

Aplikasi Pohon Keputusan dalam Memilih Persona dalam Role-playing Video Game Persona 4

Nabil Nabighah 13519168¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

¹13519168@std.stei.itb.ac.id

Abstract—Pada era teknologi game menjadi sebuah hal yang sangat mempengaruhi kehidupan seseorang. Game memiliki banyak fungsi salah satunya adalah sebagai alat penghibur. Semakin berkembangnya zaman, game semakin memiliki gaya permainan (genre) yang beragam sebagai contoh genre sport, RPG, FPS, MOBA, Horror, Action, Adventure. Salah satu contoh game RPG adalah Persona 4. Banyak orang yang bermain game tersebut secara bebas tanpa menggunakan suatu strategi, padahal jika menggunakan suatu strategi yang baik dan tepat dapat mempermudah untuk melawan musuh sehingga mudah untuk memenangkan permainan tersebut. Dengan menggunakan pohon keputusan pemain dapat memilih persona yang tepat untuk melawan musuh sehingga dapat memenangkan permainan lebih mudah.

Keywords—Game, RPG, Persona 4, Pohon Keputusan.

I. PENDAHULUAN

Teknologi berkembang sangat pesat dan semakin maju sehingga tidak bisa dipungkiri bahwa teknologi sangat mempengaruhi kehidupan manusia. Game atau permainan menjadi salah satu sumber hiburan untuk setiap orang. Game dapat dibedakan menjadi beberapa kategori, dalam hal ini game dibedakan dalam aspek waktu dan medianya. Kategori pertama adalah game tradisional, game tradisional adalah sebuah game sederhana yang tidak menggunakan teknologi sebagai media untuk bermain melainkan menggunakan suatu barang biasanya game tradisional dimainkan oleh orang zaman dahulu. Contoh game tradisional yang ada di Indonesia adalah layang-layang, congklak, lompat tali, petak umpet, dan kelereng. Kategori kedua adalah game modern, game modern adalah sebuah game yang menggunakan teknologi sebagai media untuk bermain. Game modern lebih suka dimainkan dibanding dengan game tradisional. Semakin berkembangnya zaman, beragam game modern mulai bermunculan.

Pada tahun 2008 rilis sebuah video game berjudul *Persona 4* dikenal juga sebagai *Shin Megami Tensei: Persona 4* untuk Playstation 2 oleh sebuah perusahaan bernama Atlus di Jepang. Sebuah game bergaya *role-playing*. Perkembangan game persona 4 sangat pesat hingga ke mancanegara. Pada 2011 Perusahaan Atlus merilis *remake* untuk game *Persona 4* yaitu *Persona 4 Golden*. Kedua game tersebut sangat sukses bahkan sampai 31 Desember 2019 tercatat game tersebut telah lebih dari 1.5 juta kopi terjual. Game ini bercerita mengenai seorang siswa menengah atas yang pindah ke perdesaan dari perkotaan untuk

satu tahun. Selama satu tahun ini, siswa tersebut terlibat kasus pembunuhan misterius dengan memanfaatkan sebuah kekuatan yang dinamakan 'persona'. Pemain dituntut untuk menyelesaikan kasus tersebut dan mencegah terjadinya pembunuhan. Pemain dapat menggunakan kekuatan persona tersebut untuk melawan musuh yang menyebabkan pembunuhan itu terjadi. Pemain dapat memilih banyak persona sesuai dengan atribut yang dimiliki persona tersebut. Pemain dapat memenangkan permainan dengan beberapa alternatif. Pada makalah ini akan dibahas mengenai pengaplikasian pohon yaitu pohon keputusan yang dipelajari dalam Matematika Diskrit dalam pengambilan keputusan pemilihan persona dalam melawan musuh yang disebut *Shadow*.



Gambar 1 *Shin Megami Tensei: Persona 4*

Sumber(https://en.wikipedia.org/wiki/File:Shin_Megami_Tensei_Persona_4.jpg)

II. DASAR TEORI

A. Graf (Graph)

1. Definisi Graf

Graf adalah himpunan tidak-kosong dari simpul-simpul (*vertices*) yang dihubungkan dengan sisi (*edges*). Sisi tersebut adalah anggota dari himpunan sisi E, Graf dinotasikan

sebagai $G = (V, E)$, dalam hal ini :

V = himpunan tidak-kosong dari simpul-simpul (*vertices*) = $\{v_1, v_2, \dots, v_n\}$

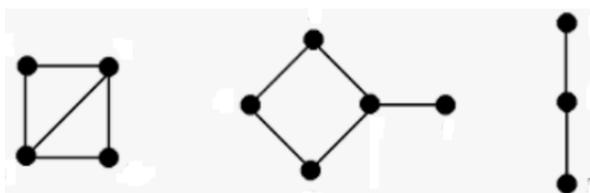
E = himpunan sisi (*edges*) yang menghubungkan sepasang simpul = $\{e_1, e_2, \dots, e_n\}$

2. Jenis - Jenis Graf

Graf dapat dikategorikan menjadi dua jenis berdasarkan keberadaan gelang atau sisi ganda pada suatu graf, yaitu :

1. Graf Sederhana (*simple graph*)

Graf sederhana adalah graf yang tidak mengandung gelang maupun sisi-ganda. Misalkan graf G_1 pada gambar 2 adalah sebuah contoh graf sederhana.

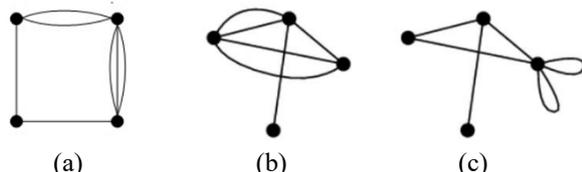


Gambar 2. Contoh graf sederhana

Sumber([https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20\(2015\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf))

2. Graf Tidak Sederhana (*unsimple-graph*)

Graf tidak sederhana merupakan graf yang mengandung sisi ganda atau gelang. Gambar 3 merupakan sebuah contoh graf tidak sederhana



Gambar 3. Contoh graf tidak sederhana

Sumber([https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20\(2015\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf))

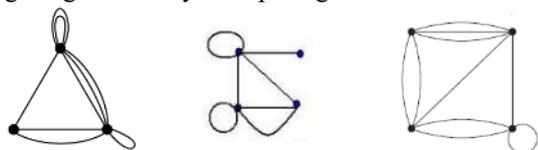
Graf tak-sederhana dibedakan lagi menjadi :

1. Graf ganda (*multi-graph*)

Graf ganda merupakan graf yang mengandung sisi ganda. Contohnya pada gambar 3(c).

2. Graf semu (*pseudoo-graph*)

Graf semu adalah graf yang mengandung sisi gelang. Contohnya ada pada gambar 4.



Gambar 4. Contoh graf ganda

Sumber([https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20\(2015\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf))

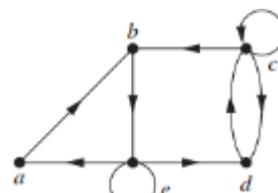
Selain berdasarkan keberadaan gelang atau sisi, graf dapat dikategorikan berdasarkan orientasi arah pada sisi graf. Secara umum graf tersebut dapat dibedakan atas dua jenis, yaitu :

1. Graf Tidak Berarah (*undirected graph*)

Graf tidak berarah merupakan graf yang sisi-sisinya tidak mempunyai orientasi arah. Contoh graf tidak berarah adalah gambar 2, gambar 3, dan gambar 4.

2. Graf Berarah (*directed graph* atau *digraph*)

Graf berarah merupakan graf yang sisi-sisinya diberikan suatu orientasi arah. Misalkan graf G_4 pada gambar 5.



Gambar 5. Contoh Graf Berarah

Sumber([https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20\(2015\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf))

Tabel 1 Jenis-jenis graf

Jenis	Sisi	Sisi ganda dibolehkan?	Sisi gelang dibolehkan?
Graf sederhana	Tak-berarah	Tidak	Tidak
Graf ganda	Tak-berarah	Ya	Tidak
Graf semu	Tak-berarah	Ya	Ya
Graf berarah	Berarah	Tidak	Ya
Graf-ganda berarah	Berarah	Ya	Ya

3. Terminologi Graf

a. Ketetanggaan (*Adjacent*)

Dua buah simpul dikatakan *bertetangga* apabila keduanya terhubung langsung oleh sisi.

b. Bersisian (*Incidency*)

Sebuah sisi $e = (v_j, v_k)$ dikatakan bersisian jika e bersisian dengan simpul v_j , atau e bersisian dengan simpul v_k .

c. Simpul Terpencil (*Isolated Vertex*)

Simpul terpencil adalah simpul yang tidak mempunyai sisi yang bersisian dengan simpul tersebut.

d. Graf kosong (*Null Graph* atau *empty graph*)

Graf kosong adalah graf yang himpunan sisinya merupakan himpunan kosong.

e. Derajat (*Degree*)

Derajat suatu simpul adalah jumlah sisi yang bersisian dengan simpul tersebut. Notasi derajat adalah $d(v)$. Pada graf berarah derajat dibedakan menjadi derajat masuk $d_{in}(v)$ dan derajat keluar $d_{out}(v)$.

f. Lintasan (*Path*)

Lintasan yang panjangnya n dari simpul awal v_0 ke simpul tujuan v_n di dalam graf G adalah barisan berselang-seling simpul-simpul dan sisi-sisi yang berbentuk $v_0, e_1, v_1, e_2, \dots, v_{n-1}, e_n, v_n$ sedemikian sehingga $e_1 = (v_0, v_1)$, $e_2 = (v_1, v_2)$, ..., $e_n = (v_{n-1}, v_n)$

(v_1, v_n) adalah sisi-sisi dari graf G .

g. Siklus (*Cycle*) atau Sirkuit (*Circuit*)

Siklus atau sirkuit adalah lintasan yang berawal dan berakhir pada simpul yang sama.

h. Keterhubungan (*Connected*)

Dua buah simpul v_1 dan simpul v_2 disebut terhubung jika terdapat lintasan dari v_1 ke v_2 . Graf G disebut graf terhubung (*connected graph*) jika untuk setiap pasang simpul v_i dan v_j dalam himpunan V terdapat lintasan dari v_i ke v_j .

Graf berarah G dikatakan terhubung jika graf tidak berarahnya terhubung (graf tidak berarah dari G diperoleh dengan menghilangkan arahnya). Dua simpul, u dan v , pada graf berarah G disebut terhubung kuat (*strongly connected*) jika terdapat lintasan berarah dari u ke v dan juga lintasan berarah dari v dan u . Sedangkan dikatakan terhubung lemah (*weakly connected*) jika u dan v tidak terhubung kuat tetapi terhubung pada graf tidak berarahnya.

Graf berarah G disebut graf terhubung kuat (*strongly connected graph*) jika untuk setiap pasang simpul a dan b pada G , terhubung kuat dan graf terhubung lemah jika tidak.

B. Pohon (Tree)

1. Definisi Pohon

Pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit. Kumpulan pohon yang saling lepas disebut hutan. Hutan juga disebut graf tidak terhubung yang tidak mengandung sirkuit.

2. Sifat-Sifat Pohon

Misalkan $G = (V, E)$ adalah graf tak-berarah sederhana dan jumlah simpulnya n . Maka beberapa pernyataan ini ekuivalen :

1. G adalah pohon
2. G terhubung dan semua sisinya adalah jembatan
3. G terhubung dan memiliki sisi m sebanyak $n-1$
4. G tidak mengandung sirkuit
5. Apabila ditambah satu sisi pada graf G maka akan membuat hanya satu sirkuit
6. Setiap pasang simpul dalam G terhubung dengan lintasan tunggal.

3. Pohon Merentang (*spanning tree*)

Pohon merentang dari graf terhubung adalah upagraf merentang yang berupa pohon. Pohon merentang diperoleh dengan cara memutus sirkuit di dalam graf. Setiap graf terhubung paling sedikit mempunyai satu buah pohon merentang. Hutan merentang adalah graf tak-terhubung dengan k komponen mempunyai k buah hutang merentang (*spanning forest*).

4. Pohon Merentang Minimum

Graf terhubung-berbobot bisa jadi mempunyai lebih dari satu pohon merentang. Pohon merentang minimum adalah pohon merentang yang mempunyai bobot minimum (*minimum spanning tree*). Untuk membuat pohon merentang minimum

dapat menggunakan dua cara, yaitu :

1. Algoritma Prim

Langkah pertama, ambil sisi dari graf G yang berbobot paling minimum, masukkan ke dalam pohon T . Langkah kedua, pilih sisi (u,v) yang mempunyai bobot minimum juga bersisian dengan simpul di T , tetapi (u, v) tidak boleh membentuk sirkuit di T . Kemudian masukkan sisi tersebut kedalam T . Langkah ketiga, ulangi langkah kedua sebanyak $n-2$ kali, n adalah jumlah simpul.

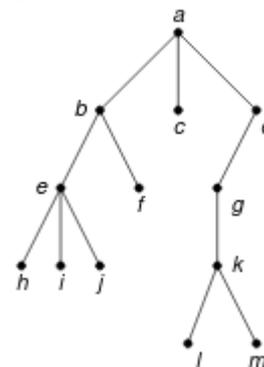
2. Algoritma Kruskal

Langkah pertama, siapkan sebuah T yang kosong. Langkah kedua, pilih sisi (u,v) yang mempunyai bobot minimum juga tidak boleh membentuk sirkuit di T . Kemudian masukkan sisi tersebut kedalam T . Langkah ketiga, ulangi langkah kedua sebanyak $n-1$ kali, n adalah jumlah simpul.

5. Pohon berakar (*rooted tree*)

Pohon berakar (*rooted tree*) adalah pohon yang sebuah simpulnya diperlakukan sebagai dan sisinya diberi arah agar menjadi graf berarah.

6. Terminologi pada Pohon Berakar



Gambar 6. Contoh sebuah pohon

Sumber([http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20\(2013\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20(2013).pdf))

- a. Anak (*child* atau *children*) dan Orang tua (*parent*)
 b, c , dan d adalah anak dari simpul a . a adalah orang tua dari b, c , dan d .
- b. Lintasan (*path*)
Lintasan dari a ke m adalah d, g, k, m dan panjang lintasan dari a ke m adalah 4.
- c. Saudara kandung (*sibling*)
 e adalah saudara kandung f sedangkan e bukan saudara kandung g karena berasal dari orang tua yang berbeda.
- d. Upapohon (*subtree*)
- e. Derajat (*degree*)
Derajat sebuah simpul adalah jumlah upapohon atau jumlah anak pada simpul tersebut dengan kata lain derajat pada pohon adalah derajat-keluar. Contohnya, derajat a adalah 3,

derajat b adalah 2, derajat d adalah 1 dan derajat c adalah 0.

f. Daun (*leaf*)

Daun adalah simpul yang berderajat nol (atau tidak mempunyai anak). Contohnya, Simpul h, i, j, f, c, l , dan m .

g. Simpul Dalam (*internal nodes*)

Simpul dalam adalah simpul yang mempunyai anak. Contohnya, simpul b, d, e, g , dan k .

h. Aras (*level*) atau Tingkat

Aras adalah panjang dari akar ke suatu simpul dalam pohon.

i. Tinggi (*height*) atau Kedalaman (*depth*)

Tinggi atau kedalaman dari suatu pohon ada aras maksimum pada pohon tersebut. Pohon pada gambar 6 mempunyai tinggi 4.

7. Pohon Terurut (*ordered tree*)

Pohon terurut adalah pohon berakar yang urutan anak-anaknya diperhatikan.

8. Pohon n -ary

Pohon n -ary adalah pohon berakar yang setiap simpul cabangnya mempunyai maksimal n buah anak. sebuah pohon n -ary disebut pohon yang teratur atau penuh jika setiap simpul cabangnya mempunyai tepat n buah anak.

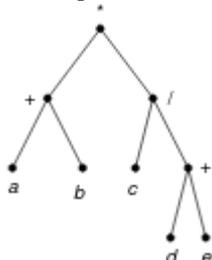
9. Pohon Biner (*binary tree*)

Pohon biner adalah pohon n -ary dengan $n = 2$. Pohon ini sangat banyak aplikasinya. Setiap simpul di dalam pohon biner mempunyai paling banyak dua buah anak. Pohon ini memperhatikan anak-anaknya, anak-anaknya dibedakan antara anak kiri (*left child*) dan anak kanan (*right child*) sehingga pohon ini dikatakan pohon terurut.

10. Aplikasi Pohon Biner

Beberapa contoh aplikasi untuk pohon biner, yaitu :

a. Pohon Ekspresi



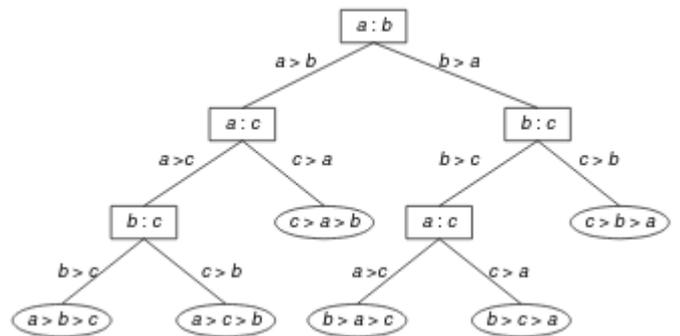
Gambar 7. Contoh Pohon Ekspresi

Sumber([http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20\(2013\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20(2013).pdf))

b. Pohon Keputusan

Pohon keputusan adalah suatu alat yang cukup populer dan efektif untuk mengklasifikasi dan prediksi. Pohon keputusan memiliki struktur yang mirip dengan *flowchart*. Setiap simpul melambangkan sebuah test. Setiap sisi

merepresentasikan hasil dari test tersebut. Sebagai contoh, di bawah ini adalah pohon keputusan untuk mengurutkan 3 buah elemen. Pada contoh dibawah atributnya adalah sebuah angka integer yang dilambangkan dengan a, b , dan c . Setiap sisi membandingkan dua buah integer tersebut dan pada akar didapatkan hasilnya.



Gambar 8. Contoh Pohon Keputusan

Sumber([http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20\(2013\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20(2013).pdf))

c. Kode Huffman

Rangkaian bit untuk string 'AAABBB' adalah 010000010100000101000001010000100100001001000010.

Panjang string tersebut adalah $6 \times 8 = 48$ bit (6 bytes). Dengan menggunakan tabel kekerapan (frekuensi) dan kode Huffman untuk string AAABBB diperoleh string 'AAABBB' panjangnya adalah 000110110110 yaitu 12 bit.

Simbol	Kekerapan	Peluang	Kode Huffman
A	3	3/7	0
B	1	1/7	110
C	2	2/7	10
D	1	1/7	111

Gambar 9. Tabel Peluang dan Kode Huffman

Sumber([http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20\(2013\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20(2013).pdf))

C. Persona 4

Persona 4 adalah sebuah game yang memiliki cerita pada suatu kota pedesaan yang bernama Inaba. Pada awal permainan, Pemain akan memulai kehidupan barunya sebagai siswa menengah atas. Kemudian saat kasus terjadi, pemain harus menyelamatkan temannya dengan melawan musuh yang disebut *shadow*. Setiap kali pemain menyelamatkan temannya, pemain akan berada pada latar belakang tempat yang berbeda dan pemain harus melawan *shadow* untuk melanjutkan ke cerita selanjutnya.

Ketika pemain berhasil menyelamatkan teman-temannya, teman yang diselamatkan tersebut akan masuk sebagai tim dan bersama dengan pemain untuk memecahkan kasus pembunuhan tersebut dan mencegah untuk tidak terbunuh. Teman-teman pemain bernama Chie Satonaka, Yukiko Amagi, Kanji Tatsumi, Rise Kujikawa, Teddie, dan Naoto Shirogane dan Yosuke Hanamura.

Kekuatan yang dimiliki oleh Pemain dinamakan persona. Persona dapat diperoleh dengan *Shuffle Time* atau melakukan *fusion*. Persona memiliki banyak jenisnya yang disebut arcana. Jenis arcana dapat seperti pada tabel 2.

Tabel 2. Jenis-Jenis Arcana Persona 4

Jenis Arcana Persona 4			
<i>Fool</i>	<i>Magician</i>	<i>Priestess</i>	<i>Empress</i>
<i>Emperor</i>	<i>Hierophant</i>	<i>Lovers</i>	<i>Chariot</i>
<i>Justice</i>	<i>Hermit</i>	<i>Fortune</i>	<i>Strength</i>
<i>Hanged Man</i>	<i>Death</i>	<i>Temperance</i>	<i>Devil</i>
<i>Tower</i>	<i>Star</i>	<i>Moon</i>	<i>Sun</i>
<i>Judgment</i>			



Gambar 10. Contoh Atribut pada Sebuah Persona

Sumber([http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20\(2013\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20(2013).pdf))

Setiap persona memiliki atributnya pada *skill*nya masing-masing yaitu *water*, *fire*, *light*, *darkness*, *wind*, *electric*, dan *physical*. Selain atribut, setiap persona memiliki *status* masing-masing. *Status* terdiri dari *Strength*, *Magic*, *Endurance*, *Agility*, dan *Luck*. *Strength* mempengaruhi efektivitas persona *physical skill*, *Magic* mempengaruhi efektivitas *magic skill* dan *magic defense* dari Persona tersebut. *Endurance* memengaruhi efektivitas persona dalam hal *physical defense*, *Agility* mempengaruhi kesempatan suatu serangan mengenai musuh, dan *Luck* mempengaruhi probabilitas dari suatu persona untuk mendapatkan *Critical Hit*. Setiap persona memiliki kelemahan dan ketahanan terhadap atributnya masing-masing.

Mekanisme untuk melawan *shadow* ada beberapa yaitu *attack*, *skill* (skill sesuai dengan persona yang sedang dipakai), *guard*, *analysis*, *item*, persona dan *escape*. *Attack* untuk menyerang musuh menggunakan senjata pemain. *Damage* yang dihasilkan dari *Attack* adalah *Physical Attack*. *Guard* untuk bertahan dari serangan musuh tetapi tetap menerima *damage* dengan skala yang kecil. *Skill* untuk menggunakan *skill* persona. Persona untuk memilih persona agar mendapat beberapa keuntungan dan *skill* persona tersebut.

III. LIMITASI PERMASALAHAN

Penulis membatasi permasalahan menjadi lebih sederhana dengan membagi persona sesuai dengan atributnya yaitu *water*, *fire*, *light*, *darkness*, *wind*, *electric*, dan *physical*. Selain itu,

penulis membatasi musuh yang dilawan oleh pemain hanya *boss*-nya saja yaitu *shadow* dari teman-teman yang akan diselamatkan oleh pemain karena banyak jenis musuh pada game ini. Penulis juga membatasi *boss* yang akan dilawan karena banyaknya teman yang akan diselamatkan. *Boss* yang akan dilawan antara lain *Shadow Yukiko*, *Shadow Yosuke*, dan *Shadow Chie*. Penulis tidak memakai *item*, tidak menggunakan *attack* untuk melawan musuh dan tidak menggunakan *guard* untuk mekanismenya. Sementara untuk *shadow* penulis menganggap bahwa musuh hanya melakukan penyerangan dan tidak melakukan pertahanan.

IV. IMPLEMENTASI PADA GAME PERSONA 4

Pada permainan ini terdapat dua fase yang akan dilewati pemain ketika melawan *boss*. Fase tersebut adalah fase *enemy* dan fase *pemain*. Kedua fase tersebut akan selalu bergantian selama pemain atau *boss* belum kehabisa HP (*Health Point*).

A. Fase Musuh

Pada fase ini, musuh dalam hal ini adalah *boss* akan melakukan gilirannya. Pada fase ini *boss* dapat melakukan serangan terhadap pemain ataupun melakukan pertahanan untuk memblokir serangan pemain pada saat fase *pemain*.

B. Fase Pemain

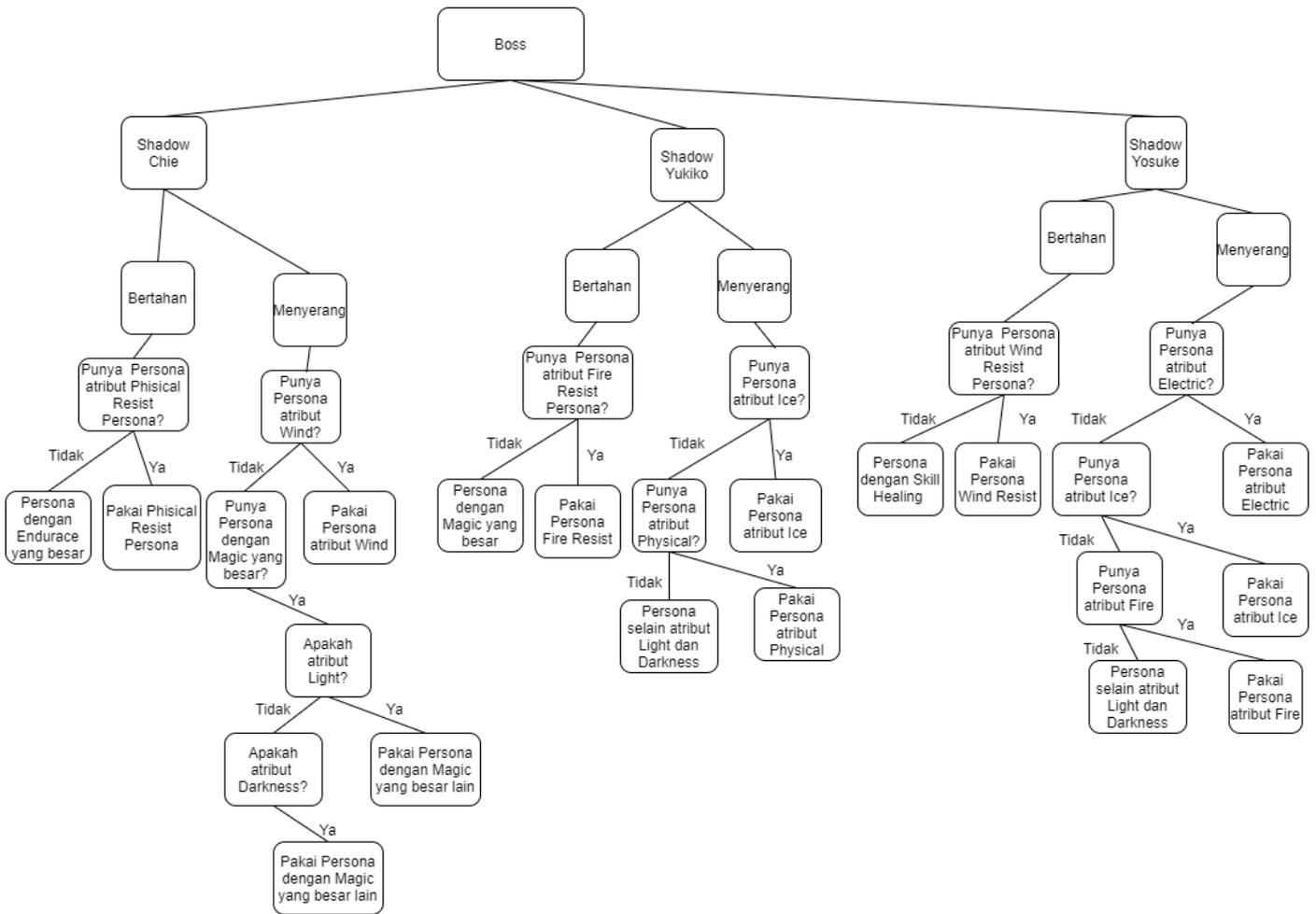
Pada saat fase ini pemain dapat melakukan gilirannya. Pada fase ini pemain dapat melakukan serangan, pertahanan, penggunaan *item*, penggunaan *skill*, analisis *enemy* untuk melihat status *enemy*, mengubah taktik permainan yaitu mengubah cara bermain tim, mengubah persona dan melakukan *escape* (tidak bisa dipakai ketika melawan *boss*). Melakukan *attack*, *guard*, *item*, dan *skill* dapat menyebabkan fase *pemain* selesai dan lanjut ke fase musuh. Melakukan *analysis*, perubahan persona dan mengubah *tactics* tidak akan membuat fase *pemain* selesai. Apabila pemain mengubah persona, persona tidak dapat diubah kembali sampai fase *pemain* selesai dan dapat diubah kembali ketika masuk ke fase *pemain* lagi.



Gambar 11. Tampilan Fase Pemain

Sumber(<https://assets.rockpapershotgun.com/images/2020/06/Persona-4-Golden-5.jpg>)

C. Analisis Penentuan Persona



Gambar 12. Pohon Keputusan Pemilihan Persona
Sumber (Dokumen Penulis)

Pada pohon di gambar 12 terdapat sebuah pilihan untuk menyerang dan bertahan sesuai dengan boss yang sedang dihadapi. Perbedaan status dan atribut pada setiap boss menjadi kunci dalam melakukan pohon keputusan ini. Pada pohon keputusan, ketika melawan boss pemain harus memilih akan bertahan atau menyerang.

Apabila pemain memutuskan untuk bertahan maka pemain memilih persona yang membuat pemain tidak akan mendapatkan damage ketika diserang oleh enemy atau mendapatkan damage yang kecil apabila tidak memenuhi kondisi tertentu. Persona yang dipilih berbeda sesuai dengan boss yang dihadapi. Pada Shadow Yosuke pemain memiliki prioritas untuk memilih persona dengan atribut Wind Resist karena serangan yang dilakukan oleh Shadow Yosuke adalah serangan dengan atribut wind apabila tidak memiliki persona tersebut maka pilih persona dengan skill healing karena damage yang diberikan shadow yosuke tidak terlalu besar. Pada Shadow Yukiko pemain diprioritaskan untuk memilih persona dengan atribut Fire Resist karena Shadow Yukiko sering menggunakan skill dengan atribut Fire dan apabila tidak memiliki persona tersebut maka pilih persona dengan Magic yang besar untuk mengurangi damage yang diberikan ketika enemy menggunakan skillnya. Ketika melawan Shadow Chie, diprioritaskan untuk

menggunakan persona dengan atribut Physical Resist karena shadow ini lebih sering menggunakan physical attack daripada skillnya. Apabila tidak memiliki persona tersebut maka pilih persona dengan Endurance yang besar untuk mengurangi physical damage yang diterima.

Apabila pemain memutuskan untuk menyerang musuh maka pemain memilih persona dengan atribut tertentu yang shadow lemah terhadapnya. Jika tidak mempunyai persona yang memiliki atribut tersebut maka pemain memilih persona dengan atribut yang tidak terblokir serangannya. Shadow Yosuke lemah terhadap Electric sehingga apabila pemain memiliki persona tersebut dapat membuat enemy tidak dapat menggunakan fasenya karena enemy terkena stun. Shadow Yukiko tidak lemah terhadap apapun tetapi dengan menggunakan atribut Ice dapat membuat shadow terkena damage yang besar. Apabila tidak memiliki persona dengan atribut Ice maka gunakan persona dengan atribut Physical karena shadow ini tidak memiliki status endurance yang besar sehingga memberikan damage yang besar terhadap musuh. Jika tidak memiliki persona dengan atribut Physical maka gunakan persona dengan atribut selain Light dan Darkness karena shadow ini kuat terhadap atribut tersebut. Shadow Chie memiliki kelemahan Wind sehingga apabila

persona pemain memiliki atribut Wind akan sangat menguntungkan bagi jalannya permainan. Jika tidak memilikinya, Gunakan persona dengan *status magic* yang besar untuk melawan *shadow* ini dan jika tidak ada maka gunakan persona dengan atribut selain *Light* dan *Darkness* karena *shadow* ini kuat terhadap atribut tersebut.

V. KESIMPULAN

Pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit. Pohon memiliki banyak aplikasi dalam kehidupan sehari-hari. Salah satu contoh aplikasinya adalah pohon keputusan. Suatu simpul dapat merepresentasikan sebuah perbandingan dan sebuah sisi dapat merepresentasikan sebuah kondisi.

Dalam makalah ini, penulis mengambil sebuah contoh pembuatan pohon keputusan dalam sebuah game Persona 4. Pohon keputusan tersebut mengenai keputusan dalam pemilihan persona untuk melawan musuh yang berada di game tersebut.

VI. UCAPAN TERIMA KASIH

Puji syukur pada Allah SWT atas kehendak-Nya penulis dapat menyelesaikan makalah ini tepat waktu.

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada seluruh pihak yang telah membantu penulis dalam menulis makalah ini. Tak lupa penulis mengucapkan terima kasih kepada Dosen Matematika Diskrit Ibu Nur Ulfa Maulidevi kelas K4 dan juga kepada dosen lain selaku tim pengajar mata kuliah IF2120.

REFERENSI

- [1] [http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20\(2013\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20(2013).pdf) diakses pada 4 Desember 2020 pada pukul 15.10.
- [2] [https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20\(2015\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf) diakses pada 4 Desember 2020 pada pukul 16.00.
- [3] https://megamitensei.fandom.com/wiki/Persona_4 diakses pada 5 Desember 2020 pada pukul 22.15.
- [4] <https://personacentral.com/p4g-ps-vita-sales-worldwide/#:~:text=According%20to%20Atlus%2C%20Persona%204,years%20after%20the%20game's%20release.> diakses pada 5 Desember 2020 pada pukul 22.20.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 5 Desember 2020



Nabil Nabighah
13519168