role Playing*

Role Playing Game (RPG), adalah game yang memfokuskan peran dari tokoh yang pemain mainkan. Biasanya pemain yang hanya mencari

aksi dari sebuah game akan menghindari game jenis ini karena banyaknya tulisan dan cerita yang harus dimengerti oleh pemain untuk menyelesaikan game ini. Contoh game RPG adalah Final Fantasy series.

5. *Simulation*

Simulation game adalah tipe game yang bertujuan melatih pemain seperti keadaan sebenarnya dan menggambarkan dunia di dalamnya sedekat mungkin dengan dunia nyata dengan memperhatikan dengan detail berbagai faktor. Salah satu contoh dari game Simulasi adalah The Sims.

6. *Strategy*

Strategy Game adalah game dengan pengambilan keputusan oleh pemain untuk dapat menyelesaikan game tersebut diperlukan keahlian berpikir dan memutuskan setiap gerakan secara hati-hati dan terencana. Contoh game seperti ini adalah Yu-Gi-Oh Duel Monster.

7. *Maze*

Game maze adalah sebuah tipe permainan yang bertujuan mencari jalan keluar serta memiliki jalan yang kompleks, dan memiliki banyak jalan buntu. Pada permainan *game maze* pemain tidak harus memiliki reflek dan akurasi seperti game lainnya, karena sifat gameplay-nya yang santai namun membutuhkan konsentrasi pemain dalam menyelesaikan suatu masalah yang ada. Contoh dari *game maze* adalah Pacman. Di dalam *game maze* biasanya pemain mencari jalan keluar, dan atau mencari jalan keluar sambil menghindari serangan musuh yang berada di dalam game tersebut. Selain itu, pada game ini pemain biasanya dihadapkan dengan berbagai teka – teki sehingga membuat pemain penasaran dan ingin kembali memainkan game tersebut.

Sedangkan untuk Jenisnya, game dibagi menjadi dua. Yaitu game murni dan game *learning*. [11].

1. Game Murni

Game murni adalah sebuah game yang tidak memiliki unsur *learning* atau pembelajaran di dalamnya. Biasanya pada game seperti ini pemain hanya bertujuan untuk bersenangsenang atau sekedar menghabiskan waktu dalam bermain.

2. Game Learning

Game learning adalah sebuah game yang memfokuskan sarana *learning* dalam bermain game, biasanya dalam *game learning* ini masih terkandung mata pelajaran tertentu, maka ketika dimainkan oleh murid mereka secara tidak sadar juga telah memperoleh pengetahuan baru dibidang mata pelajaran tersebut. *Education games* yaitu suatu kegiatan yang sangat menyenangkan dan dapat merupakan cara atau alat pendidikan yang bersifat mendidik. [11]

1.2 Algoritma Dijkstra

Algoritma ini diberi nama sesuai nama penemunya, Edsger Wybe Dijkstra. Algoritma Dijkstra mencari lintasan terpendek dalam sejumlah langkah. Algoritma ini menggunakan prinsip *Greedy* yang menyatakan bahwa pada setiap langkah kita memilih sisi yang berbobot minimum dan memasukkannya ke dalam himpunan solusi. Biasanya untuk menentukan besar bobot pada algoritma dijkstra maka digunakanlah rumus jarak rumus jarak yaitu :

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

Dimana :

d = Jarak

x_1, y_1 = Koordinat awal

x_2, y_2 = Koodinat Tujuan

Input algoritma ini adalah sebuah graph berarah yang berbobot (weighted directed graph) G dan sebuah sumber vertex s dalam G dan V adalah himpunan semua vertices dalam graph G [12]

Berikut *Pseudocode* dari algoritma Dijkstra dalam mencari rute terpendek pada sebuah graf [13]:

```
algorithm DijkstraShortestWeightedPath(G, s)
  _pre-cond_: G is a weighted (directed or undirected) graph, and s is one of
  its nodes.
  _post-cond_:  $\pi$  specifies a shortest weighted path from s to each node of G,
  and
  d specifies their lengths.
  begin d(s) = 0,  $\pi(s) = \_$ 
  for other v, d(v) =  $\infty$  and  $\pi(v) = \text{nil}$ 
  handled =  $\emptyset$ 
  notHandled = priority queue containing all nodes. Priorities given by d(v).
  loop
  _loop-invariant_: See above.
  exit when notHandled =  $\emptyset$ 
  let u be a node from notHandled with smallest d(u)
  for each v connected to u
  foundPathLength = d(u) + wu,v
  if d(v) > foundPathLength then
  d(v) = foundPathLength
   $\pi(v) = u$ 
  (update the notHandled priority queue)
  end if
  end for
  move u from notHandled to handled
  end loop
  return _d,  $\pi$ _
end algorithm.
```

Dengan Properti algoritma Dijkstra:

1. Matriks ketetanggaan $M[m_{ij}]$

$$m_{ij} = \text{bobot sisi } (i, j)$$

$$m_{ij} = 0$$

$m_{ij} = \infty$, jika tidak ada sisi dari simpul i ke simpul j

2. Larik $S = [s_i]$ yang dalam hal ini,

$s_i = 1$, jika simpul i termasuk ke dalam lintasan terpendek

$s_i = 0$, jika simpul i tidak termasuk ke dalam lintasan terpendek

3. Larik/tabel $D = [d_i]$ yang dalam hal ini,

$d_i =$ panjang lintasan dari simpul awal s ke simpul i

Algoritma Lintasan Terpendek Dijkstra (Mencari lintasan terpendek dari simpul a ke semua simpul lain)

Langkah 0 (inisialisasi):

- inisialisasi $s_i = 0$ dan $d_i = m_{ai}$ untuk $i = 1, 2, \dots, n$

Langkah 1:

- isi s_a dengan 1 (karena simpul a adalah simpul asal lintasan terpendek, jadi sudah pasti terpilih)

- isi d_a dengan ∞ (tidak ada lintasan terpendek dari simpul a ke a)

Langkah 2, 3, ..., n :

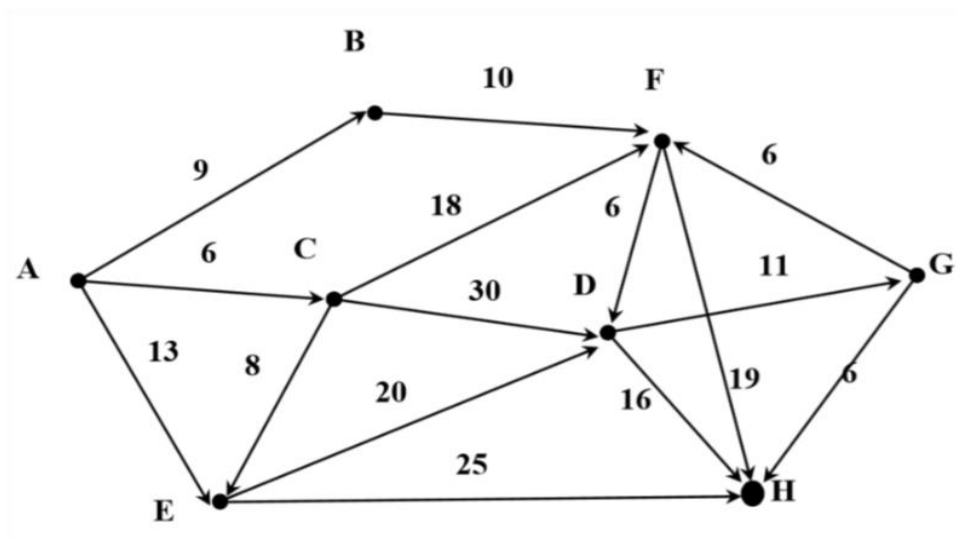
- cari j sedemikian sehingga $s_j = 0$ dan $d_j = \min\{d_1, d_2, \dots, d_n\}$

- isi s_j dengan 1

perbarui d_i , untuk $i = 1, 2, 3, \dots, n$ dengan:

d_i (baru) = $\min\{d_i$ (lama), $d_j + m_{ji}\}$.

Misal diberikan graph berbobot seperti pada gambar 2.1 di bawah [14]



Gambar 2.1 Contoh Graph Berbobot [14]

Misalnya diberikan graph berbobot dan berarah seperti gambar di atas. Akan dicari lintasan terpendek dari simpul A ke semua simpul lain.

Jadi, lintasan terpendek dari [15]:

A ke C adalah A, C dengan panjang = 6

A ke B adalah A, B dengan jarak = 9

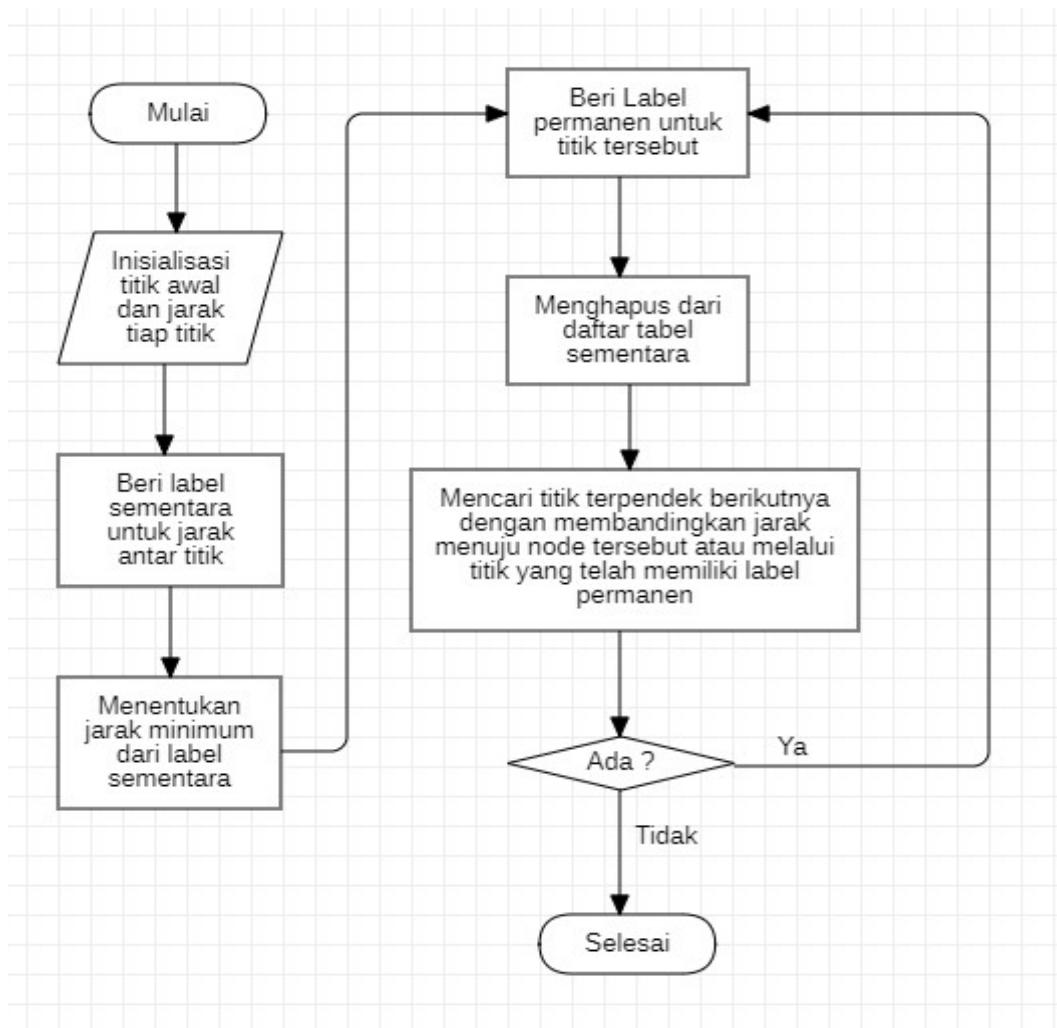
A ke E adalah A, B, E dengan jarak = 13

A ke F adalah A, B, F dengan jarak = 19

A ke D adalah A, B, F, D dengan jarak = 25

A ke G adalah A, B, F, D, G dengan jarak = 36

A ke H adalah A, B, E, H dengan jarak = 38.



Gambar 2.2 Flowchart Algoritma Dijkstra

1.3 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 menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Adapun versi android diantaranya :

Tabel 2.1 Versi dan tahun 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 Éclair</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

1.4 Unity

Unity adalah mesin permainan *cross-platform* yang dikembangkan oleh Unity Technologies, pertama kali diumumkan dan dirilis pada bulan Juni 2005 di Apple Inc. Worldwide Developers Conference sebagai mesin permainan Mac OS X-eksklusif. Pada 2018, mesin telah diperluas untuk mendukung lebih dari 25 platform [16]. Kemudian Menurut Muhammad Fauji, *Unity 3D* dapat mengolah berbagai data seperti objek tiga dimensi, suara, tekstur, dan masih banyak yang lainnya. Fitur pendukung dari *Unity 3D* ini adalah dapat menangani grafik dua dimensi maupun tiga dimensi. Namun *Unity 3D* ini lebih mengedepankan pada aspek pembuatan grafik tiga dimensi [17]. Dari beberapa game engine yang sama-sama menangani grafik tiga dimensi, *Unity 3D* dapat menangani lebih banyak platform contohnya antara lain yaitu Windows, MacOS X, iOS, PS3, wii, Xbox 360, dan Android lebih banyak daripada game engine pada umumnya. Selain itu *Unity 3D* memiliki kerangka kerja (*framework*) lengkap untuk dikembangkan secara profesional. Sistem inti engine ini menggunakan beberapa pilihan bahasa pemrograman, diantaranya C#, javascript maupun boo [18]. Adapun *Unity*

3D editor adalah salah satu fitur tambahan dari *Unity 3D* dimana fitur ini menyediakan beberapa alat untuk mempermudah pengembangan yaitu *Unity Tree* dan *terrain creator* untuk mempermudah pembuatan vegetasi dan *terrain* serta *MonoDevelop* untuk proses pemrograman.