

Aplikasi *Decision Tree* dalam Pemilihan *Build* pada Permainan *Final Fantasy V*

Jevant Jedidia Augustine – 13520133¹
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
¹13520133@std.stei.itb.ac.id

Abstract—*Final Fantasy V* merupakan permainan *role-playing* yang dikembangkan dan dirilis oleh Square Enix. *Final Fantasy V* merupakan permainan kelima di dalam serial *Final Fantasy*. Fitur utama dari *Final Fantasy V* merupakan sistem *job* yang memperbolehkan pemain untuk mengoptimisasi *build* dari karakter yang dimainkan oleh pemain. Makalah ini akan membahas penentuan *build* yang tepat berdasarkan situasi yang sedang dihadapi oleh pemain dengan menggunakan *decision tree*.

Keywords—*command*, *decision tree*, *Final Fantasy V*, *job*

I. PENDAHULUAN

Final Fantasy V adalah permainan kelima dari serial *Final Fantasy* yang merupakan salah satu serial JRPG terbesar di dunia. *Final Fantasy V* pertama kali dirilis oleh Square Enix pada tahun 1992 untuk Super Nintendo Entertainment System (SNES). *Final Fantasy V* kemudian mendapatkan perilisasi untuk GBA, PlayStation, *mobile* (iOS dan Android), dan PC pada tahun yang mendatang.

Final Fantasy V menceritakan petualangan sebuah kelompok yang bernama *Warrior of Light*. Kelompok tersebut terdiri dari 4 karakter, yaitu Bartz, Lenna, Galuf, dan Faris (Galuf akan digantikan oleh Krile nantinya). *Warrior of Light* memiliki tugas untuk melindungi kristal-kristal dari kehancuran. Figur dibalik kehancuran kristal-kristal tersebut adalah Exdeath yang memiliki rencana untuk melepaskan dirinya dari kurungan kristal-kristal tersebut dan mendapatkan kekuatan *Void* untuk menghancurkan dunia. Rencana Exdeath dalam menghancurkan kristal-kristal yang memenjarakan dirinya berhasil dilakukan oleh Exdeath sehingga *Warrior of Light* memiliki tugas yang baru yaitu mengalahkan Exdeath sebelum ia berhasil menghancurkan dunia dengan kekuatan *Void*.

Fitur utama dari *Final Fantasy V* merupakan *job system*. Fitur tersebut memperbolehkan pemain untuk mengoptimisasi *build* dari setiap karakter. Terdapat 22 (26 untuk versi GBA) *job* yang dapat dipilih oleh pemain dan setiap *job* memiliki kemampuan dan atribut yang unik. Setiap kali pemain menyelesaikan suatu *battle* dengan mengalahkan semua musuh, karakter yang memiliki *job* akan mendapatkan *Ability Point (AP)*. Bila berhasil mengumpulkan AP dengan jumlah tertentu, *job* dari karakter akan mengalami kenaikan level. Setiap kali *job* dari karakter mengalami kenaikan level, *command* dan atribut yang dapat diakses pada *job* karakter tersebut akan bertambah banyak.

Pemilihan dari *job/build* untuk tiap karakter sangatlah bergantung dengan keadaan dari permainan saat itu. Salah satu *job* bisa saja kurang bermanfaat dalam keadaan tertentu tetapi dapat menjadi sangat berguna dalam keadaan yang lain. Proses pemilihan *build* yang tepat dan efektif dapat divisualisasikan dengan *decision tree*.



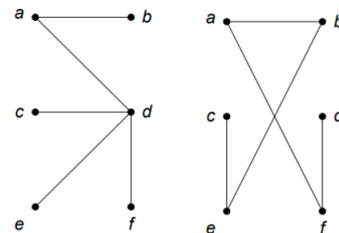
Gambar 1.1 Logo Resmi *Final Fantasy V*

(Sumber : https://square-enix-games.com/en_GB/home/final-fantasy-v-now-available-pre-purchase-steam)

II. LANDASAN TEORI

A. Pohon

Dalam teori graf, pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit.



Gambar 2.1 Contoh dari Pohon

(Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag1.pdf>)

Pohon memiliki sifat-sifat atau teorema sebagai berikut.

Misalkan $G = (V, E)$ adalah graf tak-berarah sederhana dengan jumlah simpul n . Maka, semua pernyataan di bawah ini adalah benar:

- G adalah pohon.
- Setiap pasang simpul di dalam G terhubung dengan

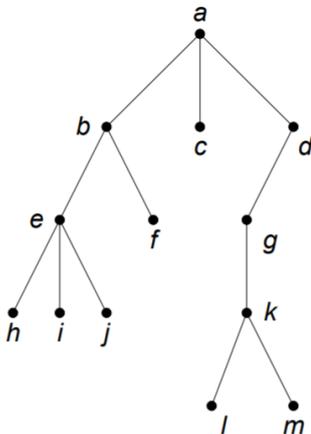
lintasan tunggal.

- G terhubung dan memiliki $m = n - 1$ buah sisi.
- G tidak mengandung sirkuit dan memiliki $m = n - 1$ buah sisi.
- G tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit.
- G terhubung dan semua sisinya adalah jembatan.

Pohon dapat diklasifikasi menjadi sebagai berikut.

1. Pohon Berakar

Pohon berakar merupakan pohon yang satu buah simpulnya diperlakukan sebagai akar dan sisi-sisinya diberi arah sehingga menjadi graf berarah.



Gambar 2.2 Pohon Berakar

(Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag2.pdf>)

Terminologi pada pohon berakar adalah sebagai berikut.

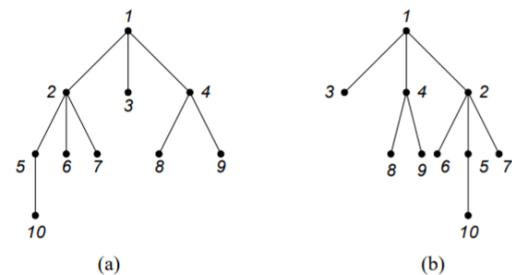
- a. Anak (*child/children*) dan Orangtua(*parent*)
Jika sebuah simpul A memiliki cabang B dengan tingkatan yang lebih rendah dibandingkan dengan tingkatan simpul A, maka simpul A merupakan orangtua dari simpul B dan simpul B merupakan anak dari simpul A.
- b. Lintasan (*path*)
Bila dilihat pada gambar 2.2, lintasan dari a ke m adalah a, d, g, k, m dengan panjang lintasan sebesar 4.
- c. Saudara Kandung (*sibling*)
Sebuah simpul A merupakan saudara kandung simpul B jika dan hanya jika kedua simpul memiliki orangtua yang sama.
- d. Upapohon (*subtree*)
Pohon yang merupakan bagian dari pohon yang lebih besar.
- e. Derajat (*degree*)
Derajat sebuah simpul merupakan jumlah anak atau jumlah upapohon dari simpul tersebut.

Derajat maksimum dari semua simpul merupakan derajat dari pohon itu sendiri.

- f. Daun (*leaf*)
Simpul yang berderajat nol atau tidak memiliki anak.
- g. Simpul Dalam (*internal node*)
Simpul yang mempunyai anak disebut simpul dalam.
- h. Aras (*level*) atau Tingkat
Tingkatan yang dihitung dari akar, dimana akar memiliki tingkat 0, anak dari akar memiliki tingkat 1, anak dari anak dari akar memiliki tingkat 2, dan seterusnya hingga tercapai sebuah daun.
- i. Tinggi (*height*) dan Kedalaman (*depth*)
Aras atau tingkat tertinggi dari suatu pohon.

2. Pohon Terurut

Pohon terurut merupakan pohon yang urutan dari anak-anaknya penting.



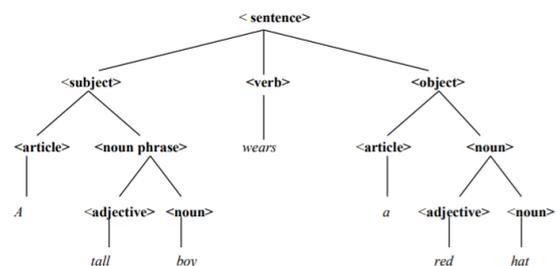
Gambar 2.3 Pohon Terurut

(Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag2.pdf>)

3. Pohon n -ary

Pohon n -ary merupakan pohon berakar yang setiap simpul cabangnya memiliki paling banyak n buah anak.



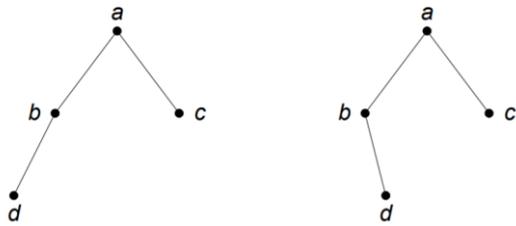
Gambar 2.4 Pohon n -ary

(Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag2.pdf>)

4. Pohon Biner

Pohon biner merupakan pohon n -ary dengan $n = 2$, sehingga setiap simpul dalam pohon terdapat paling banyak 2 buah anak yang dibedakan menjadi anak kanan dan anak kiri.



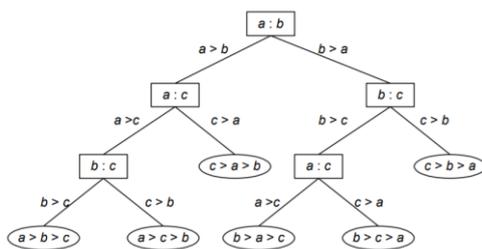
Gambar 2.5 Pohon Biner

(Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag2.pdf>)

B. Pohon Keputusan

Pohon keputusan (*decision tree*) merupakan salah satu pengaplikasian dari pohon biner. Pohon keputusan digunakan untuk mengambil suatu keputusan mengenai suatu masalah tertentu.



Gambar 2.6 Pohon Keputusan

(Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag2.pdf>)

C. Sistem Job/Kelas pada *Final Fantasy V*

Fitur utama dari *Final Fantasy V* adalah sistem *job* yang memperbolehkan pemain untuk mengoptimisasi karakter dengan *job* yang tersedia. Pemain dapat mendapatkan *job* secara berkelompok setelah mencapai titik tertentu pada permainan. *Freelancer* didapatkan pada awal permainan; *Knight*, *Monk*, *Thief*, *Black Mage*, *White Mage*, dan *Blue Mage* didapatkan setelah *Wind Crystal* hancur; *Red Mage*, *Time Mage*, *Summoner*, *Berserker*, dan *Mystic Knight* didapatkan setelah *Water Crystal* hancur; *Beastmaster*, *Geomancer*, *Ninja*, *Ranger*, dan *Bard* didapatkan setelah *Fire Crystal* hancur; dan *Dragoon*, *Dancer*, *Samurai*, dan *Chemist* didapatkan setelah *Earth Crystal* hancur. *Mime* akan didapatkan nantinya setelah pemain melanjutkan permainan.



Gambar 2.7 Tampilan Pemilihan *Job*

(Sumber : <https://www.ign.com/games/final-fantasy-v-advance>)

Terdapat 22 kelas yang dapat digunakan oleh pemain di Final Fantasy V versi SNES, yaitu:

1. *Freelancer*

Kelas yang pertama kali diberikan kepada pemain. Tidak memiliki kemampuan yang khusus bila dibandingkan dengan kelas yang lain, akan tetapi *Freelancer* memiliki kebebasan untuk menggunakan senjata dan pakaian apapun. *Freelancer* juga memiliki kebebasan untuk memilih *command* dari kelas apapun asalkan kelas tersebut telah dikuasai (level dari kelas tersebut sudah maksimum). *Freelancer* juga akan mendapatkan atribut dan stats bonus dari kelas yang sudah dikuasai.

2. *Knight*

Memiliki atribut *physical* yang tinggi sehingga dapat menjadi penyerang *physical* yang cukup efektif. *Knight* mendapatkan akses ke kemampuan *Cover*, dimana *Knight* akan melindungi karakter lain apabila *HP* mereka sedang kritis, dan *command Guard*, yang menyebabkan *Knight* menjadi kebal dari semua serangan *physical*, sehingga *Knight* juga dapat menjadi kelas yang defensif secara *physical*.

3. *Monk*

Kelas yang mendapatkan bonus apabila tidak menggunakan senjata sama sekali (bertangan kosong). Bila suatu karakter tidak menggunakan senjata sama sekali, maka karakter tersebut akan melakukan serangan *physical* sebanyak dua kali dalam sekali serangan. *Monk* juga mendapatkan *skill Counter* yang menyebabkan *Monk* untuk melakukan serangan balik apabila *Monk* mendapatkan serangan *physical* dari musuh.

4. *Thief*

Kelas tercepat dari semua kelas yang ada sehingga pada kebanyakan kasus, akan menjadi yang pertama menyerang di dalam suatu *battle*. *Command* utama dari *Thief* adalah *Steal* dan *Mug*. Kedua *command* tersebut menyebabkan *Thief* untuk mencuri barang yang ada pada musuh. *Mug* juga menyebabkan *Thief* untuk menyerang musuh secara sekaligus.

5. *Black Mage*

Kelas *magical* menyerang yang dapat mengakses *Black Magic* untuk menyerang musuh. Kebanyakan serangan yang berasal dari *Black Magic* adalah serangan elemental, sehingga *Black Mage* dapat mengeksploitasi kelemahan elemental musuh.



Gambar 2.8 Tampilan Kelas *Black Mage* pada *Final*

Fantasy V

(Sumber :

[https://finalfantasy.fandom.com/wiki/Black_Mage_\(Final_Fantasy_V\)](https://finalfantasy.fandom.com/wiki/Black_Mage_(Final_Fantasy_V)))

6. White Mage

Kelas *magical* defensif yang dapat mengakses *White Magic*. *White Mage* membantu *party* di dalam *battle* dengan menyembuhkan *party* atau memberikan status dengan efek positif kepada *party*. *White Mage* hanya memiliki satu *spell* menyerang yaitu *Holy*.

7. Blue Mage

Kelas *magical* utilitas yang dapat mengakses *Blue Magic*, yaitu *spell* dengan kemampuan menyerang dan juga defensif. *Blue Mage* hanya dapat menggunakan suatu *spell* apabila *Blue Mage* sudah pernah melihat *spell* tersebut digunakan oleh musuh, sehingga semua *spell* yang digunakan oleh *Blue Mage* pada intinya merupakan *spell* dari musuh.

8. Red Mage

Kelas hibrida antara *Black Mage* dan *White Mage*. Dapat mengakses *Red Magic*, yaitu gabungan dari beberapa *Black Magic* dan *White Magic* yang memiliki level rendah. Saat kelas *Red Mage* mencapai level maksimum, *Red Mage* akan mendapat akses ke *Dualcast*. Dengan *Dualcast*, *Red Mage* dapat melakukan *Red Magic*, *Blue Magic*, *White Magic*, atau *Summon* sebanyak 2 kali berturut-turut dalam satu giliran.

9. Time Mage

Kelas *magical* defensif yang dapat mengakses *Time Magic*. *Spell* defensif yang dapat diakses oleh *Time Mage* bermanfaat untuk memberikan status efek positif kepada *party* atau memberikan status efek negatif kepada musuh. Status efek yang diberikan oleh *Time Mage* berbasis waktu yang dapat memengaruhi urutan giliran dari musuh dan *party* di dalam *battle*.

10. Summoner

Kelas *magical* yang dapat memanggil *esper* untuk membantu *Summoner* di dalam *battle* dengan *command Summon*. Serangan dari *Summon* yang menyerang adalah serangan area, sehingga *Summoner* dapat menyerang banyak musuh secara sekaligus. *Summoner* juga memiliki beberapa *Summon* yang defensif seperti *Carbuncle* yang memantulkan seluruh serangan *magical* ke *party* kembali ke siapapun yang menyerang.

11. Berserker

Kelas yang selalu memiliki status *berserk* di dalam *battle*. *Berserk* menyebabkan karakter menyerang dengan 50% *damage* lebih, akan tetapi pemain tidak dapat mengontrol aksi maupun siapa yang akan diserang oleh karakter tersebut, sehingga *Berserker* akan terus menerus menyerang musuh secara acak hingga *battle* berakhir.

12. Mystic Knight

Kelas yang dapat mengaruniai senjata mereka dengan beberapa *spell* dari *Black Magic*. Hal tersebut menyebabkan serangan *physical Mystic Knight* memiliki properti yang sama dengan *spell* dari *Black Magic* yang dikaruniai. Contohnya senjata yang dikaruniai *Fira* akan memiliki elemen api (*Fira*) saat menyerang musuh.



Gambar 3.9 Tampilan Kelas *Mystic Knight* pada *Final Fantasy V*

(Sumber :

[https://finalfantasy.fandom.com/wiki/Mystic_Knight_\(Final_Fantasy_V\)](https://finalfantasy.fandom.com/wiki/Mystic_Knight_(Final_Fantasy_V)))

13. Mime

Kelas yang mirip dengan *Freelancer* akan tetapi lebih fokus kepada serangan *magical*. *Mime* tidak memiliki kebebasan pemilihan senjata dan pakaian yang dimiliki oleh *Freelancer*. *Command* utama *Mime* adalah *Mimic*. *Mimic* menyebabkan *Mime* untuk menyalin aksi terakhir yang dilakukan oleh *party* pemain.

14. Beastmaster

Kelas yang dapat menangkap musuh dan melepaskannya di dalam *battle* untuk menyerang musuh dengan *command Catch* dan *Release*. *Beastmaster* juga dapat mengontrol musuh yang ada di dalam *battle* dengan *command Control*. *Skill* dari musuh yang dikontrol oleh *Beastmaster* dapat diakses dan digunakan untuk membantu *Beastmaster* di dalam *battle*.

15. Geomancer

Kelas utilitas yang menggunakan kekuatan alam untuk menyerang musuh. Dengan *command Gaia*, *Geomancer* dapat menyerang musuh dengan serangan berdasarkan lokasi dari *battle*. *Geomancer* juga memiliki akses ke *skill Find Pits* dan *Light Step* yang dapat melindungi *party* dari tempat-tempat yang berbahaya (seperti saat berjalan di atas lava) sehingga *party* tetap aman saat berjelajah.

16. Ninja

Kelas yang dapat menggunakan 2 senjata secara sekaligus sehingga serangan *physical Ninja* akan terjadi sebanyak 2 kali dalam satu giliran. *Ninja* juga mendapatkan akses ke *command Throw* yang memperbolehkan *Ninja* untuk melempar senjata atau

gulungan elemental yang ada di tas *party* kepada musuh untuk menyerang musuh.

17. *Ranger*

Satu dari dua kelas yang dapat menggunakan busur sebagai senjata. Dengan menggunakan busur, *Ranger* dapat menyerang musuh dari barisan belakang maupun depan dengan damage penuh, walaupun akurasi serangan berkurang menjadi 70%. *Command* utama dari *Ranger* adalah *Rapid Fire* yang menyebabkan *Ranger* untuk menyerang 4 kali dalam satu giliran.

18. *Bard*

Kelas utilitas yang membantu *party* dengan *command Sing*. *Sing* memperbolehkan *Bard* untuk beryanyi sepanjang *battle*. Nyanyian *Bard* dapat memberikan efek positif kepada *party* atau efek negatif kepada musuh. Selama berlajunya *battle*, *Bard* akan terus beryanyi hingga *Bard* mendapatkan serangan, terbunuh, atau mendapatkan efek negatif yaitu *petrified* atau *silence*.

19. *Dragoon*

Salah satu kelas yang dapat menggunakan *Spear* sebagai senjata. *Command* utama dari *Dragoon* adalah *Jump* yang menyebabkan *Dragoon* untuk melompat ke atas sehingga *Dragoon* akan menghilang dari *battle*. *Dragoon* akan menghindari semua serangan saat *Dragoon* berada di udara. Saat *Dragoon* mendapatkan gilirannya lagi, *Dragoon* akan turun dan menyerang musuh dari udara, lalu *Dragoon* akan kembali ke posisi semula saat sebelum melakukan *command Jump*.



Gambar 3.10 Tampilan Kelas *Dragoon* pada *Final Fantasy V*
(Sumber : [https://finalfantasy.fandom.com/wiki/Dragoon_\(Final_Fantasy_V\)](https://finalfantasy.fandom.com/wiki/Dragoon_(Final_Fantasy_V)))

20. *Dancer*

Kelas yang dapat mengakses *command Dance*. Dengan *Dance*, *Dancer* akan melakukan satu dari 4 aksi berikut, memberikan status *confuse* kepada musuh, menyerap *HP* musuh, menyerap *MP* musuh, atau menyerang dengan *damage* sebesar 4 kali lipat *damage* biasa.

21. *Samurai*

Kelas menyerang *physical* yang dapat menyerang dengan serangan *physical* atau dengan menggunakan *command*. Pada level maksimum, *Samurai* mendapatkan akses ke 3

command menyerang, yaitu: *Mineuchi*, serangan yang dapat memberikan efek *paralyze* kepada musuh yang diserang, *Zeninage*, serangan area dengan *damage* yang bergantung dengan jumlah uang yang dimiliki *party*, dan *Iainuki*, serangan area yang memiliki kemungkinan untuk membunuh semua musuh secara instan.

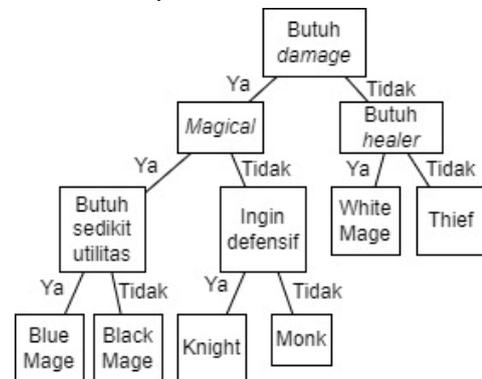
22. *Chemist*

Kelas utilitas yang mendapatkan akses ke *command Mix*. *Mix* memperbolehkan *Chemist* untuk menggabungkan dua buah *item* yang berada di tas untuk menyerang musuh, memberikan efek positif kepada *party*, atau memberikan efek negatif kepada musuh. Selain itu, dengan *skill Pharmacology*, saat mengonsumsi atau menggunakan *item* dari tas, efek dari *item* yang digunakan menjadi 2 kali lipat efek biasa.

III. APLIKASI *DECISION TREE* DALAM PEMILIHAN *JOB*

Pemilihan dari *Job* atau kelas dapat dibagi menjadi 4 *Decision Tree* berdasarkan *Crystal* yang telah hancur, yaitu setelah *Wind Crystal* hancur, setelah *Water Crystal* hancur, setelah *Fire Crystal* hancur, dan setelah *Earth Crystal* hancur.

A. Setelah *Wind Crystal* hancur



Gambar 3.1 *Decision Tree* Setelah *Wind Crystal* Hancur

Pemilihan kelas setelah *Wind Crystal* hancur masih relatif sederhana dibandingkan pemilihan kelas yang lain. Pada gambar 3.1, dapat dilihat bahwa kelas *Freelancer* tidak ada pada pohon, Hal tersebut dikarenakan kelas *Freelancer* kurang berguna pada awal-awal permainan karena kelas tersebut membutuhkan suatu karakter untuk menguasai kelas yang relatif banyak (kurang lebih $\frac{3}{4}$ dari kelas total) agar *Freelancer* dapat menjadi berguna. Perlu diperhatikan juga utilitas yang dimaksud pada pemilihan adalah kemampuan *supportive* yaitu, pemberian status negatif kepada musuh, pemberian status positif kepada *party*, atau memberikan *heal* kepada *party*.

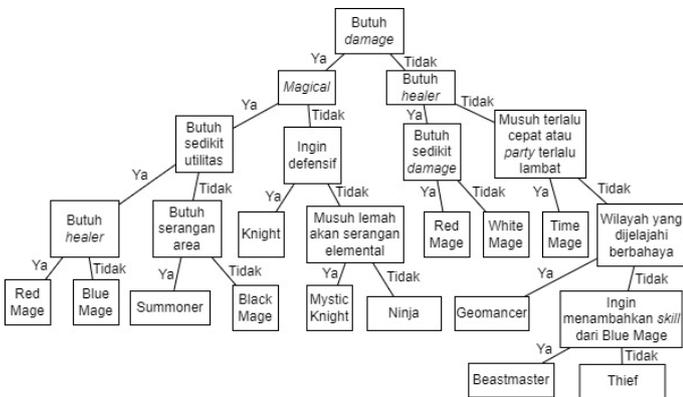
B. Setelah Water Crystal hancur



Gambar 3.2 Decision Tree Setelah Water Crystal hancur

Pohon keputusan setelah *Water Crystal* hancur merupakan pohon keputusan pada gambar 3.1 dengan beberapa tambahan simpul dan daun. Pada gambar 3.2, kelas *Monk* tidak berada di pohon karena bila dibandingkan dengan kelas yang lain, khususnya *Berserker*, *Monk* kurang berguna. *Berserker* dapat melakukan apa yang dilakukan oleh *Monk* dengan lebih baik, kecuali dalam beberapa hal akan tetapi hal tersebut tidak cukup signifikan untuk diperhitungkan.

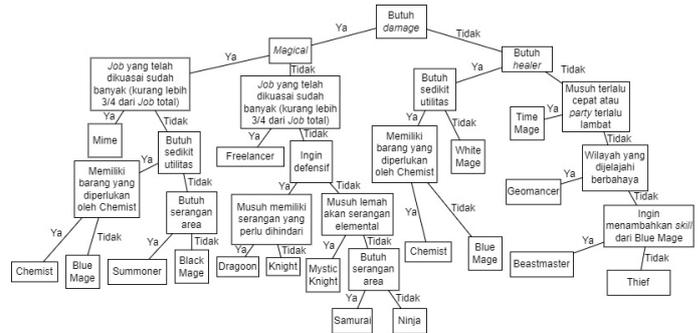
C. Setelah Fire Crystal hancur



Gambar 3.3 Decision Tree Setelah Fire Crystal hancur

Pada gambar 3.3 dapat dilihat bahwa pohon keputusan untuk bagian setelah *Fire Crystal* hancur merupakan modifikasi dari pohon keputusan pada gambar 3.2. *Ninja* menggantikan posisi *Berserker* dengan alasan yang sama sebagaimana *Berserker* menggantikan posisi *Monk*. Penambahan *Beastmaster* pada pohon memiliki alasan yang unik. Untuk kelas *Blue Mage*, untuk mendapatkan suatu *spell Blue Magic*, *Blue Mage* harus “melihat” *spell* tersebut dilakukan dan *spell* yang dapat dipelajari dan dilakukan oleh *Blue Mage* merupakan *spell* musuh. Dengan *command Control*, *Beastmaster* dapat memaksa musuh untuk melakukan *spell* yang dapat dipelajari oleh *Blue Mage*, maka dari itu *Beastmaster* berguna untuk menambahkan *spell* dari *Blue Mage* dengan *command Control*-nya. Kelas *Ranger* dan *Bard* tidak terdapat pada pohon karena terdapat kelas lain yang dapat memenuhi peran kedua kelas tersebut dengan lebih baik. *Ninja* merupakan alternatif dari *Ranger* yang lebih baik, sedangkan alternatif yang lebih baik dari *Bard* adalah *Time Mage*.

D. Setelah Earth Crystal hancur



Gambar 3.4 Decision Tree Setelah Earth Crystal Hancur

Gambar 3.4 merupakan pohon keputusan setelah *Earth Crystal* hancur atau setelah pemain mendapatkan semua 22 kelas yang ada pada permainan. Dengan didaptnya semua kelas yang ada pada permainan, kelas *Freelancer* dan *Mime*, yang bergantung dengan jumlah kelas yang telah dikuasai karakter, menjadi salah satu pilihan kelas yang sangat efektif. Setelah *Earth Crystal* hancur, pemain dapat mendapatkan seluruh *spell* dari *Blue Mage* sehingga *Blue Mage* menggantikan posisi *Red Mage* di pohon. *Chemist* juga menggantikan posisi *Red Mage* dan memiliki peran yang mirip dengan *Blue Mage*, bahkan lebih kuat daripada *Blue Mage*. Perbedaan antara kedua kelas tersebut adalah *Chemist* membutuhkan beberapa *item* supaya *command Mix* dari *Chemist* dapat digunakan secara efektif sementara *Blue Mage* hanya perlu mempelajari *spell* dari musuh. *Dragoon* memiliki *command Jump* yang menyebabkan *Dragoon* untuk lompat ke udara dan menghindari serangan musuh. Serangan yang perlu dihindari biasanya dapat membunuh suatu karakter secara instan sehingga *Dragoon* dengan *command Jump*-nya menjadi salah satu cara yang ampuh untuk menghindari serangan tersebut. Kelas *Dancer* tidak terdapat pada pohon karena kelas tersebut tidak konsisten dalam perannya. *Command Dance* yang dimiliki oleh *Dancer* hanya memiliki peluang $\frac{1}{4}$ untuk melakukan aksi yang diinginkan.

Bila diperhatikan, kelas *Thief* selalu ada pada keempat pohon keputusan yang menyebabkan *Thief* menjadi salah satu kelas yang relevan dari awal permainan hingga akhir permainan. Hal tersebut dikarenakan musuh yang ditemukan pada awal, tengah, dan akhir permainan kebanyakan akan memiliki barang yang dapat dicuri *Thief*, sehingga kemampuan *Thief* untuk mencuri barang dari musuh akan selalu bermanfaat.

V. SIMPULAN

Pemilihan *build* pada *Final Fantasy V* bergantung kepada keadaan yang sedang dihadapi. Suatu kelas bisa saja bermanfaat pada keadaan tertentu dan tidak bermanfaat pada keadaan lainnya. Walaupun demikian, bila ingin dipilih kelas yang paling efektif pada keadaan tertentu, maka akan ada beberapa kelas yang tidak akan dipilih apapun keadaannya. Penyebab dari hal tersebut adalah kelebihan dari kelas tersebut tidak cukup signifikan atau terdapat alternatif kelas lain yang lebih baik dan efektif sehingga disarankan untuk memilih kelas yang jauh lebih efektif.

VI. UCAPAN TERIMA KASIH

Saya mengucapkan terima kasih kepada Tuhan Yang Maha Esa karena atas berkat dan karunia-Nya lah saya dapat menyelesaikan makalah “Aplikasi *Decision Tree* dalam Pemilihan *Build* pada Permainan *Final Fantasy V*” dengan tepat waktu. Saya juga mengucapkan terima kasih kepada orang tua saya yang selalu mendukung saya selama masa perkuliahan saya di ITB. Terakhir, saya ingin mengucapkan terima kasih kepada ibu Nur Ulfa Maulidevi yang telah membimbing saya selama satu semester sebagai dosen pengampu IF2120 Matematika Diskrit Kelas 3.

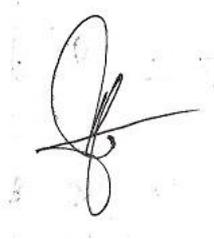
REFERENCES

- [1] Munir, Rinaldi. 2021. Pohon (Bagian 1). <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag1.pdf>, diakses 7 Desember 2021 pukul 19.32 WIB
- [2] Munir, Rinaldi. 2021. Pohon (Bagian 2). <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag2.pdf>, diakses 7 Desember 2021 pukul 20.12 WIB
- [3] https://finalfantasy.fandom.com/wiki/Final_Fantasy_V_jobs, diakses 10 Desember 2021 pukul 20.04 WIB
- [4] <https://steamcommunity.com/sharedfiles/filedetails/?id=524231112>, diakses 13 Desember 2021 pukul 20.25 WIB

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, 13 Desember 2020



Jevant Jedia Augustine
13520133