

# Penerapan Kombinatorial dan Pohon Keputusan dalam Menentukan Langkah Terbaik pada Permainan *Sheep Fight*

Ainun Fitryh Vianiryzki - 13517057  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
13517057@std.stei.itb.ac.id

**Abstrak**—Aplikasi permainan dalam *smartphone* semakin beragam, salah satunya adalah Hago. Hago merupakan aplikasi pertemanan yang di dalamnya terdapat banyak *mini-games* yang dapat dimainkan secara *multiplayer*. Pengguna Hago akan dicocokkan secara random untuk kemudian bermain permainan yang dipilih. *Sheep Fight* adalah salah satu permainan yang disediakan Hago. Makalah ini akan membantu pemain *Sheep Fight* agar selalu bisa memenangi pertandingan dengan memprediksi kemungkinan kombinasi domba milik lawan.

**Kata Kunci**—Graf, Pohon, Hago, *Sheep Fight*.

## I. PENDAHULUAN

Perkembangan teknologi yang semakin melaju pesat, membuat setiap orang tidak dapat menghindari kemajuan dari teknologi. Segala aspek kehidupan, baik itu sosial, budaya, pendidikan, kesehatan dan bahkan politik pun tak lepas dari sentuhan teknologi.

Maraknya perkembangan teknologi, berbanding lurus dengan munculnya berbagai macam aplikasi yang bertujuan untuk menunjang, membantu, dan mempermudah pekerjaan manusia.

Aplikasi pada umumnya dikembangkan di dua jenis *device*, yaitu di telefon genggam dan *Personal Computer (PC) / MAC*. Saat ini, banyak *developer* yang memilih untuk mengembangkan aplikasinya di telefon genggam, utamanya pada telefon genggam yang memiliki sistem operasi Android.

Menurut statistik, pada tahun 2017, jumlah aplikasi baru yang dirilis di Google Play berjumlah kurang lebih 2500 aplikasi per-hari (2500 apps/day) atau sekitar 100 aplikasi per-jam (100 apps/hour). *Developer* cenderung memilih Android karena membuat aplikasi di Android dirasa lebih mudah dan mengkomersikan ke Google Play juga jauh lebih sederhana dan tidak sesulit mengkomersikan ke App Store<sup>1</sup>. Alasan lain *developer* memilih Android adalah karena pengguna Android lebih banyak daripada pengguna iPhone. Pada tahun 2016, pengguna Android menyentuh angka 86.2%, sedangkan pengguna iPhone hanya sebanyak 12.9%.

Pada tahun 2018, salah satu tren aplikasi yang disukai oleh

masyarakat Indonesia adalah aplikasi yang memungkinkan penggunaannya untuk bertemu dan berkenalan secara daring untuk kemudian menjalin pertemanan.

Hago adalah salah satu contoh aplikasi yang memasangkan antar penggunaannya secara acak untuk kemudian bermain *mini-game* bersama secara *multiplayer*. Pemasangan pengguna secara acak ini tentunya menarik karena dengan begitu, pengguna Hago dapat memiliki teman baru.

*Mini-game* yang terdapat di Hago sangatlah banyak. Salah satu permainan yang paling digemari oleh pengguna Hago adalah *Sheep Fight*.

*Sheep Fight* adalah permainan yang dimainkan oleh dua pemain. Masing-masing pemain akan diberikan deretan domba-domba yang harus diletakkan di lintasan. Setelah domba diletakkan di lintasan, domba tersebut akan berjalan hingga sampai ke *base* lawan. Untuk melawan, pemain lainnya dapat meletakkan dombanya di lintasan yang sama. Pengurangan nyawa akan terjadi jika pemain berhasil memasukkan domba ke *base* pemain lainnya. Pemain yang memenangkan permainan adalah pemain yang mampu menguras habis nyawa pemain lainnya.

Teori kombinatorial dan pohon merentang dapat diterapkan dalam permainan ini agar pemain bisa selalu memangkan permainan. Kombinatorial digunakan dalam memprediksi domba-domba milik lawan, sedangkan pohon merentang digunakan dalam peletakan domba pada lintasan.

## II. PERMAINAN SHEEP FIGHT

*Sheep Fight* adalah permainan yang terdapat dalam Hago, aplikasi yang saat ini sedang digandrungi oleh anak muda Indonesia. Hago merupakan sebuah aplikasi yang menyediakan fitur *chat* dan *real-time<sup>2</sup> mini-game*.

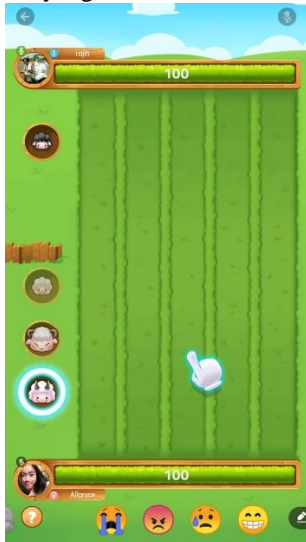
*Sheep Fight* dimainkan oleh dua orang yang dipasangkan secara acak oleh Hago. Jika tidak ingin dipasangkan secara acak, pengguna Hago bisa mengajak pengguna Hago lain berada dalam “daftar teman” untuk bermain.

Permainan *Sheep Fight* dimulai dengan lima lintasan kosong, yang nantinya pemain harus menaruh domba-domba yang

<sup>1</sup> Sebuah platform distribusi digital yang dikembangkan oleh perusahaan Apple Inc., untuk menelusuri dan mengunduh aplikasi telefon genggam yang menggunakan sistem operasi iOS.

<sup>2</sup> Permainan berjalan dalam waktu sebenarnya dan serentak antara semua pihak.

dimiliki pada lintasan yang telah disediakan.



[7] Gambar 2.1

Gambar 2.1 menunjukkan keadaan awal permainan, dimana kelima lintasan masih dalam keadaan kosong. Pada bagian kiri gambar, terdapat antrian tiga domba berwarna putih, dan satu domba berwarna hitam. Domba yang berwarna putih adalah domba milik pemain, sedangkan yang berwarna hitam merupakan milik lawan. Pemain dapat melihat isi keseluruhan antrian domba yang dimiliki, namun hanya dapat melihat satu domba pada antrian domba milik lawan. Selanjutnya, pemain dapat meletakkan domba yang berada pada antrian paling bawah (yang terdapat *border* lingkaran berwarna putih-biru) pada salah satu lintasan dengan mengetuk layar pada posisi lintasan.



[7] Gambar 2.2

Pada Gambar 2.2, *border* putih-biru pada domba paling bawah belum membentuk lingkaran sempurna, ini berarti pemain belum bisa menaruh domba pada lintasan. Pemain harus menunggu sekitar tiga detik agar bisa meletakkan domba pada lintasan. Bar hijau dengan parameter angka di bagian atas dan bawah menunjukkan nyawa yang dimiliki masing-masing pemain.

Prinsip permainan ini adalah *battle*. Domba-domba milik pemain akan saling dorong-mendorong hingga salah masuk ke

dalam *base*. Domba lawan yang berhasil masuk ke *base* akan mengurangi nyawa pemain, begitu juga sebaliknya. Bobot pengurangan yang dimiliki setiap domba berbeda, bergantung pada ukuran domba. Bobot domba dengan pengurangan nyawa berbanding terbalik. Semakin besar domba, maka pengurangan nyawa akan semakin kecil begitupun sebaliknya. Perbandingan bobot domba dengan pengurangan nyawa dapat dilihat pada Gambar 2.3.



[7] Gambar 2.3

Permainan dianggap selesai jika salah satu pemain mampu menguras habis nyawa pemain lainnya.

### III. TEORI KOMBINATORIAL

#### A. Definisi

Kombinatorial adalah cabang matematika untuk menghitung jumlah penyusunan objek-objek tanpa harus mengenumerasi semua kemungkinan susunannya.

#### B. Kaidah Dasar Menghitung

##### 1. Kaidah Perkalian (*rule of product*)

Percobaan 1:  $p$  hasil

Percobaan 2:  $q$  hasil

Percobaan 1 **dan** 2:  $p \times q$  hasil

##### 2. Kaidah Penjumlahan (*rule of sum*)

Percobaan 1:  $p$  hasil

Percobaan 2:  $q$  hasil

Percobaan 1 **atau** 2:  $p \times q$  hasil

#### C. Perluasan Kaidah Dasar Menghitung

Misalkan ada  $n$  percobaan, masing-masing dengan  $p_i$  hasil

##### 1. Kaidah Perkalian (*rule of product*)

$p_1 \times p_2 \times \dots \times p_n$  hasil

##### 2. Kaidah Penjumlahan (*rule of sum*)

$p_1 + p_2 + \dots + p_n$  hasil

#### D. Permutasi

Permutasi adalah jumlah urutan berbeda dari pengaturan objek-objek. Permutasi merupakan bentuk khusus aplikasi kaidah perkalian.

Misalkan jumlah objek adalah  $n$ , maka urutan pertama dipilih dari  $n$  objek, urutan kedua dipilih dari  $n-1$  objek, urutan ketiga dipilih dari  $n-2$  objek

...

urutan terakhir dipilih dari 1 objek yang tersisa.

Menurut kaidah perkalian, permutasi dari  $n$  objek adalah

$$n(n-1)(n-2)\dots(2)(1) = n!$$

### E. Permutasi $r$ dari $n$ Elemen

Permutasi  $r$  dari  $n$  elemen adalah jumlah kemungkinan urutan  $r$  buah elemen yang dipilih dari  $n$  buah elemen dengan  $r \leq n$ , yang dalam hal ini, pada setiap kemungkinan urutan tidak ada elemen yang sama. Persamaan umum permutasi (dilambangkan dengan  $P(n, r)$ ) dapat dilihat pada persamaan 3.1.

$$P(n, r) = n(n-1)(n-2)\dots(n-(r-1)) = \frac{n!}{(n-r)!}$$

Persamaan 3.1

### F. Kombinasi

Bentuk khusus dari permutasi adalah kombinasi. Jika pada permutasi urutan kemunculan diperhitungkan, maka ada kombinasi, urutan kemunculan diabaikan.

$C(n, r)$  dapat dibaca sebagai “ $n$  diambil  $r$ ” yang berarti  $r$  objek diambil dari  $n$  buah objek. Kombinasi  $r$  elemen dari  $n$  elemen atau  $C(n, r)$  adalah jumlah pemilihan yang tidak terurut  $r$  elemen yang diambil dari  $n$  buah elemen. Persamaan umum kombinasi dapat dilihat pada Persamaan 3.2.

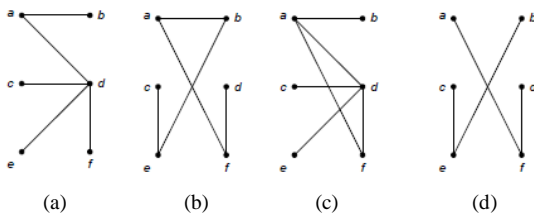
$$C(n, r) = \frac{n(n-1)(n-2)\dots(n-(r-1))}{r!} = \frac{n!}{r!(n-r)!}$$

Persamaan 3.2

## IV. TEORI POHON

### A. Definisi

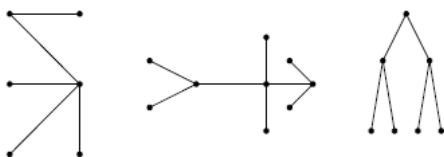
Pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit.



Gambar 4.1

Gambar 4.1 menunjukkan empat buah graf. Graf (a) dan (b) adalah contoh dari pohon, sedangkan graf (c) dan (d) bukan pohon.

Hutan (*forest*) adalah kumpulan pohon yang saling lepas, atau graf tidak terhubung yang tidak mengandung sirkuit. Setiap komponen di dalam graf terhubung tersebut adalah pohon.



Gambar 4.2

Gambar 4.2 merupakan hutan yang terdiri dari 3 buah pohon.

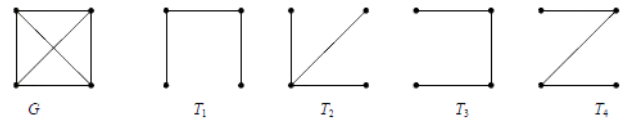
### B. Sifat-sifat (property) Pohon

**Teorema:** Misalkan  $G = (V, E)$  adalah graf tak-berarah sederhana dan jumlah simpulnya  $n$ , Maka, semua pernyataan berikut adalah ekuivalen:

1.  $G$  adalah pohon.
  2. Setiap pasang simpul di dalam  $G$  terhubung dengan lintasan tunggal.
  3.  $G$  terhubung dan memiliki  $m = n - 1$  buah sisi.
  4.  $G$  tidak mengandung sirkuit dan memiliki  $m = n - 1$  buah sisi.
  5.  $G$  tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuatnya hanya satu sirkuit.
  6.  $G$  terhubung dan semua sisinya adalah jembatan
- Teorema 1—6 dapat dikatakan sebagai definisi lain dari pohon.

### C. Pohon Merentang (spanning tree)

Pohon merentang dari graf terhubung adalah upagraf<sup>3</sup> merentang yang berupa pohon. Pohon merentang diperoleh dengan memotong sirkuit<sup>4</sup> di dalam graf. Setiap graf terhubung mempunyai paling sedikit satu buah pohon merentang. Graf tak-terhubung dengan  $k$  komponen mempunyai  $k$  buah hutan merentang yang disebut hutan merentang (*spanning forest*).

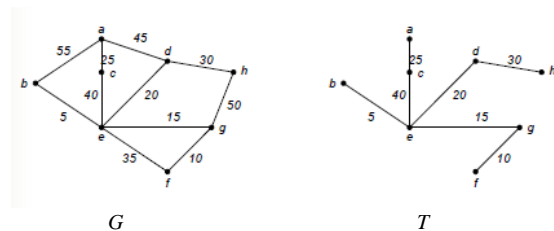


Gambar 4.3

Pada Gambar 4.3,  $T_1, T_2, T_3,$  dan  $T_4$  merupakan pohon merentang dari graf  $G$ .

### D. Pohon Merentang Minimum

Graf terhubung-berbobot<sup>5</sup> mungkin mempunyai lebih dari satu buah pohon merentang. Pohon merentang yang berbobot minimum disebut dengan **pohon merentang minimum** (*minimum spanning tree*).



Gambar 4.4

Pada Gambar 4.4,  $T$  merupakan pohon merentang minimum dari graf  $G$ .  $T$  memiliki bobot total sebesar  $25+40+20+30+15+10+5 = 145$ .

Pohon merentang minimum dapat dibuat dengan dua macam cara, algoritma Prim dan algoritma Kruskal.

#### 1. Algoritma Prim

- i. Ambil sisi dari graf  $G$  yang berbobot minimum,

<sup>3</sup> Bagian dari graf

<sup>4</sup> Graf yang jika ditelusuri lintasannya, maka dapat kembali ke titik awal

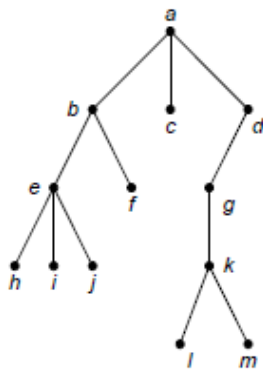
<sup>5</sup> Graf yang memiliki nilai di setiap sisinya

masukkan ke dalam  $T$ .

- ii. Pilih sisi  $(u, v)$  yang mempunyai bobot minimum dan bersisian dengan simpul di  $T$ , tetapi  $(u, v)$  tidak membentuk sirkuit di  $T$ . Masukkan  $(u, v)$  ke dalam  $T$ .
  - iii. Ulangi langkah ii sebanyak  $n-2$  kali.
2. Algoritma Kruskal  
(langkah 0: sisi-sisi graf sudah diurut menaik berdasarkan bobotnya, dari bobot terkecil ke bobot terbesar.)
- i.  $T$  masih kosong.
  - ii. Pilih sisi  $(u, v)$  dengan bobot minimum yang tidak membentuk sirkuit di  $T$ . Tambahkan  $(u, v)$  ke dalam  $T$ .
  - iii. Ulangi langkah ii sebanyak  $n-1$  kali.

### E. Pohon Berakar (rooted tree) dan Terminologinya

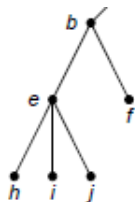
Pohon berakar adalah pohon yang satu buah simpulnya diperlakukan sebagai akar dan sisi-sisinya diberi arah sehingga menjadi graf berarah. Contoh dari pohon berakar dapat dilihat pada Gambar 4.5.



Gambar 4.5

Pohon berakar memiliki terminology, yaitu:

1. Anak (*child* atau *children*) dan Orangtua (*parent*)  
Pada Gambar 4.5,  $b, c,$  dan  $d$  adalah anak-anak dari simpul  $a$ , dan  $a$  adalah orangtua dari anak-anak itu.
2. Lintasan (*path*)  
Pada Gambar 4.5, lintasan dari  $a$  ke  $j$  adalah  $a, b, e, j$ . Panjang lintasan dari  $a$  ke  $j$  adalah 3.
3. Saudara Kandung (*sibling*)  
Pada Gambar 4.5,  $f$  adalah saudara kandung  $e$ , tapi  $g$  bukan saudara kandung  $e$ , karena orangtua mereka berbeda.
4. Upapohon (*subtree*)  
Pohon pada Gambar 4.6 merupakan upapohon dari pohon pada Gambar 4.5.



Gambar 4.6

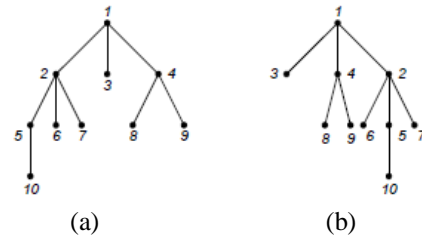
5. Derajat (*degree*)  
Derajat dari sebuah simpul adalah jumlah upapohon (atau jumlah anak) pada simpul tersebut. Pada Gambar 4.5,  $a$  berderajat 3,  $b$  berderajat 2, serta derajat  $c$  dan  $d$  adalah 0. Derajat maksimum dari semua simpul merupakan derajat keseluruhan pohon. Jadi, pada Gambar 4.5, pohon tersebut

berderajat 3.

6. Daun (*leaf*)  
Daun adalah simpul yang berderajat nol (tidak mempunyai anak). Pada Gambar 4.5, simpul  $h, i, j, f, c, l,$  dan  $m$  adalah daun.
7. Simpul Dalam (*internal nodes*)  
Simpul dalam merupakan simpul yang mempunyai anak. Pada Gambar 4.5, yang merupakan simpul dalam adalah simpul  $b, d, e, g,$  dan  $k$ .
8. Aras (*level*) atau Tingkat  
Pada Gambar 4.5, simpul  $a$  beraras 0, simpul  $b, c,$  dan  $d$  beraras 1, simpul  $e, f,$  dan  $g$  beraras 2, simpul  $h, i, j,$  dan  $k$  beraras 3, simpul  $l$  dan  $m$  beraras 4.
9. Tinggi (*height*) atau Kedalaman (*depth*)  
Tinggi atau kedalaman adalah aras maksimum dari suatu pohon. Pada Gambar 4.5, tinggi pohon tersebut adalah 4.

### F. Pohon Terurut (ordered tree)

Pohon terurut adalah pohon berakar yang urutan anak-anaknya penting. Contoh dari pohon terurut terdapat pada Gambar 4.7.

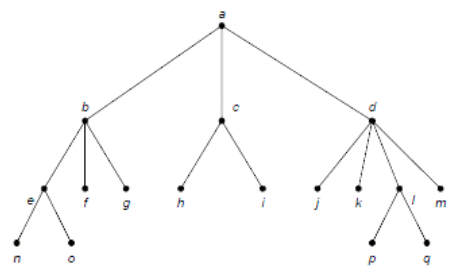


Gambar 4.7

Pada Gambar 4.7, pohon (a) dan (b) merupakan dua pohon terurut yang berbeda.

### G. Pohon $n$ -ary

Pohon  $n$ -ary adalah pohon berakar yang setiap simpulnya memiliki paling banyak  $n$  buah anak. Pohon  $n$ -ary dikatakan teratur atau penuh (*full*) jika setiap simpul cabangnya tepat memiliki  $n$  anak.

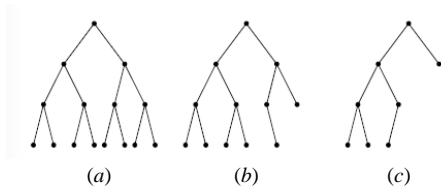


Gambar 4.8

Gambar 4.8 merupakan contoh pohon 4-ary karena simpul  $d$  memiliki jumlah anak anak terbanyak, yaitu 4.

### G. Pohon Biner (binary tree)

Pohon biner adalah pohon  $n$ -ary dengan  $n = 2$ . Setiap simpul pada pohon biner memiliki paling banyak 2 buah anak. Pohon biner seimbang merupakan pohon biner yang tinggi upapohon kiri dan tinggi upapohon kanan seimbang, yaitu berbeda maksimal 1.



Gambar 4.9

Pada Gambar 4.9, pohon biner (a) dan (b) adalah seimbang, sementara pohon biner (c) tidak seimbang.

Pohon biner merupakan pohon yang paling penting karena memiliki banyak penerapan, salah satunya adalah **Pohon Keputusan**.

## V. POHON KEPUTUSAN

Salah satu metode klasifikasi dan prediksi yang sangat kuat dan terkenal adalah Pohon Keputusan. Metode pohon keputusan adalah mengubah fakta yang sangat besar menjadi pohon keputusan yang merepresentasikan aturan. Pohon keputusan dapat digunakan untuk menemukan hubungan tersembunyi antara sejumlah calon variabel *input* dengan sebuah variabel target. Pohon keputusan memadukan antara eksplorasi data dan pemodelan sehingga sangat sesuai jika digunakan sebagai langkah awal dalam proses pemodelan (Kusrini, 2009).

Pohon keputusan dikembangkan untuk membantu dalam mengambil keputusan yang nantinya akan dibuat serangkaian keputusan yang melibatkan peristiwa ketidakpastian. Penggunaan pohon keputusan dapat mengurangi kekacauan potensial dalam suatu masalah kompleks dan memungkinkan pengambil keputusan menganalisis masalah secara rasional (Sutabri, 2005).

### A. Prosedur Pembentukan Pohon Keputusan

Pohon keputusan adalah sebuah pohon dimana setiap simpul merepresentasikan atribut yang telah diuji, setiap cabang merupakan suatu pembagian hasil uji, dan daun merepresentasikan kelompok kelas tertentu. Akar pada pohon merupakan atribut yang memiliki pengaruh terbesar pada suatu kelas tertentu. Umumnya, strategi pencarian solusi yang digunakan dalam pohon keputusan adalah *top-down*. Pada proses mengklasifikasi data yang tidak diketahui, nilai atribut akan diuji dengan cara melacak jalur dari akar sampai daun.

### B. Membangun Pohon Keputusan Menggunakan Algoritma ID3 atau C4.5

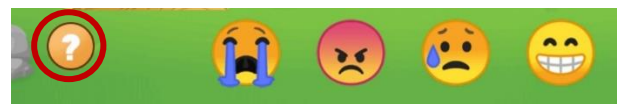
Algoritma ID3 atau C4.5 merupakan algoritma yang diperkenalkan oleh Ros Quinlan. ID3 sendiri merupakan singkatan dari *Iterative Dichotomiser 3* atau *Induction of Decision 3*. Algoritma ID3 membentuk sebuah pohon keputusan dengan metode *divide and conquer* data secara rekursif dari atas ke bawah. Strategi dalam pembentukan pohon keputusan menggunakan algoritma ID3 adalah sebagai berikut:

1. Pohon dimulai sebagai simpul tunggal (akar).
2. Data pada akar akan diukur dengan *information gain* untuk dipilih atribut mana yang akan dijadikan atribut pembagiannya.
3. Sebuah cabang dibentuk dari atribut yang dipilih menjadi pembagi dan data akan didistribusikan ke dalam cabang masing-masing.

4. Algoritma akan terus diulang (rekursif) untuk dapat membuat sebuah pohon keputusan. Ketika sebuah atribut telah dipilih mejadi pembagi atau cabang, maka atribut tersebut tidak diikuti lagi dalam penghitungan nilai *information gain*.
5. Proses pembagian rekursif akan berhenti jika salah satu dari kondisi ini terpenuhi:
  - i. Semua data dari anak cabang telah masuk ke dalam kelas yang sama.
  - ii. Tidak terdapat data pada anak cabang yang baru.

## VI. PENERAPAN PERMUTASI PADA SHEEP FIGHT

Pada permainan *sheep fight*, pemain hanya dapat melihat domba lawan yang terletak pada antrian paling ujung. Hal ini tentu menjengkelkan karena pemain tidak bisa tahu bagaimana susunan antrian domba lawan. Banyak pemain *sheep fight* yang mengira bahwa antrian domba benar-benar diberikan secara acak oleh sistem, sehingga antar pemain tidak bisa memprediksi domba jenis apa yang akan masuk ke dalam antrian. Namun, nyatanya tidak. Algoritma dari game ini akan menyeimbangkan jumlah jenis domba yang akan masuk ke antrian. Dengan asumsi, kedua pemain tidak memberikan *delay* waktu dalam meletakkan domba pada lintasan. Jumlah domba yang telah diletakkan di lintasan oleh masing-masing pemain dapat dilihat dengan cara memencet tombol bulat berwarna kuning dengan simbol “?” di bagian pojok kiri layar, seperti yang ditunjukkan oleh lingkaran merah pada Gambar 6.1.



[7] Gambar 6.1





Setelah mengeklik tombol tersebut, akan ditampilkan informasi berupa tabel yang merepresentasikan jumlah domba per-jenis (dibedakan dengan level) yang telah diletakkan di lintasan oleh masing-masing pemain.

SHEEP COUNTS				
YOU	0	0	0	0
OPPONENT	0	0	0	0

[7] Gambar 6.2

Gambar 6.1 merupakan kondisi awal permainan dimana pemain dan lawan sama-sama belum meletakkan domba pada lintasan. Angka pada tabel akan bertambah sesuai dengan peletakan domba pada lintasan oleh masing-masing pemain. Kolom 1 merepresentasikan jumlah domba level 1, kolom 2 merepresentasikan domba level 2, kolom 3 merepresentasikan domba level 3, dan kolom 4 merepresentasikan domba level 4.



SHEEP COUNTS				
YOU	3	0	0	1
OPPONENT	2	0	1	1

[7] Gambar 6.3

Pada Gambar 6.3, permainan telah berjalan beberapa saat. Tabel pada Gambar 6.3 dapat diartikan sebagai: “Pemain telah meletakkan 3 domba level 1 dan 1 domba level 4 di lintasan. Sedangkan lawan telah meletakkan 2 domba level 1, 1 domba level 3, dan 1 domba level 4 di lintasan.”

Algoritma pada permainan ini adalah menyamakan jumlah jenis domba yang diletakkan oleh masing-masing pemain, atau dengan kata lain, jumlah domba pada baris 1 kolom 1 akan sama dengan domba pada baris 2 kolom 1, dan begitupun seterusnya. Pada Gambar 6.4, jumlah domba pada baris 1 sama dengan baris 2, sehingga dapat dicari kemungkinan kombinasi susunan antrian domba milik lawan.

SHEEP COUNTS				
YOU	5	3	1	2
OPPONENT	5	3	1	2

[7] Gambar 6.4

#### A. Penghitungan Kemungkinan Kombinasi Antrian Domba Milik Lawan

Mencari kemungkinan kombinasi antrian domba lawan adalah dengan menerapkan permutasi. Karena yang sistem adalah antrian (*first in first out*), maka urutan antrian domba tidak dapat diabaikan. Langkah menghitung dan memprediksi antrian domba milik lawan adalah:

1. Ambil  $n$ , yang merupakan  $n$  jenis domba yang jumlahnya kurang dari atau sama dengan milik pemain.
2. Ambil  $r = 2$ , karena slot antrian domba yang tidak diketahui adalah dua.
3. Hitung jumlah kemungkinan kombinasi dengan Persamaan 3.1.
4. Rancang kemungkinan kombinasi.

Contoh kasus dapat dilihat melalui Gambar 6.5.

SHEEP COUNTS				
YOU	7	6	2	2
OPPONENT	8	5	2	2

[7] Gambar 6.5

Penghitungan kombinasi dapat dihitung cara:

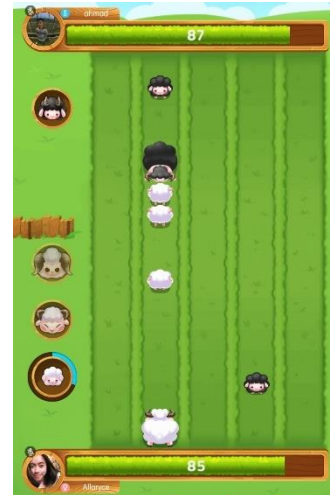
1.  $n = 3$  (karena domba level 2, 3, dan 4 milik lawan  $\leq$  pemain).
2.  $r = 2$ .
3.  $P(n, r) = \frac{n!}{(n-r)!} = \frac{3!}{1!} = 6$

4. Kemungkinan kombinasi dimasukkan ke dalam Tabel 6.1 (pada Tabel 6.1, “lv.” dapat diartikan sebagai “level”).

Kombinasi 1		Kombinasi 2		Kombinasi 3	
lv. 2	lv.3	lv. 3	lv.2	lv. 2	lv. 4
Kombinasi 4		Kombinasi 5		Kombinasi 6	
lv. 4	lv. 2	lv. 3	lv. 4	lv. 4	lv. 3

Tabel 6.1

Kasus menarik dapat terjadi ketika jumlah jenis domba yang dimiliki masing-masing pemain sama, seperti pada Gambar 6.4. Jika hal ini telah terjadi, maka untuk memprediksi kemungkinan kombinasi antrian domba haruslah melihat domba yang berada di antrian paling ujung. Baik domba milik pemain, maupun milik lawan.



[7] Gambar 6.6

Pada Gambar 6.6, dapat dilihat bahwa antrian paling ujung milik pemain adalah domba level 1, sedangkan pada lawan, antrian paling ujung adalah domba level 2. Cara mengkalkulasikan kemungkinan kombinasi menjadi sedikit dimodifikasi, yaitu dengan menambah jumlah domba milik masing-masing pemain dengan 1 sesuai dengan jenis domba yang berada pada paling ujung antrian. Jika mengacu pada Gambar 6.4, jumlah domba level 1 milik pemain menjadi 6 dan jumlah domba level 2 milik lawan menjadi 4. Kemudian dilakukan penghitungan kembali:

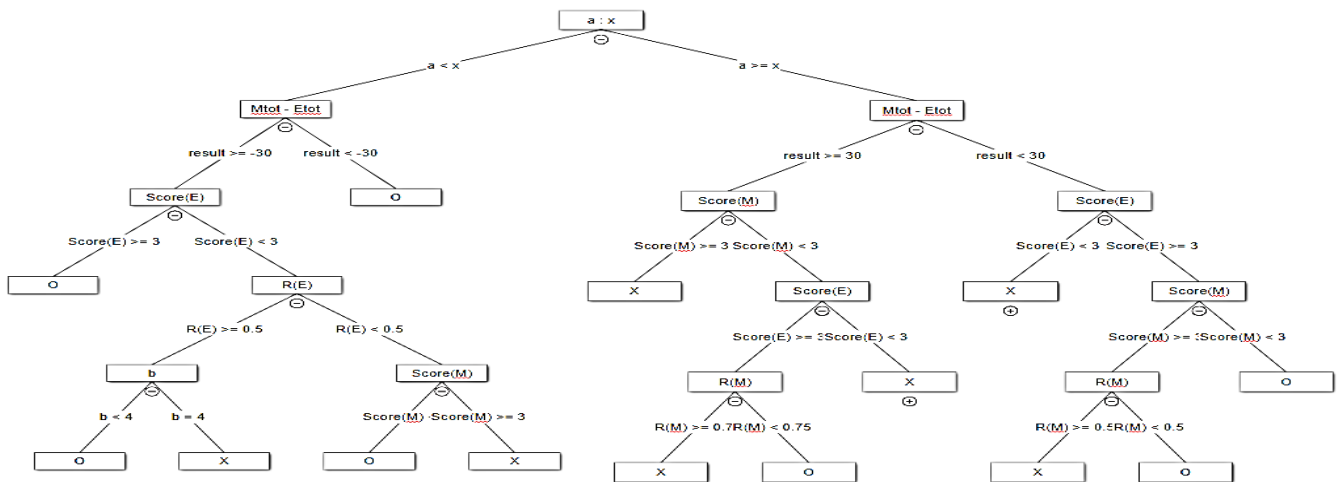
1.  $n = 3$  (karena domba level 1, 3, dan 4 milik lawan  $\leq$  pemain).
2.  $r = 2$ .
3.  $P(n, r) = \frac{n!}{(n-r)!} = \frac{3!}{1!} = 6$
4. Kemungkinan kombinasi dimasukkan ke dalam Tabel 6.2 (pada Tabel 6.2, “lv.” dapat diartikan sebagai “level”).

Kombinasi 1		Kombinasi 2		Kombinasi 3	
lv. 1	lv.3	lv. 3	lv.1	lv. 1	lv. 4
Kombinasi 4		Kombinasi 5		Kombinasi 6	
lv. 4	lv. 1	lv. 3	lv. 4	lv. 4	lv. 3

Tabel 6.2

Dengan mengetahui kemungkinan kombinasi yang mungkin terjadi, akan membantu pemain dalam menentukan langkah selanjutnya. Pemilihan langkah selanjutnya akan dibantu dengan menggunakan **Pohon Keputusan**

### VII. MODEL POHON KEPUTUSAN PADA SHEEP FIGHT



Gambar 7.1

Penting bagi pemain untuk menentukan peletakan domba pada lintasan yang terbaik. Oleh karena itu, penulis mencoba untuk membuat model pohon keputusan pada Gambar 7.1 yang dapat digunakan untuk membantu pemain agar selalu memenangi pertandingan dalam permainan sheep fight.

Pohon keputusan pada Gambar 7.1 memiliki banyak symbol-simbol, yang artinya dapat dilihat pada Tabel 7.1

Simbol	Arti	Contoh
a	Domba yang berada pada ujung antrian domba milik pemain	Pada Gambar 6.6, a bernilai 1 (karena domba pada ujung antrian berlevel 1) Pada Gambar 2.1, a bernilai 2.
b	Domba yang berada pada urutan kedua pada antrian domba milik pemain	Pada Gambar 6.6, b bernilai 3 (karena domba yang berada pada antrian ke-2 berlevel 3)
x	Domba yang berada pada ujung antrian domba milik lawan	Pada Gambar 6.6, x bernilai 2 (karena domba yang berada pada ujung antrian lawan berlevel 2)
Mtot	Total beban domba terbanyak milik pemain yang berada dalam lintasan dengan akumulasi domba lawan + domba pemain terbanyak	Pada Gambar 7.2, Mtot = 110

Etot	Total beban domba terbanyak milik lawan yang berada dalam lintasan dengan akumulasi domba pemain + domba lawan terbanyak	Pada Gambar 7.2, Etot = 110
Score(Q)	Level domba terbesar, Q dapat disubstitusi dengan E yang berarti <i>Enemy</i> (lawan), dan M yang berarti <i>Me</i> (pemain)	Score(E) pada Tabel 6.2 adalah kemungkinan level domba terbesar, yaitu 4. Score(M) pada Gambar 6.6 adalah 4, karena level domba terbesar yang berada pada antrian adalah level 4.
R(Q)	Perbandingan antara panjang total lintasan dengan posisi terkini Q. Q dapat disubstitusi dengan E dan M. E berarti <i>Enemy</i> (lawan), sedangkan M berarti <i>Me</i> (pemain).	Pada Gambar 7.2, pada lintasan 3 (tengah), $R(E) = R(M) = 0.5$ . Sedangkan pada lintasan 4 (kedua dari kanan), $R(E) < 0.5$ , dan $R(M) \geq 0.75$ .
Result	Hasil penghitungan operasi matematika	Pada Gambar 7.2, $Result = Mtot - Etot = 0$ .

Tabel 7.1



[7] Gambar 7.2

Pada daun pohon keputusan, terdapat simbol “O” dan “X”. Simbol ini merepresentasikan langkah yang harus diambil oleh pemain dalam meletakkan domba pada lintasan.

“O” berarti *defense*, artinya pemain harus meletakkan domba pada lintasan dengan akumulasi jumlah domba yang minimum. Sedangkan “X” berarti *attack*, artinya pemain harus meletakkan domba pada lintasan dengan akumulasi domba yang maksimum.

Pada Gambar 7.2, jika daun dari pohon keputusan pada Gambar 7.1 bernilai “X”, maka pemain akan meletakkan domba pada lintasan 3, sedangkan jika daun bernilai “O”, maka pemain dapat meletakkan di lintasan 1, 2, 4, atau 5.

Pohon keputusan yang dibuat oleh penulis sudah melalui kalkulasi dan eksperimen pada permainan sheep fight, dengan catatan asumsi, tidak ada *delay*<sup>6</sup> dalam peletakan domba-domba pada lintasan oleh kedua pemain maupun lawan.

Pohon keputusan pada Gambar 7.1 memiliki akar yang membandingkan  $a$  dengan  $x$ . Hal ini merupakan langkah awal yang penting karena pemain hanya dapat melihat nilai  $x$  (antrian domba paling ujung milik lawan). Hal lain yang sangat penting pada pohon keputusan tersebut adalah Score(Q). Score(E) didapatkan dari penghitungan kemungkinan kombinasi antrian domba-domba milik lawan, sedangkan Score(M) dapat langsung didapatkan karena pemain dapat melihat isi antrian domba miliknya. Etot dan Mtot juga haruslah diperhatikan, karena Mtot akan menentukan kekuatan domba-domba milik pemain dalam satu lintasan untuk ‘mendorong’ domba-domba lawan yang berada pada lintasan yang sama, sedangkan Etot adalah kekuatan domba-domba milik lawan yang akan melawan ‘dorongan’ dari domba-domba milik pemain. R(Q), yang berarti rasio jarak juga tidak dapat diabaikan. Pada R(M), semakin besar rasionya, maka akan semakin menambah peluang pemain untuk memenangkan pertandingan. Pemain haruslah berhati-hati terhadap R(E), jika rasio R(E) besar, maka harus meletakkan domba dengan  $level \geq 3$  untuk melawan.

Tanpa menggunakan pohon keputusan, pemain cenderung

meletakkan domba-dombanya pada satu lintasan saja, padahal cara seperti ini justru berisiko. Juga, pemain akan cenderung untuk meletakkan domba besar ( $level \geq 3$ ) pada lintasan dengan  $Mtot - Etot \leq 0$ , padahal sebenarnya, lawan juga akan mendapatkan domba besar pula. Hal ini akan berakibat kekalahan pada pihak pemain.

Jika dalam memainkan sheep fight, pohon keputusan pada Gambar 7.1, diterapkan dengan memperhatikan parameter-parameter yang ada pada simpul-simpul pohon, maka pemain dapat selalu memenangkan pertandingan sheep fight, sehingga dapat menambah *reard* dan *profile rank* dalam Hago juga dapat naik. Jika *profile rank* bertambah, maka akan semakin besar kemungkinan bagi pemain untuk bertemu dan berinteraksi dengan orang-orang baru secara daring.

## VIII. KESIMPULAN

Anggapan masyarakat umum mengenai permainan *sheep fight* selama ini yang mengatakan bahwa domba-domba diberikan oleh sistem game secara acak, sehingga kemenangan bergantung pada keberuntungan adalah salah. Domba-domba diberikan oleh sistem game ini tidak secara acak, namun akan menyamakan jumlah antar pemain, dan jumlah ini akan selalu sama dan seimbang. Algoritma dari game ini memang akan memberikan secara acak, namun hanya ketika kondisi seimbang antar kedua pemain telah tercapai.

Karena itulah, digunakan teori kombinatorial dan pohon keputusan untuk menentukan langkah terbaik yang harus diambil oleh pemain agar kemenangan dapat selalu diraih.

Kombinatorial digunakan untuk menentukan Score(E) yang akan digunakan dalam pohon keputusan.

## IX. UCAPAN TERIMAKASIH

Pertama-tama, penulis mengucapkan syukur kepada Allah swt. yang telah mencurahkan rizki dan rahmatnya sehingga penulis dapat menyelesaikan makalah ini tepat waktu. Penulis juga berterimakasih kepada orangtua penulis yang telah memberikan dukungan dalam proses penyusunan makalah ini. Tak lupa, ucapan terimakasih ditujukan kepada Dosen Pengampu Mata Kuliah Matematika Diskrit K-03, Bapak Judhi Santoso yang telah memberikan ilmu yang tak terkira berharganya. Terimakasih juga penulis sampaikan kepada teman-teman penulis yang selalu ada untuk memberikan semangat dalam mengerjakan makalah ini.

Akhir kata, penulis memohon maaf karena makalah ini belum sempurna dan masih terdapat kekurangan baik yang disengaja maupun yang tidak disengaja. Semoga ke depannya makalah ini dapat dikembangkan lebih lanjut dan memberikan model yang efektif dan efisien.

## X. KRITIK DAN SARAN

### A. Kritik

Dalam penyusunan makalah ini, penulis melakukan eksperimen dengan cara memainkan permainan *sheep fight*

<sup>6</sup> Ketika border biru-putih pada antrian domba paling ujung telah membentuk lingkaran sempurna, kedua pemain langsung meletakkan domba pada lintasan, tanpa memberikan jeda.



secara terus menerus agar bisa menemukan pola dari pemberian domba oleh sistem. Hal ini dirasa kurang efektif karena memakan banyak waktu. Selain itu, penulis belum bisa menunjukkan kemungkinan kombinasi dan permodelan pohon keputusan apabila salah satu atau kedua pemain memberikan *delay* dalam peletakan domba-domba dalam lintasan.

#### A. Saran

Untuk kedepannya, pola dapat ditentukan lebih cepat apabila penulis mempunyai *source code* dari aplikasi atau permainan terkait, sehingga tidak diperlukan waktu yang terlalu lama untuk menemukan pola dari permainan ini. Selain itu, makalah ini juga dapat dikembangkan dengan cara menambahkan kemungkinan jika pemain memberikan *delay* terhadap peletakan domba pada lintasan. Memang kemungkinan kombinasi antrian domba milik lawan akan semakin banyak dan bervariasi, karena tidak akan mungkin tercapai kondisi seimbang (dimana jumlah jenis domba akan sama pada pemain dan pada lawan), serta pohon keputusannya akan semakin rumit dan memiliki kedalaman yang tinggi, namun ini akan membantu pemain untuk mendapatkan kemenangan absolut.

#### REFERENSI

- [1] Munir, Rinaldi, Matematika Diskrit. Bandung : Penerbit Informatika.
- [2] Witten, Ian H. dan Eibe Frank. 2005. Data Mining: Practical machine learning tools and techniques, 2nd Edition. Morgan Kaufmann. San Francisco.
- [3] Santosa, Budi. 2007. Data Mining : Teknik Pemanfaatan Data untuk keperluan Bisnis. Graha Ilmu. Yogyakarta.
- [4] Turban, Efraim, Decision Support Systems and Intellegents System, 9th Edition, Pearson/Prentice Hall, 2011.
- [5] Nugroho, Antonius, Definisi Game dan Jenis-Jenisnya. <https://chikhungunya.wordpress.com/2011/05/26/definisi-game-dan-jenis-jenisnya/> (diakses 7 Desember 2018).
- [6] <https://www.quora.com/How-many-new-apps-are-added-to-Google-Play-everyday> (diakses 7 Desember 2018)
- [7] HAGO. <https://play.google.com/store/apps/details?id=com.yy.hiyo&hl=en> (diakses 7 Desember 2018)

#### 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 2018



Ainun Fitryh Vianiryzki 13517057