

Perancangan *Decision Tree* untuk *Matchmaking* Permainan Daring *Battle Royale* Fortnite

Matthew Mahendra-13521007¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

¹13521007@std.stei.itb.ac.id

Abstrak—*Battle royale* adalah bentuk pertandingan yang mengumpulkan sejumlah pemain untuk saling mengeliminasi sesama pemain hingga tersisa satu pemain di akhir pertandingan. Salah satu contoh permainan ini adalah Fortnite. Salah satu keluhan dari pertandingan dengan bentuk ini adalah lawan yang tidak seimbang, dalam artian memiliki tingkat kemampuan yang jauh berbeda. Untuk itu, penulis mencoba untuk merancang *decision tree* untuk menyelesaikan persoalan ini. Hasilnya adalah, dapat dibentuk *decision tree* untuk menyelesaikan permasalahan ini dengan mempertimbangkan faktor-faktor dari setiap pemain agar menghasilkan pertandingan yang adil.

Kata Kunci—*battle royale*, *decision tree*, *tree*, *matchmaking*, permainan daring, Fortnite

I. PENDAHULUAN

Permainan daring dengan bentuk *battle royale* sudah menjadi hal yang umum pada dekade ini. Dapat diberikan beberapa contoh permainan daring ini seperti Fortnite, Player's Unknown Battleground (PUBG), Free Fire, dll. Permainan-permainan yang telah disebutkan menggunakan bentuk *battle royale* dengan setiap pertandingan memiliki ± 100 (seratus) pemain yang akan saling mengeliminasi satu sama lain hingga tersisa satu pemain saja. Pada makalah ini, akan digunakan permainan Fortnite.

Proses pembentukan pertandingan dengan 100 pemain ini disebut sebagai *matchmaking*. Proses *matchmaking* dalam *battle royale* masih memiliki beberapa kekurangan. Misalnya ada kemungkinan pemain dengan tingkat kemampuan 1 dimasukkan ke pertandingan dengan mayoritas tingkat kemampuan 10 ataupun sebaliknya. Hal ini dapat membuat sebuah pertandingan menjadi tidak adil dan jika tidak ditangani, merusak citra dari permainan daring.

Untuk itu, penulis menulis makalah ini untuk mencoba membuat rancangan *decision tree* *matchmaking* permainan daring *battle royale*. Dalam proses perancangan ini, perlu diperhatikan beberapa hal yaitu faktor-faktor *matchmaking* serta kompleksitas *decision tree*. Faktor-faktor yang dipertimbangkan akan memengaruhi luaran dari *decision tree* dan kompleksitas dari *tree* yang dibuat akan memengaruhi lama waktu dari *matchmaking*.

II. DASAR TEORI

A. *Tree*

Tree atau pohon adalah suatu modifikasi dari *graph*. Jika pada *graph* diperbolehkan *vertex*-nya saling terhubung mem-

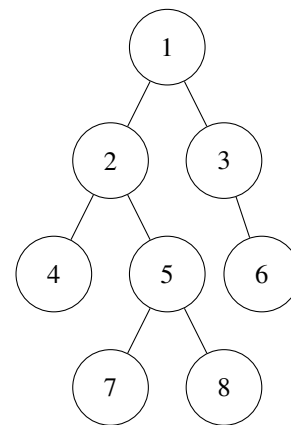
buat sirkuit, maka pada *tree* tidak diperbolehkan untuk memiliki sirkuit. Oleh sebab itu, sebuah *tree* tidak akan memiliki lintasan tertutup. Setiap *tree* memiliki *vertex* V dan *edge* E [1]. *Tree* dibentuk secara hirarki, dan setiap V dapat terhubung ke V lainnya. Jumlah V yang terhubung pada sebuah *tree* bisa jadi 0 atau disebut sebagai *tree* yang kosong. Untuk setiap V dapat dibentuk suatu sub-*tree* [2].

Suatu *tree* akan memiliki struktur sebagai berikut [3],

- 1) *Root* atau 'akar' adalah *vertex* asal dari sebuah *tree*
- 2) *Child* adalah *vertex* 'anak' dari suatu *vertex*. Untuk setiap *child* yang tidak memiliki *vertex* *child* lagi, maka *child* tersebut dapat dikatakan sebagai sebuah *leaf* atau daun
- 3) *Parent* merupakan asal *vertex* dari *child*

Sebuah *child* dapat memiliki 'saudara' atau *sibling* jika pada tingkatan yang sama memiliki *vertex* yang berasal dari suatu *vertex* yang sama. Tingkatan pada *tree* merupakan kedalaman setiap *vertex*.

Untuk lebih jelasnya, perhatikan gambar 1. *Vertex* 1 meru-



Gambar 1. Contoh *Tree*

pakan *root*. *Vertex* 2 dan 3 merupakan *child* dari 1 dengan 2 dan 3 adalah *sibling*. Lalu 4 dan 5 adalah *child* dari 2 dan keduanya merupakan *sibling*. Kemudian 6 adalah *child* dari 3. Terakhir, 7 dan 8 adalah *child* dari 5 dan keduanya merupakan *sibling*.

Tingkatan pada gambar 1 dimulai dari *root*, dengan *root* adalah tingkatan pertama. Tingkatan kedua adalah pada 2 dan

3. Tingkatan ketiga berada pada 4,5, dan 6. Tingkatan keempat berada pada 7 dan 8. Oleh karena itu *tree* ini memiliki empat tingkatan. Dapat diperhatikan bahwa terdapat *vertex* yang tidak memiliki *child*. Hal ini diperbolehkan pada *tree* karena tidak melanggar syarat sebuah *tree*.

Suatu *tree* dapat disusun secara rekursi untuk membantu menyelesaikan persoalan rekursi [4]. Penyelesaian rekursi menggunakan metode *divide and conquer*.

B. Aritas Tree

Aritas dari suatu *tree* ditentukan dari jumlah *child* terbanyak dari suatu *vertex* atau juga dari derajat tertinggi dari semua *vertex*. Akibatnya, sebuah *n*-ary tidak akan memiliki *child* yang lebih dari dua. Gambar 1 merupakan *tree* dengan aritas biner atau dapat dikatakan *binary tree* karena *child* maksimal dari setiap *vertex*-nya adalah dua.

Aritas secara tidak langsung memengaruhi kompleksitas dari algoritma yang menggunakan struktur data *tree*. Pada algoritma insersi dan delesi pada heap (sebuah struktur data berbasis *tree*, kompleksitas waktu bergantung pada jumlah *vertex tree* [3]. Oleh karena itu, semakin banyak jumlah *vertex* dan *edge*, maka kompleksitas waktu sebuah algoritma dapat bertambah.

C. Decision Tree

Sebuah *tree* dapat digunakan untuk menentukan sebuah keputusan. *Tree* ini disebut sebagai *decision tree*. *Decision tree* dapat disusun secara *n*-ary Struktur dari sebuah *decision tree* adalah sebagai berikut [5],

- 1) Diberikan *root* yang menjadi suatu pertanyaan. Jawaban pertanyaan diarahkan ke beberapa *child* untuk beberapa macam kemungkinan jawaban
- 2) *Child* yang sudah ditentukan kemudian dapat menjadi pertanyaan tambahan. Proses dapat diulang sesuai kebutuhan hingga berujung pada *leaf*
- 3) Sebuah *leaf* merupakan keputusan

D. Contoh Kasus Decision Tree Indeks Prestasi

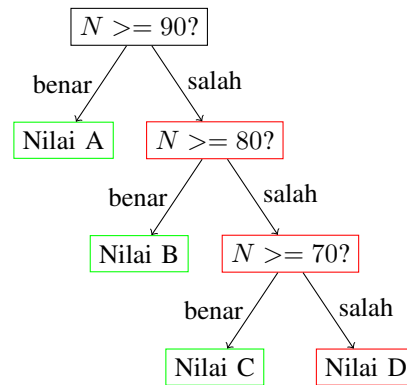
Misalkan dibuat sebuah *decision tree* untuk menentukan indeks dari mahasiswa. Maka rancangan dari *tree* tersebut adalah,

- 1) Menentukan total kemungkinan indeks. Misalnya ada 4 indeks yaitu A, B, C, dan D
- 2) Menentukan batasan dari setiap indeks. Misalkan, batas bawah dari nilai A adalah 90, B adalah 80, C adalah 70, dan dibawah 70 adalah D

Dari rancangan tersebut, salah satu macam pohon keputusan yang dapat dibuat adalah sebagai berikut,

Tree pada gambar 2 akan menerima *N* yaitu nilai mahasiswa. Diperiksa apakah $N \geq 90$. Jika iya, maka akan diberikan nilai A, jika tidak akan mengevaluasi $N \geq 80$. Hal ini dilakukan terus hingga berada pada *leaf* 'Nilai D'. *Tree* pada gambar 2 memiliki aritas biner (*binary*).

Bentuk *decision tree* seperti pada gambar 2 sering digunakan dalam bahasa pemrograman dalam pembangunan kasus *conditional*. Gambar 2 sendiri dapat dinyatakan menggunakan bahasa pemrograman.



Gambar 2. *Decision tree* sederhana untuk kasus indeks mahasiswa

E. Permainan Daring Fortnite

Fortnite adalah permainan daring yang dikembangkan oleh Epic Games pada tahun 2017 [6]. Bentuk permainan ini adalah *battle royale* dengan jumlah setiap pertandingan 100 pemain. Permainan dimenangkan jika hanya tersisa 1 pemain. Fortnite sendiri memiliki alur ceritanya sendiri yang dibagi menjadi musim-musim. Dalam permainan Fortnite, untuk mempersingkat waktu permainan, diadakan badai yang dapat mengeliminasi pemain jika terlalu lama berada dalam badai tersebut. Badai ini akan menutupi seluruh peta permainan pada akhirnya. Contoh badai dapat dilihat pada gambar 3



Gambar 3. Badai yang akan mengeliminasi pemain. Sumber: fortnite-archive fandom

Untuk melakukan serangan, sama seperti permainan *battle royale* umumnya, penyerangan dilakukan menggunakan senjata untuk menyerang musuh hingga nyawa dari musuh tersebut menjadi habis atau dengan kata lain, tereliminasi. Pada Fortnite dapat dikategorikan beberapa macam senjata yaitu *explosive, melee, assault, SMG, shotgun, sniper, pistol*.

Berbeda dari permainan *battle royale* lainnya, pemain Fortnite dapat membangun bangunan kecil seperti tembok, lantai, tangga, dan atap untuk membantu permainan, seperti menghalangi serangan musuh atau mengakses daerah yang lebih tinggi seperti atap rumah, bangunan, maupun bukit.

Untuk setiap permainan yang dilalui, maka setiap pemain akan diberikan *experience points* (XP) yang akan digunakan untuk menaikkan tingkat pemain. Setiap musim dalam permainan ini, tingkatan akan kembali di-*reset* menjadi 0, akan



Gambar 4. Proses eliminasi pemain dengan menggunakan senjata. Terlihat berapa nyawa yang dihilangkan. Sumber: Kanal Siaran Youtube SENSEI

tetapi akumulasi XP tetap disimpan. Begitu pula jumlah permainan yang dimenangkan, kalah, dan totalnya.

Alur bermain pada permainan Fortnite adalah sebagai berikut,

- 1) Pemain memasuki lobi permainan. Lalu peladen permainan akan melakukan *matchmaking*
- 2) Selesai *matchmaking* pemain akan dimasukkan ke pertandingan bersama ± 100 pemain lainnya (lebih kurang biasanya dikarenakan ada pemain yang keluar sehingga biasanya berada pada kisaran 96 – 100 pemain)
- 3) Permainan dimulai. Pemain saling mengeliminasi dan dimenangkan oleh pemain terakhir yang menang
- 4) Semua pemain, setelah menyelesaikan permainannya akan diberikan XP untuk setiap eliminasi, lama pemain dalam permainan, dll.



Gambar 5. Proses pembangunan tangga pada permainan. Sumber: Kanal Siaran Youtube MONGRAAL

III. HASIL PENELITIAN

A. Data yang Digunakan

Untuk melakukan perancangan *decision tree*, maka diperlukan data-data pertimbangan untuk *matchmaking*. Dari alur permainan dan juga dari struktur permainan, akan diambil beberapa data yang akan digunakan sebagai pertimbangan. Berikut adalah data dan alasan penggunaannya dalam *decision tree* ini,

- Tingkatan pemain. Tingkatan pemain dapat membedakan antara pemain baru dan pemain lama. Misalnya diambil batasan pemain baru adalah mereka yang berada pada

tingkatan akumulasi atau tingkatan setiap musimnya berada di bawah 50. Untuk pemain yang baru, maka pemain akan dikelompokkan dengan pemain yang baru dan juga untuk yang sudah lama dengan pemain yang lama

- Keahlian pemain. Keahlian pemain dapat dikategorikan sebagai pemain yang lebih sering membangun, pemain yang lebih sering menggunakan senjata, pemain yang lebih sering bertahan hidup, atau macam gabungan dari beberapa kemungkinan tersebut
- *Win-Loss rate pemain*. Merupakan persentase seringnya pemain menang atau kalah dalam setiap pertandingan. Pemain yang sering menang akan dikelompokkan dengan pemain yang sering menang dan untuk pemain yang sering kalah akan dikelompokkan dengan pemain yang sering kalah

Untuk kemungkinan keahlian pemain, akan dipecah kembali menjadi beberapa pertimbangan yaitu sebagai berikut,

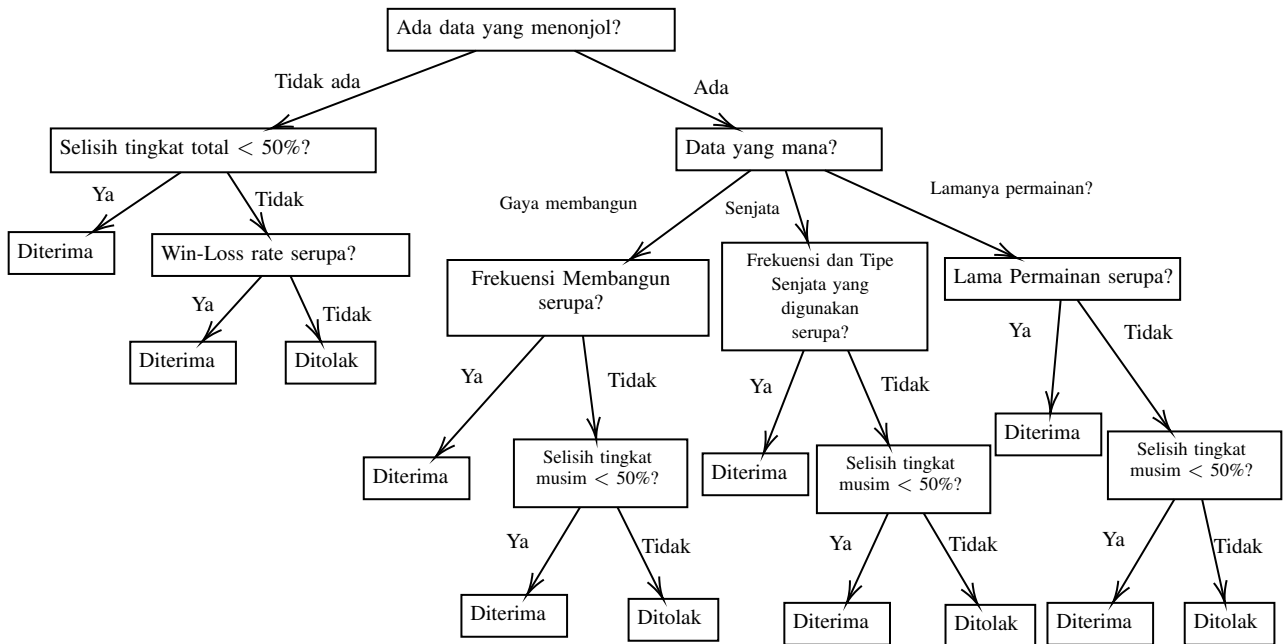
- Gaya membangun. Proses membangun sebuah struktur pada Fortnite sangat dipengaruhi oleh keahlian pemain. Ada pemain yang dapat membangun dengan sangat cepat dan juga ada yang lama. Frekuensi tipe bangunan yang digunakan juga akan menjadi pertimbangan dalam *decision tree*
- Senjata yang digunakan. Akan dihitung frekuensi penggunaan tipe senjata apa. Tipe senjata berdasarkan sub-bab II-E
- Lama pemain dalam pertandingan. Dilihat rata-rata dari pemain dalam pertandingan. Lama dalam pertandingan menandakan keahlian bertahan hidup, entah karena dieeliminasi oleh pemain atau tereliminasi oleh elemen lainnya misalnya badai atau karena lingkungan, misalnya terjatuh dari tempat yang tinggi.

Proses pengumpulan data dilakukan oleh peladen (*server*) permainan ini berdasarkan masukan dari peserta dan juga statistik yang disimpan. Teknik pengumpulan data ini tidak dibahas dalam tulisan ini karena tidak termasuk dalam lingkup pembahasan tulisan.

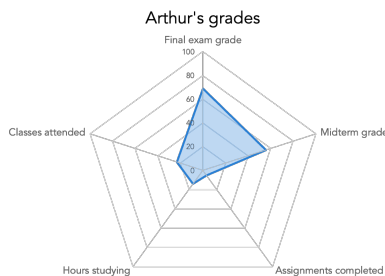
B. Pembuatan Decision Tree

Decision tree yang akan dibuat bertujuan untuk mengetahui apabila sembarang pemain *A* dapat disatukan dalam satu pertandingan bersama dengan pemain *B, C, ..., 100* pemain. Berdasarkan data-data pada sub-bab III-A, maka alur dari *decision tree* yang akan dibuat adalah sebagai berikut,

- 1) *Tree* akan menerima data pemain *A* dan *B*. Akan diuji apakah *A* dan *B* setara atau tidak
- 2) Pertama dilihat apakah dari data-data keahlian apakah ada yang menonjol atau tidak. Bagian yang dianggap menonjol adalah pada bagian keahlian pemain. Menonjol pada hal ini dapat direpresentasikan sebagai data yang ditunjuk pada sebuah *radar chart* (contoh pada gambar 7). Sebagai catatan, *radar chart* merupakan sebuah representasi data untuk menunjukkan data dengan beberapa dimensi yang unik, misalnya dalam hal ini adalah keahlian dari pemain [7]. Jika ada yang menonjol, apakah menonjol pada data yang sama atau tidak.



Gambar 6. Hasil *Decision Tree* untuk *Matchmaking*



Gambar 7. Contoh Radar Chart. Sumber: Disadur dari [7]

C. Hasil *Decision Tree*

Dari III-B, maka dibuat *decision tree* pada gambar 6. Dibuat *tree* yang memiliki spesifikasi sebagai berikut,

- 1) memiliki empat tingkatan
- 2) merupakan *tree* ternary (derajat tiga)
- 3) memiliki total sembilan *vertex* dan dua belas *leaf* yang menjadi sembilan kondisi dan dua belas keputusan yang dapat diambil. Keputusan hanya terdiri dari diterima atau ditolak seorang pemain untuk bermain bersama dengan pemain yang dibandingkan
- 4) untuk *child* kanan dari *root* jika ada beberapa data yang sama-sama saling menonjol, maka *vertex* yang lebih kiri memiliki *precedence* yang lebih utama. Dimungkinkan untuk melakukan rekursi apabila data yang sama. Jika semakin banyak keputusan diterima, maka bisa semakin besar kemungkinannya untuk dipasangkan dengan pemain yang menjadi tes. Proses rekursi dimulai dari pertanyaan "Data yang mana?"

Penggunaan tingkat musim dan tingkat total dibedakan posisi tempatnya pada *decision tree*. Tingkat total digunakan ketika tidak ada data yang menonjol karena bisa jadi data belum lengkap karena melalui pergantian musim sehingga datanya menjadi kurang relevan dan salah satu pemain belum pernah melakukan pertandingan. Tingkat musim digunakan ketika ada data yang menonjol karena digunakan untuk memastikan apakah pemain ini adalah sudah pernah bermain pada musim yang sedang berjalan atau belum. Menjadi pertimbangan bahwa untuk musim tertentu dapat terjadi perubahan mekanisme permainan sehingga pemain perlu waktu untuk menyesuaikan gaya bermainnya.

- a) Jika pada data yang sama, maka akan dilakukan perbandingan berikutnya
 - b) Jika tidak sama, maka pemain *B* akan dilihat dari tingkatannya. Jika tingkatannya pada suatu musim *X* memiliki selisih yang terlalu jauh dengan pemain *A*, maka akan diperiksa terlebih dahulu *win-loss rate*-nya
- 3) Kedua, dari data keahlian yang menonjol, dicoba untuk mencari kesamaan. Jika ada data yang cenderung sama, maka pemain *B* diterima untuk bertanding dengan pemain *A*. Akan tetapi jika tidak ada, maka akan dilihat berdasarkan tingkatan totalnya dan dihitung seperti kasus sebelumnya

Prioritas pada *decision tree* ini adalah keahlian pemain, lalu tingkatannya baik dilihat dari musim maupun secara keseluruhan.

IV. PEMBAHASAN

Dari *tree* yang dibuat pada gambar 6, dilakukan uji coba terhadap beberapa kasus untuk menghitung dan juga menjelaskan cara kerja *tree* ini.

A. Data Uji Coba

Diberikan empat pemain yaitu pemain *A*, pemain *B*, pemain *C*, dan pemain *D*. Diuji coba apakah untuk pemain *B*, *C*, dan *D* dapat bertanding dengan pemain *A* pada suatu *matchmaking*. Data disajikan pada tabel I, tabel II, tabel III, dan tabel IV.

Nilai dari data, khususnya yang pada bagian yang termasuk dalam keahlian dari peserta dibuat dalam bentuk persentase frekuensi keahlian tersebut. Hal ini dibuat agar dapat membuat suatu data yang menonjol diantara data-data yang digunakan.

TABEL I
DATA PEMAIN A

Data Pemain	Nilai Data
Win Rate	30%
Frekuensi dan Tipe Penggunaan Senjata	Shotgun (70%), SMG(30%)
Frekuensi Membangun	40%
Lama Permainan	15 menit
Tingkatan Musim	Tingkat 70
Tingkatan Total	Tingkat 120

TABEL II
DATA PEMAIN B

Data Pemain	Nilai Data
Win Rate	0%
Frekuensi dan Tipe Penggunaan Senjata	Pistol (100%)
Frekuensi Membangun	5%
Lama Permainan	10 menit
Tingkatan Musim	Tingkat 5
Tingkatan Total	Tingkat 5

TABEL III
DATA PEMAIN C

Data Pemain	Nilai Data
Win Rate	35%
Frekuensi dan Tipe Penggunaan Senjata	Pistol (45%), Shotgun (55%)
Frekuensi Membangun	45%
Tingkatan Musim	Tingkat 75
Tingkatan Total	Tingkat 900

TABEL IV
DATA PEMAIN D

Data Pemain	Nilai Data
Win Rate	0%
Frekuensi dan Tipe Penggunaan Senjata	Semua (0%)
Frekuensi Membangun	0%
Tingkatan Musim	Tingkat 0
Tingkatan Total	Tingkat 0

B. Proses Uji Coba

1) *Membandingkan Pemain A dengan Pemain B*: Akan dilakukan pengujian apakah pemain *A* dapat disatukan dengan pemain *B* dalam satu pertandingan yang sama.

Pertama, data yang menonjol pada pemain *A* adalah data Frekuensi dan Tipe Penggunaan Senjata, yaitu pada tipe *shotgun* dengan frekuensi 30%. Pada pemain *B*, data yang menonjol adalah pada data Frekuensi dan Tipe Penggunaan Senjata, namun dengan tipe senjata yang berbeda yaitu pada *pistol* dengan frekuensi 100%.

Menggunakan gambar 6, penelusuran memasuki sub-*tree* kanan karena ada data yang menonjol dan juga sama yaitu pada data penggunaan senjata. Selisih dari kedua data ini adalah 30% (di bawah 50% yang digunakan sebagai ambang batas), sehingga salah satu syarat terpenuhi. Akan tetapi dikarenakan dari tipe senjata yang sering digunakan tidak sama yaitu antara *shotgun* dengan *pistol*, maka tidak lulu pengujian untuk kasus benar.

Karena itu, selanjutnya ditelusuri dari sub-*tree* kanan untuk diperiksa selisih tingkat musimnya. Tingkat musim dari pemain *A* adalah 70 sedangkan untuk pemain *B* adalah 5. Karena selisihnya melampaui 50%, maka pada *decision tree*, pemain *B* ditolak untuk bermain dengan pemain *A*.

2) *Membandingkan Pemain A dengan Pemain C*: Akan dilakukan pengujian apakah pemain *A* dapat bermain dengan pemain *C* dalam satu pertandingan yang sama.

Pertama, data yang menonjol pada pemain *A* tetap sama seperti pada pembahasan subbab sebelumnya. Untuk pemain *C*, data yang menonjol adalah data frekuensi dan tipe penggunaan senjata, yaitu pada pistol dengan frekuensi 45% dan pada membangun dengan frekuensi 45%. Melalui *decision tree*, karena ada data yang menonjol, maka ditelusuri sub-*tree* kanan.

Karena ada dua data yang menonjol, maka sesuai aturan *decision tree* yang dibuat, dikunjungi terlebih dahulu sub-*tree* yang paling kiri yaitu pada frekuensi membangun. Frekuensi membangun dari *A* dan *C* berada pada 45% dan 40% yang berarti keduanya serupa. Oleh sebab itu diterima.

Karena *decision tree* yang dibuat memungkinkan untuk terjadinya rekursi atau pengulangan jika ada lebih dari satu data yang menonjol, maka untuk pengujian kedua, diuji terhadap data senjatanya.

Data yang menonjol pada senjata memiliki selisih 35% namun tipe yang berbeda, sehingga tidak diterima. Akan tetapi, pada data yang lain, sudah diterima. Oleh karena itu, pemain ini akan tetap diterima untuk bermain dengan pemain *A*.

3) *Membandingkan Pemain A dengan Pemain D*: Data pemain *D* tidak ada yang menonjol dan semua masih 0. Diasumsikan pemain *D* adalah pemain baru. Oleh sebab itu, tidak ada data yang menonjol dari pemain *D* sehingga *decision tree* akan menggunakan sub-*tree* kiri.

Dari selisih tingkatan total, didapatkan bahwa selisihnya sebesar 120%, oleh sebab itu, selanjutnya diperiksa apakah *win-loss rate*-nya serupa atau tidak. Didapatkan bahwa *win-loss rate*-nya tidak serupa sehingga pemain *D* ditolak.

C. Hasil Akhir

Dari pengujian, didapatkan bahwa pemain yang dapat bertanding dengan pemain A dengan adil hanyalah pemain C , sehingga pemain B dan D harus melalui *matchmaking* dengan pemain yang lain.

Didapatkan bahwa proses penelusuran terbanyak adalah sebanyak tiga kali, sesuai dengan kedalaman dikurangi satu dari *decision tree* yang dibuat yaitu tiga.

V. SIMPULAN DAN SARAN

Dari hasil *decision tree* yang telah dibuat diperoleh simpulan sebagai berikut,

- 1) proses *matchmaking* dapat menggunakan *decision tree* untuk memetakan pemain *battle royale* berdasarkan kemampuannya
- 2) pembuatan *decision tree* didasari pada kemungkinan-kemungkinan kasus dibuatnya *decision tree* tersebut. Pada permainan ini, didasari pada sistem dari permainan itu sendiri
- 3) kedalaman dari suatu tingkatan dapat menentukan banyaknya pengujian yang harus dilakukan terhadap suatu data. Dalam kompleksitas algoritma, hal ini setara dengan menentukan kasus terburuk dari algoritma yang dibuat

Dari pembuatan *decision tree* ini pula diberikan saran sebagai berikut,

- 1) membuat *decision tree* yang lebih mudah untuk dimodifikasi terhadap perubahan sistem permainan. Misalnya karena adanya mekanisme yang dapat menguntungkan proses permainan misalnya berlari kencang, menggunakan kendaraan untuk melakukan mobilisasi, melompati sebuah tembok, dll.
- 2) untuk membuat algoritma menjadi lebih cepat dalam aplikasinya, dapat disimpan terlebih dahulu data mana yang menonjol (jika ada). Dengan demikian, proses *matchmaking* dapat menjadi lebih cepat karena sudah ditentukan dan dikelompokkan pemain-pemain yang dapat berpotensi untuk dimasukkan dalam suatu pertandingan yang sama yang berakibat proses pemeriksaan juga menjadi lebih cepat

UCAPAN TERIMA KASIH

Puji dan syukur diucapkan kepada Tuhan yang Maha Esa sehingga makalah ini dapat diselesaikan. Tidak lupa kepada Orang tua penulis yang telah mendukung proses pendidikan penulis sehingga makalah ini dapat disusun, kepada Dr. Ir. Rinaldi Munir, M.T. sebagai dosen pengampu IF2120 yang telah menurunkan ilmunya tentang Matematika Diskrit dan kepada seluruh kolega yang telah menyemangati proses pembuatan makalah ini.

REFERENSI

- [1] G. Valiente, *Algorithms on Trees and Graphs*, 2nd ed. Cham: Springer, 2021.
- [2] A. Papoutsaki and W. Devanny, "Lecture 16: Binary tree," <https://cs.pomona.edu/classes/cs062-2018sp/lectures/Lecture16/Lecture16.pdf>, 2018, diakses: 6-12-2022.

- [3] J. Wengrow, *A Common-Sense Guide to Data Structures and Algorithms*, 2nd ed., B. Macdonald, Ed. Raleigh: The Pragmatic Programmers, 2020.
- [4] H. Kumar, "Recursion tree method," <https://www.scaler.com/topics/data-structures/recursion-tree-method/>, 2022, diakses: 6-12-2022.
- [5] R. Munir, *Matematika Diskrit*, 7th ed. Bandung: Penerbit Informatika, 2016.
- [6] Anonymous, "Battle royale," <https://fortnite.fandom.com/wiki/>, diakses: 3-12-2022.
- [7] M. Cisneros, "What is a spider chart?" <https://www.storytellingwithdata.com/blog/2021/8/31/what-is-a-spider-chart>, 2021, diakses: 8-12-2022.

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 2022



Matthew Mahendra
NIM: 13521007