

Implementasi Pohon Keputusan untuk Menganalisa Desain Sistem *Battle* pada Game *Brightsouls*

Kevin Erdiza Yogatama, 13515016¹
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13515016@std.stei.itb.ac.id

Abstract—*Brightsouls* adalah sebuah game yang menjadi tugas besar peserta kuliah IF 2110 Algoritma & Struktur Data Semester I tahun 2016 yang memiliki sistem *Battle* berupa *turn-based* dimana setiap *turn*-nya, *player* diberi kesempatan untuk memilih set aksi untuk melawan set aksi musuh. Makalah ini akan menunjukkan bahwa desain sistem *Battle* yang berbeda dengan menunjukkan pohon keputusan yang terbentuk akan membuat satu sistem menjadi lebih baik dari sistem yang lainnya.

Keywords—*Brightsouls*, *player*, aksi, *damage*.

I. PENDAHULUAN

Dalam mendesain sebuah *game*, banyak aspek yang perlu diperhatikan. Hanya dengan satu aspek mekanik *game* yang tidak seimbang bisa menjadikan *game* secara keseluruhan menjadi tidak bagus. Tidak terkecuali dengan *game* yang menjadi tugas besar peserta kuliah IF2110, *Brightsouls*.

Pada tugas besar ini, spesifikasi *game Brightsouls* sudah jelas mendeskripsikan sebagian besar mekanik permainan dari awal sampai akhir, namun kebanyakan tim yang mengerjakan tugas tersebut mengklaim bahwa game mereka sulit untuk diselesaikan dan hanya pernah menyelesaikan dengan *cheat* yang mereka telah buat sendiri. Hal ini bisa terjadi karena spesifikasi memang belum lengkap dalam menjelaskan aspek *game Brightsouls* terutama aspek *rule*.

Rule merupakan salah satu aspek dasar dalam mendesain sebuah *game* [1] dan pada makalah ini, kita akan fokus terhadap satu *rule* yang tidak terdefinisi oleh spesifikasi tugas besar ini, yaitu *rule* yang mengatur bagaimana *damage* dikalkulasi saat sistem *Battle* dan menunjukkan dengan pohon keputusan bahwa dengan menggunakan *rule* yang berbeda bisa menghasilkan cara bermain yang berbeda walaupun menggunakan mekanik permainan yang sama.

II. BRIGHTSOULS



Gambar 1. Logo game *Brightsouls*

Brightsouls merupakan sebuah proyek game yang menjadi tugas besar peserta kuliah IF 2110 Algoritma & Struktur Data Semester I tahun 2016. *Game* ini merupakan permainan dengan *genre text-based RPG* yang dimainkan dengan cara memasukkan perintah melalui *command-line interface*.

Tujuan utama permainan *Brightsouls* ini adalah memainkan satu karakter untuk menjelajahi *map* permainan dan memiliki misi utama yaitu mengalahkan suatu *final boss* yang berada di suatu lokasi di *map*. Setiap *map* akan muncul musuh yang bisa dilawan oleh karakter agar bisa menaikkan *level* karakter tersebut dimana hal itu dibutuhkan agar karakter bisa mengalahkan *final boss*.

Untuk melawan musuh dan mengalahkannya, akan ada suatu sistem *battle* yang akan dijalani. *Battle* pada *game* ini dibagi menjadi ronde dan setiap rondonya akan ditampilkan set aksi musuh dimana *player* akan memasukkan set aksi yang serupa untuk menangkal set aksi tersebut sesuai dengan *rule* yang telah ditentukan. Yang membuat *battle* menjadi menarik adalah set aksi musuh akan ditutup sebagian sehingga *player* tidak bisa selalu menangkal seluruh aksi musuh.

Selain itu, ada sistem atribut, *level*, dan *skill* yang dimiliki oleh karakter. Sistem atribut yang dimiliki

karakter yaitu merupakan atribut *battle* yang terdiri dari *HP*, *STR*, *DEF* dimana atribut ini yang menjadi faktor kemenangan *battle* dan bagaimana *damage* yang diberikan dan diterima di *battle* dikalkulasi. Kemudian, sistem *level* dan *skill* yang dimiliki karakter akan memberi keuntungan pada karakter untuk bisa mengalahkan lebih banyak musuh. Setiap kenaikan *level* akan memberikan keuntungan pada player yang bisa berupa penambahan atribut karakter atau *skillpoint* yang digunakan untuk mengaktifkan *skill* yang bisa menambahkan efek positif untuk karakter saat *battle*.

Di makalah ini, kita hanya akan fokus pada sistem *battle* dan sistem atribut dimana kedua hal ini akan dijelaskan lebih detail pada bagian keempat makalah ini.

III. LANDASAN TEORI

Landasan teori yang digunakan adalah struktur graf berupa pohon, yang lebih khusus lagi berupa pohon keputusan.

A. Graf

Graf didefinisikan sebagai pasangan himpunan (V,E) , yang dalam hal ini V adalah himpunan tidak-kosong dari simpul-simpul dan E adalah himpunan sisi yang menghubungkan sepasang simpul. Istilah simpul dan sisi juga akan dipakai pada bagian penjelasan pohon dan pohon keputusan.

B. Pohon

Pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit [2]. Pada kebanyakan pengaplikasian pohon, simpul tertentu pada pohon tersebut diperlakukan sebagai akar dimana hal ini adalah yang disebut dengan pohon berakar. Berikut ini adalah terminologi yang berhubungan dengan pohon berakar

a) Anak dan Orangtua

Suatu simpul x adalah anak dari suatu simpul y jika ada sisi yang menghubungkan x dan y . Dalam kasus ini, simpul y adalah orang tua dari simpul x .

b) Lintasan

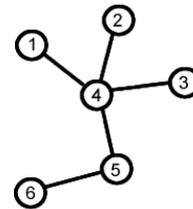
Lintasan dari simpul v_i ke simpul v_k adalah runtunan simpul v_1, v_2, \dots, v_k sedemikian sehingga v_i adalah orangtua dari v_{i+1} untuk $1 \leq i < k$.

c) Internal node dan Daun/Terminal

Internal node adalah seluruh simpul pada pohon yang memiliki anak, dan yang bukan termasuk *internal node* adalah daun atau *terminal*.

d) Saudara Kandung

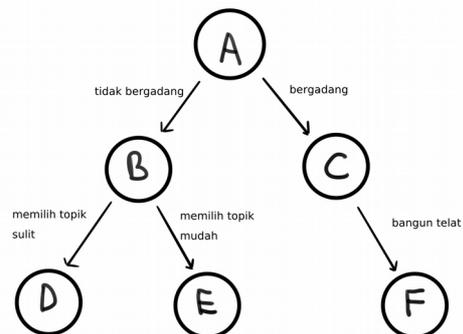
Suatu simpul x adalah saudara kandung dari suatu simpul y jika x dan y memiliki orang tua yang sama.



Gambar 2. Contoh struktur pohon

C. Pohon Keputusan

Pohon keputusan adalah pohon berakar yang digunakan untuk memodelkan persoalan yang terdiri dari serangkaian keputusan yang mengarah ke solusi atau hasil dari keputusan tersebut.

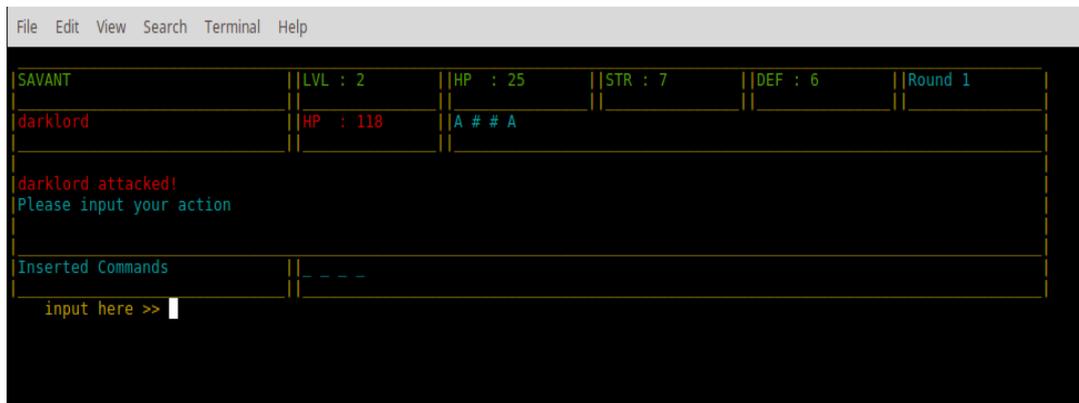


Gambar 3. Contoh pohon keputusan

IV. SISTEM BATTLE GAME BRIGHTSOULS

Seperti yang sudah dijelaskan di bagian sebelumnya, *battle* terbagi menjadi sejumlah ronde dimana setiap rondonya, *player* diberi kesempatan untuk menentukan set aksi untuk karakternya melawan set aksi musuh yang sudah ditampilkan di layar namun sebagian ditutup dan tidak diketahui *player*. Komponen yang berhubungan dengan *battle* dan yang telah didefinisikan oleh spesifikasi adalah jumlah ronde setiap *battle*, atribut karakter yang dimainkan dan musuh, dan jenis serta jumlah aksi yang bisa digunakan setiap rondonya.

Setiap *battle* terdiri dari maksimum 10 ronde untuk musuh biasa dan 20 ronde untuk *final boss*. *Battle* akan selesai jika *player* atau musuh kalah, atau ronde telah



Gambar 4. salah satu *interface* sistem *battle game Brightsouls*

melebihi ronde maksimum, dimana untuk kasus tersebut, *battle* akan dinyatakan sebagai *draw* (seri).

Karakter yang dimainkan dan musuh memiliki 3 atribut yang mempengaruhi hasil *battle*, yaitu HP, STR, dan DEF. HP menyatakan atribut *Health points* dimana jika satu pihak memiliki HP 0 atau negatif maka dinyatakan kalah. Sedangkan STR menyatakan atribut *Strength* yang menentukan seberapa besar *damage* yang bisa diberi dan DEF menyatakan atribut *Defense* yang menentukan seberapa besar *damage* yang diterima bisa dikurangi. *Game* tidak menunjukkan nilai STR dan DEF yang dimiliki oleh musuh apapun kepada *player*.

Setiap ronde akan ada 4 aksi musuh yang ditampilkan ke layar dan *player* akan memasukkan 4 aksi untuk melawan set aksi pihak musuh. 2 aksi musuh ditutup sehingga *player* tidak mengetahuinya. Aksi yang bisa digunakan dinyatakan dengan A, B, dan F yang menyatakan *Attack*, *Block*, dan *Flank*. *Rule* yang digunakan untuk membandingkan aksi mana yang bisa mengalahkan aksi lain mirip dengan permainan gunting batu kertas, yaitu A mengalahkan F, F mengalahkan B, dan B menahan A. Hanya A dan F yang bisa memberi *damage* pada lawan.

Kemudian, satu hal yang menjadi fokus makalah ini adalah *rule* yang mengatur bagaimana *damage* dikalkulasi. Disini akan didefinisikan 2 *rule* yang berbeda yang sama-sama mengatur bagaimana hubungan atribut STR dan DEF dengan *damage* dimana pada bagian selanjutnya akan dilakukan perbandingan. Untuk memudahkan penyebutan *rule*, kita akan memberi nama pada kedua *rule* tersebut, yakni Nebby dan Joko. Seterusnya, sistem *battle* yang memakai *rule* Nebby, misalnya, akan disebut sebagai sistem Nebby.

A. Rule Nebby

Pada sistem Nebby, ada 2 jenis *damage* yang bisa

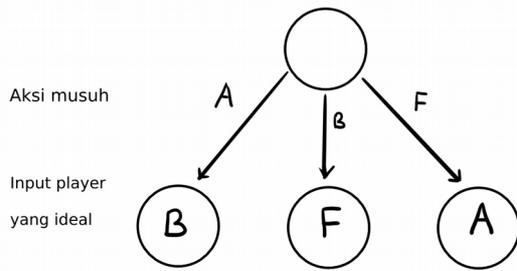
dihasilkan, yaitu *damage* murni dan *damage* yang dikurangi. *damage* murni dihasilkan jika aksi A melawan F atau aksi F melawan B sedangkan *damage* yang dikurangi dihasilkan jika aksi A melawan aksi A atau aksi F melawan aksi F. *damage* murni bernilai sama dengan nilai STR pihak yang memberikan *damage*, sedangkan *damage* yang dikurangi bernilai sama dengan nilai STR pihak yang memberi *damage* dikurangi nilai DEF yang menerima *damage* dengan nilai minimal *damage* adalah 0. *Rule* Nebby memberi tambahan aturan yaitu jika aksi A melawan aksi A atau aksi F melawan aksi F, yang berhasil memberi *damage* adalah pihak yang memiliki STR lebih tinggi. Jika nilai STR kedua pihak sama, maka tidak ada *damage* yang diterima oleh kedua pihak.

B. Rule Joko

Rule Joko adalah sama dengan *rule* Nebby namun memiliki satu perbedaan, yakni jika aksi A melawan aksi A atau aksi F melawan aksi F, maka kedua pihak akan tetap memberi *damage* ke lawan dengan jenis *damage* yang dikurangi tanpa memperhitungkan pihak mana yang memiliki STR yang lebih tinggi.

V. ANALISIS SISTEM BATTLE MENGGUNAKAN POHON KEPUTUSAN

Implementasi pohon keputusan pada analisis ini adalah untuk menunjukkan alur berpikir *player* saat memainkan sistem *battle game Brightsouls*. Gambar 4. menunjukkan pohon keputusan alur berpikir *player* saat menentukan aksi untuk melawan satu aksi musuh yang tidak ditutup dimana *player* memiliki 100% kemungkinan untuk menang.



Gambar 5. Alur berpikir *player* untuk menentukan aksi yang melawan aksi musuh yang diketahui

Untuk selanjutnya, pohon keputusan akan menunjukkan alur berpikir *player* saat menentukan aksi untuk melawan aksi musuh yang ditutup dimana pada analisis ini, beberapa asumsi akan dibuat, yakni:

- *Player* tidak mengetahui nilai STR dan DEF musuh.
- Nilai DEF musuh lebih kecil dari nilai STR karakter.
- Nilai HP musuh jauh lebih besar dari nilai STR *player*, yang berarti karakter tidak bisa langsung *one-hit KO* musuh.
- *Player* mengikuti strategi sebagai berikut: memprioritaskan aksi yang membuat *damage* yang diterima oleh karakter paling sedikit, dan

kemudian memprioritaskan aksi yang membuat *damage* yang diberi kepada musuh paling besar.

Dari pohon keputusan yang dihasilkan, kita akan melihat apakah suatu *rule* tertentu

- memiliki strategi dominan, dimana satu aksi menjadi solusi untuk banyak skenario [3]
- variasi dampak untuk aksi yang sama

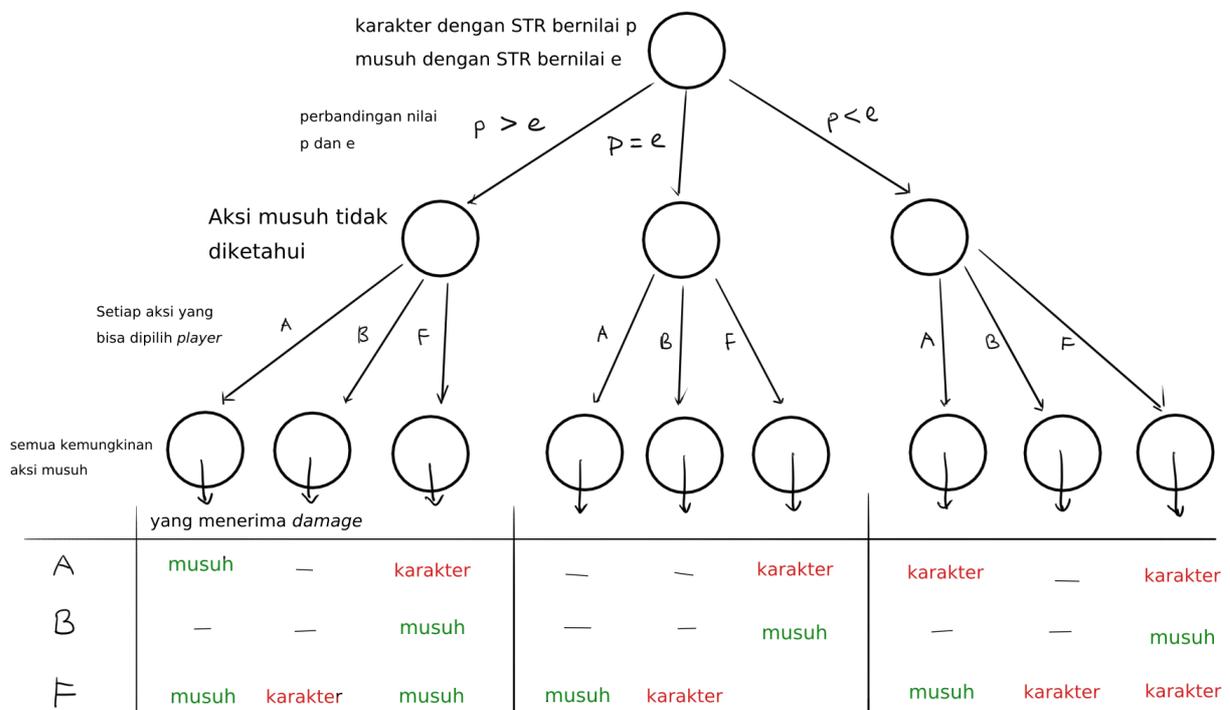
Dengan itu, kita bisa membandingkan kedua *rule* yang akan dianalisa dan menentukan *rule* mana yang menciptakan pengalaman bermain yang lebih baik.

A. Analisis Sistem Nebby

Pada *rule* Nebby, ada suatu skenario dimana atribut STR karakter dan musuh diperhitungkan untuk menentukan siapa yang berhasil memberi *damage*, dan karena *player* tidak mengetahui nilai STR musuh, maka pada pembuatan pohon keputusan, *player* akan membagi skenario menjadi 3 bagian, yaitu skenario dimana STR karakter lebih tinggi dari musuh, skenario dimana STR karakter lebih rendah dari musuh, dan skenario dimana nilai STR kedua pihak sama.

Pohon keputusan yang terbentuk ada pada gambar 6.

Dari pohon tersebut bisa kita lihat bahwa aksi A memiliki kemungkinan lebih kecil untuk karakter



Gambar 6. Pohon keputusan yang menunjukkan alur berpikir *player* pada sistem Nebby

menerima *damage* di setiap skenario. Karena itu, *rule* Nebby dikatakan memiliki strategi dominan dimana *player* bisa akan selalu memilih aksi A untuk melawan aksi musuh yang ditutup karena untuk musuh dengan nilai STR apapun, aksi A akan mengeluarkan hasil yang memiliki kemungkinan yang selalu menguntungkan bagi *player* dibanding aksi lainnya. Dengan ini, kita mengatakan bahwa *rule* Nebby bukanlah *rule* yang baik karena *game* dengan strategi dominan akan membuat permainan bisa dimenangkan oleh hanya satu strategi [4].

Hal lain yang bisa diperhatikan dari pohon tersebut yaitu adanya variasi hasil yang dikeluarkan untuk aksi yang sama. Dengan hal ini kita bisa mengatakan bahwa *rule* Nebby memiliki variasi dampak untuk aksi yang sama.

A. Analisis Sistem Joko

Untuk menganalisa *rule* Joko, pertama kita akan menggunakan skenario yang digunakan untuk menganalisa *rule* Nebby yaitu dengan memperhitungkan perbandingan nilai STR karakter dan musuh.

Pohon keputusan yang terbentuk ada pada gambar 7.

Dari pohon tersebut bisa kita lihat perbedaannya dengan sistem Nebby dimana untuk sistem Joko, aksi A memiliki lebih tinggi resiko walaupun aksi A masih lebih menguntungkan dibanding dengan lainnya. Hal ini

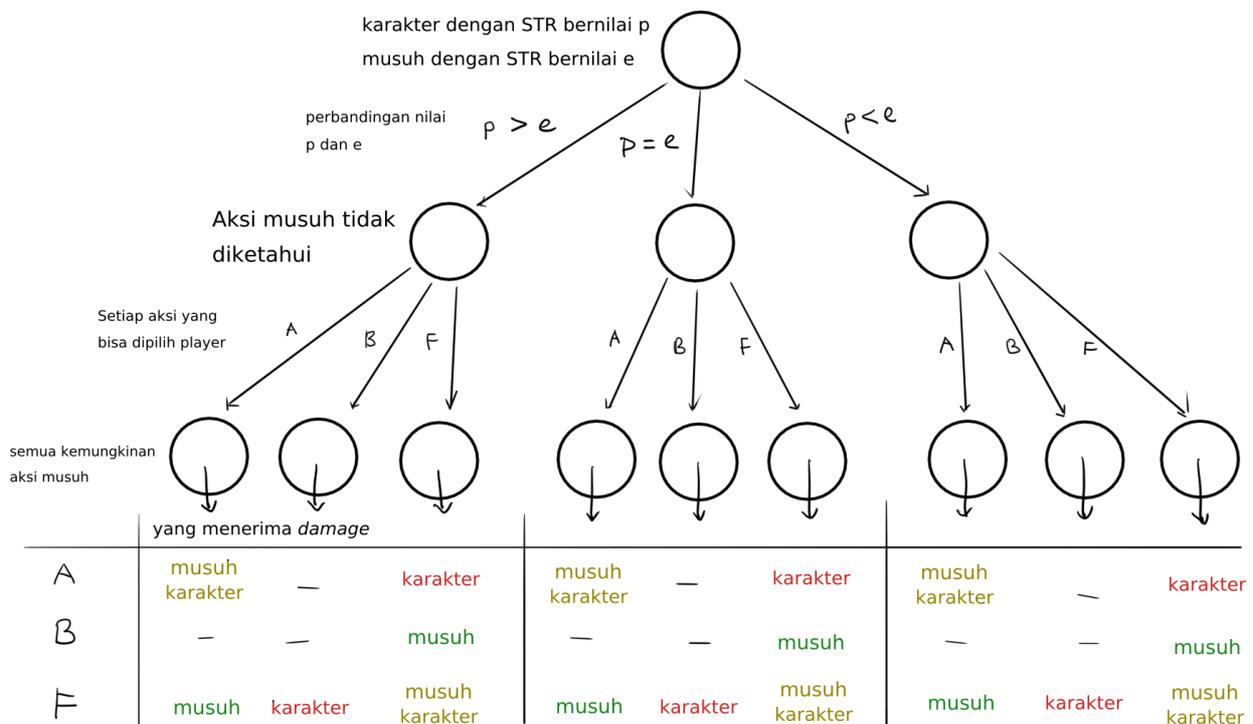
membuat aksi A di sistem Joko tidak sedominan aksi A di sistem Nebby. Namun ada hal lain yang perlu diperhatikan, yaitu ketiga aksi akan mengeluarkan hasil yang sama sehingga bisa dikatakan perbandingan nilai STR tidak menyebabkan variasi hasil untuk aksi yang sama di sistem Joko.

Hal yang berbeda bisa kita lihat jika kita mengubah skenario dimana *player* memperhitungkan perbandingan nilai DEF karakter dengan nilai STR musuh. Sistem Joko memungkinkan karakter menerima *damage* 0 untuk situasi aksi A melawan aksi A atau aksi F melawan aksi F sehingga pohon keputusan yang berbeda bisa terbentuk.

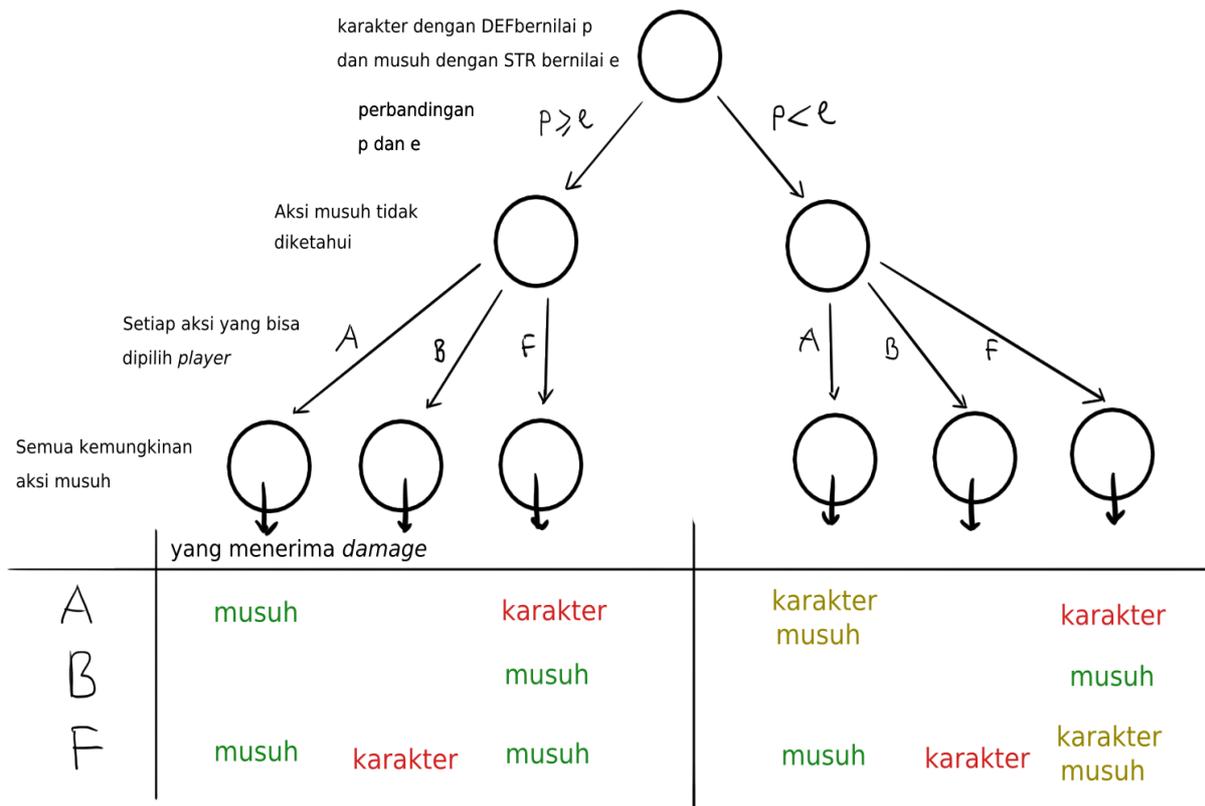
Pohon keputusan yang terbentuk untuk skenario yang memperhitungkan perbandingan DEF karakter dengan STR musuh ada pada gambar 8. dari situ terlihat bahwa ada variasi yang terjadi jika skenarionya diubah seperti yang disebutkan.

V. KESIMPULAN

Dengan makalah ini, kita telah mendemonstrasikan penggunaan pohon keputusan untuk membandingkan aspek yang membentuk suatu *game* dimana untuk makalah ini aspek tersebut adalah sebuah *rule* untuk *game* *Brightsouls*. Dengan menggunakan pohon keputusan, perbandingan kedua *rule* menjadi lebih



Gambar 7. Pohon keputusan yang menunjukkan alur berpikir *player* pada sistem Joko



Gambar 8. Pohon keputusan yang menunjukkan alur berpikir *player* pada sistem Joko dengan skenario yang berbeda

kuantitatif dan terlihat jelas setiap dampaknya pada pemain dan permainan secara umum.

Dari hasil yang kita dapat, kita bisa mengatakan bahwa *rule* Nebby merupakan *rule* yang membuat permainan memiliki strategi dominan namun memiliki variasi dampak untuk setiap aksi yang dilakukan sedangkan *rule* Joko adalah *rule* yang tidak terlalu banyak memiliki variasi dampak untuk satu aksi namun *rule* tersebut tidak membuat permainan secara keseluruhan memiliki strategi dominan. Dari cara kita membandingkan kedua *rule*, bisa juga disimpulkan bahwa masih banyak aspek lain yang bisa menyeimbangi aspek-aspek ini karena perbandingan yang dilakukan hanya memperhitungkan sedikit faktor dari banyak faktor yang bisa mempengaruhi permainan di *game Brightsouls* ini. Seperti faktor atribut HP, dimana besarnya nilai atribut tersebut bisa mempengaruhi cara bermain seorang *player*.

Untuk kedepannya, diharapkan kepada para desainer *game* yang membaca makalah ini untuk dapat mengerti konsep dasar penggunaan pohon keputusan untuk melakukan perbandingan aspek-aspek yang dimiliki suatu *game* seperti mekanik atau *rule*, serta diharapkan juga agar dapat mengimplementasikannya ke proses

desain *game* agar karya yang dihasilkan menjadi lebih baik.

DAFTAR PUSTAKA

- [1] <https://www.makeschool.com/gamernews/298/5-basic-elements-of-game-design> diakses tanggal 9 Desember 2016 pk. 00.55 WIB
- [2] Munir, Rinaldi. 2003. *Matematika Diskrit*. Bandung : Penerbit Informatika.
- [3] Schell, Jesse. 2008. *The Art of Game Design: A Book of Lenses*. Florida : CRC Press.
- [4] <http://blog.wolfire.com/2009/01/game-theory-applied-to-game-design/> diakses tanggal 9 Desember 2016 pk. 10.03 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, 9 Desember 2016
ttd



Kevin Erdiza Yogatama,13515016