

Aplikasi Algoritma Greedy dalam Strategi Permainan Papan Gobblet Gobblers

Damianus Clairvoyance Diva Putra - 13520035

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13520035@std.stei.itb.ac.id

Abstrak—Gobblet Gobblers merupakan permainan papan dua pemain yang serupa dengan Tic-Tac-Toe. Pada setiap giliran, pemain harus memindahkan bidaknya, yang memiliki tiga ukuran berbeda, ke papan permainan, sedemikian sehingga pemain yang membentuk tiga bidak berderet (*three-pieces-in-a-row*) terlebih dahulu akan memenangkan permainan. Salah satu pendekatan dalam penyusunan strategi permainan ini ialah algoritma Greedy.

Kata kunci—Gobblet Gobblers, Greedy, Tic-Tac-Toe

I. LATAR BELAKANG

Gobblet Gobblers merupakan permainan papan (*board game*) dua pemain yang dirilis oleh Blue Orange Games pada tahun 2009, sebagai edisi revisi dari Gobblet Junior yang diluncurkan enam tahun sebelumnya oleh Gigamic dan Blue Orange Games. Selain perubahan desain produk, tidak terdapat perbedaan signifikan lain antara kedua permainan tersebut. Gobblet Junior sendiri merupakan versi lebih lugas (dengan papan berukuran 3x3 dan bidak sebanyak 12) dari pendahulunya, Gobblet (dengan papan berukuran 4x4 dan bidak sebanyak 24). Sebagai informasi sejarah tambahan, Gobblet didesain oleh Thierry Denoual dan dipublikasikan pertama kali pada tahun 2001, juga oleh Gigamic dan Blue Orange Games.

Gobblet Gobblers menarik karena setiap pemain tak hanya harus mengandalkan strategi, tetapi juga analisis dan memori. Berkat keunikan itu, permainan papan ini berhasil menyabet sejumlah penghargaan, seperti Oppenheim Gold Seal, Dr. Toy 10 Best Games, dan NAPPA Gold Award.

Seperti permainan papan lainnya, penyusunan strategi Gobblet Gobblers dapat memanfaatkan berbagai pilihan algoritma, mulai dari Greedy, BFS, DFS, Backtracking, hingga Branch-and-Bound. Akan tetapi, untuk makalah ini, hanya akan dikaji algoritma Greedy. Pertimbangannya ialah algoritma Greedy merupakan salah satu strategi yang paling mudah diimplementasikan oleh pemain di tengah permainan. Pemain hanya bertugas memilih langkah terbaik dari setiap kondisi pada setiap giliran, tanpa harus menelusuri kemungkinan langkah lawan yang mengharuskan pembangkitan pohon ruang status. Maka, pemain tidak memerlukan komputasi waktu dan ruang yang terlalu besar, apalagi tanpa bantuan program komputer.



Fig. 1. Permainan Papan Gobblet Gobblers. (Sumber: ultraboardgames.com)

II. DASAR TEORI

A. Tic-Tac-Toe

Pada makalah ini, permainan papan Tic-Tac-Toe akan sedikit dikaji karena kedekatan hubungan kekerabatannya dengan Gobblet Gobblers. Dalam tulisannya, Ostermiller menyatakan bahwa terdapat empat tipe, atau tepatnya level, pemain Tic-Tac-Toe, dengan karakteristik langkahnya masing-masing, yang dijabarkan sebagai berikut.

- Pemain *Novice* (pemula): memilih sel secara acak.
- Pemain *Intermediate* (menengah): memilih sel yang menghalangi kemenangan lawan.
- Pemain *Experienced* (ulung): memilih sel pertama yang tidak mengarah pada kekalahan pemain.
- Pemain *Expert* (ahli): tidak akan pernah kalah.

Ostermiller juga melakukan eksperimen permainan antara setiap kombinasi pasangan sebanyak masing-masing 1000 babak. Hasilnya menunjukkan peluang permainan berakhir seri yang tinggi. Bahkan, apabila kedua pemain mengetahui strategi optimal, permainan akan selalu berakhir seri. Gobblet Gobblers hadir untuk mengatasi kelemahan Tic-Tac-Toe tersebut, terlebih karena kemungkinan langkah yang terbatas. XKCD menggambarkan kemungkinan kombinasi permainan Tic-Tac-Toe sebagai pemain giliran pertama (X) dan pemain giliran kedua (O) dalam ilustrasi berikut.

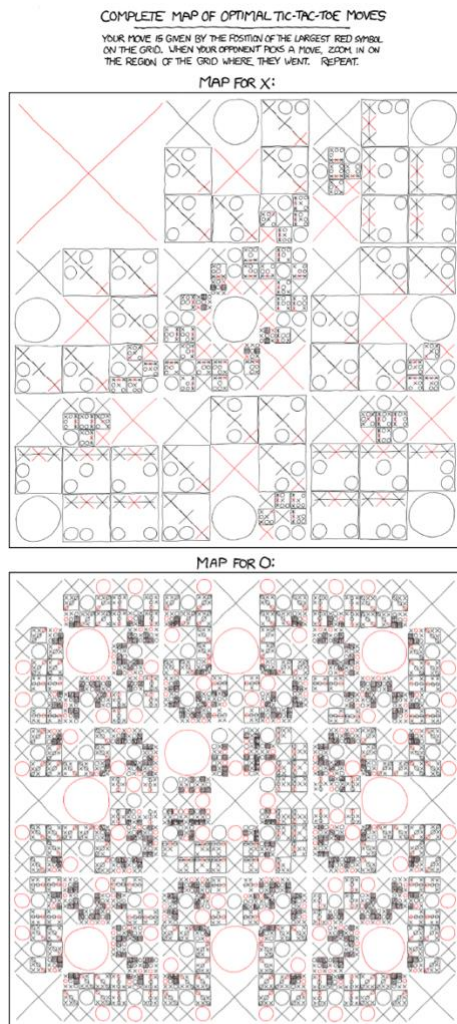


Fig. 2. Kemungkinan Kombinasi Permainan Tic-Tac-Toe sebagai Pemain Giliran Pertama (X) dan Giliran Kedua (O). (Sumber: xkcd.com)

B. Gobblet Gobblers

Satu paket permainan Gobblet Gobblers terdiri atas sebuah papan permainan berukuran 3x3 (9 sel), 6 bidak milik pemain pertama, dan 6 bidak milik pemain kedua. Dari 6 bidak tersebut, 2 berukuran kecil, 2 berukuran sedang, dan 2 berukuran besar. Bidak selanjutnya disebut juga dengan Gobbler. Sama seperti Tic-Tac-Toe, tujuan seorang pemain ialah meletakkan bidaknya ke papan permainan sedemikian sehingga tersusun tiga bidak berderet (*three pieces in a row*), baik secara horizontal, vertikal, maupun diagonal, sebelum lawan berhasil melakukan hal yang sama. Langkah terakhir dari pemain yang berhasil memenangkan permainan (menyusun tiga bidak berderet pertama) akan selanjutnya disebut langkah kemenangan. Permainan berjalan secara bergiliran antara pemain pertama dan kedua, dengan memilih salah satu langkah sah berikut.

(1) Place:

Pemain dapat meletakkan Gobbler dari tangannya ke salah satu sel dalam papan permainan, dengan syarat Gobbler lebih besar dari Gobbler yang sudah ada di sel tersebut. Langkah menumpuk Gobbler lawan dengan Gobbler pemain selanjutnya disebut juga *gobble*.

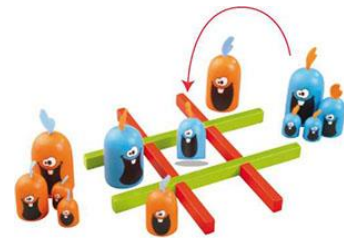


Fig. 3. Langkah Place Gobblet Gobblers. (Sumber: ultraboardgames.com)

(2) Move:

Pemain dapat memindahkan Gobbler miliknya yang sedang aktif (belum di-*gobble* oleh Gobbler lain) pada papan permainan ke salah satu sel lain pada papan permainan, juga dengan syarat Gobbler lebih besar dari Gobbler yang sudah lebih dahulu ada di sel tersebut.

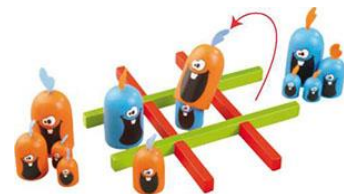


Fig. 4. Langkah Move Gobblet Gobblers. (Sumber: ultraboardgames.com)

Permainan ini disebut melibatkan memori karena lawan bisa jadi memenangkan pertandingan ketika pemain tanpa sengaja memindahkan Gobbler-nya, sedemikian sehingga Gobbler di bawahnya (yang telah di-*gobble* sebelumnya) membentuk tiga bidak berderet. Oleh karena itu, terdapat peraturan bahwa pemain harus melangkah dengan Gobbler yang pertama kali disentuh dengan melakukan *place* atau *move* ke sel lainnya.

C. Algoritma Greedy

Algoritma Greedy adalah algoritma yang menyelesaikan persoalan secara langkah per langkah (*step-by-step*) sedemikian sehingga pada setiap langkah, diambil pilihan terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan (prinsip "*take what you can get now!*") dan diharapkan setiap pilihan optimum lokal berakhir dengan optimum global. Dalam algoritma Greedy, terdefinisi enam elemen dengan penjelasannya sebagai berikut

(1) Himpunan kandidat (C):

Berisi kandidat yang akan dipilih pada setiap langkah, misalnya sisi graf, objek, atau langkah permainan.

(2) Himpunan solusi (S):

Berisi kandidat yang sudah dipilih.

(3) Fungsi solusi (*solution function*):

Menentukan (menghasilkan nilai *boolean*) apakah seluruh kandidat dari himpunan kandidat yang telah dimasukkan ke dalam himpunan solusi sudah memberikan solusi.

(4) Fungsi seleksi (*selection function*):

Memilih kandidat dari himpunan kandidat berdasarkan strategi Greedy tertentu.

(5) Fungsi kelayakan (*feasibility function*):

Menentukan (menghasilkan nilai *boolean*) apakah kandidat yang dipilih oleh fungsi seleksi layak dimasukkan ke dalam himpunan solusi.

(6) Fungsi objektif (*objective function*):

Memaksimalkan atau meminimalkan parameter tertentu dalam keseluruhan algoritma.

Untuk memperjelas pembahasan, berikut merupakan skema umum strategi Greedy dalam notasi algoritma.

Skema Umum Algoritma Greedy

```
function Greedy(C: himpunan_kandidat) ->
himpunan_solusi
```

Deklarasi

```
x : kandidat
S : himpunan_solusi
```

Algoritma

```
S <- {}
while (not SOLUSI(S)) and (C ≠ {}) do
    X <- SELEKSI(C)
    C <- C - {x}
    if LAYAK(S ∪ {x}) then
        S <- S ∪ {x}
    endif
endwhile
{ SOLUSI(S) or C = {} }

if SOLUSI(S) then
    return S
else
    { tidak ada solusi }
endif
```

Sesuai namanya, pada setiap langkah, algoritma Greedy menyeleksi setiap kandidat secara tamak berdasarkan parameter tertentu untuk menghasilkan optimum lokal, sehingga akhirnya dapat menghasilkan optimum global. Akan tetapi, optimum global ini belum tentu merupakan solusi optimal karena setiap kandidat bukanlah merupakan hasil operasi yang menyeluruh terhadap semua kemungkinan solusi (seperti *exhaustive search*). Selain itu, fungsi seleksi yang disusun bisa jadi bukanlah formula yang tepat untuk menghasilkan solusi optimal.

III. PEMBAHASAN

Untuk mengilustrasikan pembahasan selanjutnya, berikut merupakan visualisasi sederhana Gobblet Gobblers, yang terdiri atas papan berukuran 3x3, 6 lingkaran berwarna merah sebagai bidak pemain pertama, dan 6 lingkaran berwarna biru sebagai bidak pemain kedua. Ukuran Gobbler ditandai dengan ukuran lingkaran beserta penomoran di atasnya: 1 (kecil), 2 (sedang), dan 3 (besar). Sel juga dinomori 1 sampai 9. Nomor 1, 3, 7, dan 9 disebut sel sudut; nomor 2, 4, 6, dan 8 disebut sel tepi; dan nomor 5 disebut sel tengah.

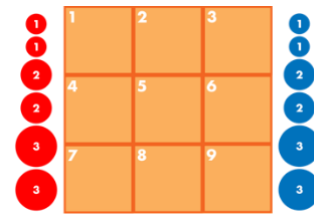


Fig. 5. Ilustrasi Gobblet Gobblers. (Sumber: Arsip Penulis).

A. Wawasan Gobblet Gobblers

Seperti permainan papan lainnya, pemain Gobblet Gobblers harus berusaha mengurangi kemungkinan kekalahan atau meningkatkan kemungkinan kemenangan pada setiap langkah.

Berdasarkan pengalaman, diskusi dalam forum Board Game Geek, dan ilustrasi Fig 2 di atas, dapat diekstraksi wawasan (*insight*) bahwa pemain dapat meningkatkan kemungkinan kemenangan dengan menyusun Gobblet sedemikian sehingga pada giliran selanjutnya, pemain memiliki kesempatan untuk membentuk tiga bidak berderet melalui lebih dari satu konfigurasi. Konfigurasi tersebut selanjutnya akan disebut *fork*. Dengan begitu, lawan tidak akan mampu menghalangi (menge-*block*) langkah kemenangan pemain [Fig 6]. Namun, perlu diperhatikan bahwa sebagai syarat mutlak terbentuknya *fork*, bidak konektor, seperti yang ditunjuk tanda panah pada Fig 6, harus merupakan Gobblet ukuran besar (3) agar tidak dapat di-*gobble* Gobblet lawan. Strategi ini selanjutnya akan disebut Greedy by Fork.

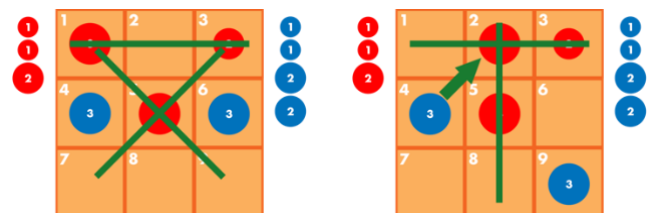


Fig. 6. Ilustrasi Fork: Tripe Fork (3 kemungkinan tiga bidak berderet sekaligus, kiri) dan Double Fork (2 kemungkinan tiga bidak berderet sekaligus, kanan). (Sumber: Arsip Penulis).

Sebaliknya, pada strategi lainnya, pemain berfokus menge-*block* langkah lawan semaksimal mungkin. Implementasi lanjutannya ialah pemain harus berusaha untuk meng-*gobble* Gobblet lawan dalam rangka mengurangi ruang gerak lawan. Strategi ini selanjutnya akan disebut Greedy by Gobblers. Untuk strategi terakhir, pemain akan memperhitungkan keseluruhan faktor, baik *fork* maupun *gobble*. Strategi ini selanjutnya akan disebut Greedy by Formula.

Perlu diperhatikan bahwa untuk ketiga strategi, diasumsikan pemain mampu mengingat Gobblet di bawah setiap Gobblet aktif untuk menyerupai kondisi optimal saja. Berikut merupakan analisis umum elemen algoritma Greedy-nya.

(1) Himpunan kandidat (C):

Berisi seluruh langkah yang mungkin dari setiap Gobblet yang aktif di atas papan dan di tangan.

(2) Himpunan solusi (S):

Berisi seluruh langkah dari himpunan kandidat yang diambil, mulai awal hingga akhir permainan.

(3) Fungsi solusi (*solution function*):

Menentukan apakah Gobbler pemain sudah membentuk tiga bidak berderet.

(4) Fungsi seleksi (*selection function*):

Akan dijelaskan lebih lanjut pada masing-masing pembahasan strategi Greedy.

(5) Fungsi kelayakan (*feasibility function*):

Menentukan apakah langkah yang baru dipilih merupakan langkah yang sah.

(6) Fungsi objektif (*objective function*):

Memaksimalkan kemungkinan kemenangan atau meminimalkan kemungkinan kekalahan.

B. Analisis Strategi Greedy: Greedy by Fork

Algoritma ini lebih mengarah pada strategi agresif (menyerang). Algoritma mengutamakan upaya memperbanyak terbentuknya *fork*. Sebagai bentuk kuantifikasi fungsi seleksi, akan didefinisikan perhitungan kualitas sel dan *fork* dalam bentuk profit. Sel yang lebih baik memiliki kemungkinan konfigurasi jalur tiga bidak berderet yang lebih besar. Perhatikan bahwa urutan posisi dengan kualitas terbaik adalah sel tengah, sudut, lalu tepi, seperti pada ilustrasi berikut.

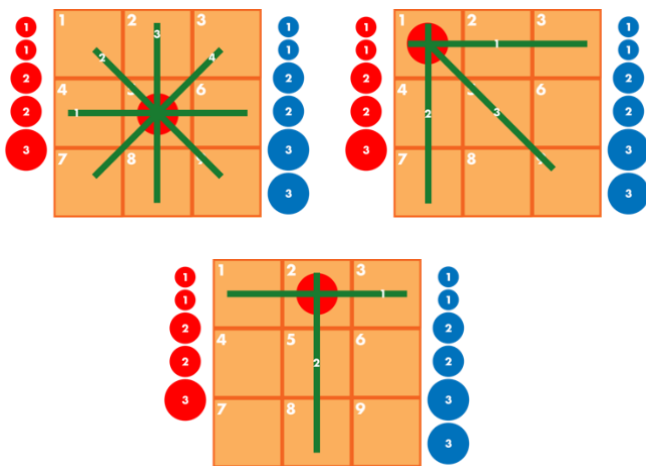


Fig. 7. Ilustrasi Kemungkinan Konfigurasi Tiga Bidak Berderet dari Sel Tengah, Sudut, dan Tepi (Sumber: Arsip Penulis).

Fork yang lebih baik memiliki karakteristik sebagai berikut.

(1) Memiliki kualitas sel yang lebih baik.

Perhatikan juga bahwa untuk mendapatkan kualitas sel terbaik, Gobbler harus diletakkan pada sel dengan jumlah jarak terjauh dari Gobbler pemain di atas papan.

TABLE I. PERHITUNGAN PROFIT KUALITAS SEL

Kondisi	Profit
Move atau <i>place</i> Gobbler ke sel tengah	+8
Move atau <i>place</i> Gobbler ke sel sudut	+6
Move atau <i>place</i> Gobbler ke sel tepi	+4

(2) Mengakuisisi lajur terbuka dan semi-terbuka terbanyak.

Lajur terbuka didefinisikan sebagai kondisi di mana lajur horizontal, vertikal, ataupun diagonal dari suatu sel belum ditempati oleh Gobbler lawan. Namun, walaupun sel pada suatu lajur telah disinggahi oleh Gobbler lawan, selagi itu bukan Gobbler ukuran terbesar (3), sel tersebut masih dapat diakuisisi oleh pemain. Kondisi ini didefinisikan sebagai lajur semi-terbuka.

TABLE II. PERHITUNGAN PROFIT LAJUR TERBUKA/SEMI-TERBUKA

Kondisi	Profit
Setiap sel kosong pada setiap arah lajur	+2
Setiap Gobbler lawan ukuran kecil (1)	-2
Setiap Gobbler lawan ukuran sedang (2)	0
Setiap Gobbler lawan ukuran besar (3)	-4

Langkah dengan bobot yang dianggap tertinggi hingga terendah pada strategi ini tersaji sebagai berikut.

- (1) Menge-*place* atau menge-*move* Gobbler untuk langkah kemenangan. Profitnya sebesar +18.
- (2) Menge-*place* atau menge-*move* Gobbler ukuran besar sebagai konektor *fork*. Profitnya sebesar +16.
- (3) Menge-*place* atau menge-*move* Gobbler untuk membentuk *fork* dengan kualitas terbesar. Profitnya sebesar perhitungan pada tabel I dan II.
- (4) Memastikan Gobbler tidak di-*gobble* lawan. Profitnya sebesar perhitungan pada tabel III.

TABLE III. PERHITUNGAN PROFIT TER-GOBBLED

Kondisi	Profit
Gobbler pemain ukuran kecil (1)	-3
Gobbler pemain ukuran sedang (2)	-9
Gobbler pemain ukuran besar (3)	0

Maka, fungsi seleksi strategi Greedy ini adalah memilih jumlah profit terbesar dari setiap elemen himpunan kandidat.

C. Analisis Strategi Greedy: Greedy by Gobblers

Algoritma ini lebih mengarah pada strategi defensif (bertahan). Algoritma mengutamakan upaya mempersempit ruang gerak Gobbler lawan. Langkah dengan bobot yang dianggap tertinggi hingga terendah pada strategi ini tersaji sebagai berikut.

- (1) Menge-*block* langkah kemenangan lawan dengan menge-*place* atau menge-*move* Gobbler ukuran besar. Profitnya sebesar +20.
- (2) Meng-*gobble* Gobbler lawan dengan Gobbler yang berukuran satu lebih besar. Profitnya sebesar perhitungan pada tabel IV.

TABLE IV. PERHITUNGAN PROFIT GOBBLE

Kondisi	Profit
Gobbler sedang (2) <i>gobble</i> Gobbler kecil (1)	0
Gobbler besar (3) <i>gobble</i> Gobbler sedang (2)	+9
Gobbler besar (3) <i>gobble</i> Gobbler kecil (1)	+3

- (3) Menge-*place* Gobbler ukuran besar, kecil, lalu sedang. Profitnya sebesar perhitungan pada tabel V. Gunakan tabel I untuk membantu penentuan sel, hanya apabila terdapat kesamaan profit.

TABLE V. PERHITUNGAN PROFIT PLACE

Kondisi	Profit
Place Gobbler pemain ukuran kecil (1)	+4
Place Gobbler pemain ukuran sedang (2)	0
Place Gobbler pemain ukuran besar (3)	+8

- (4) Menghindari perpindahan Gobbler pemain membuka tiga bidak berderet lawan. Profitnya sebesar -20.

Maka, fungsi seleksi strategi Greedy ini adalah memilih jumlah profit terbesar dari setiap elemen himpunan kandidat.

D. Analisis Strategi Greedy: Greedy by Formula

Algoritma ini lebih mengarah pada strategi seimbang, dengan menggabungkan kedua algoritma Greedy di atas. Maka, fungsi seleksinya adalah hasil penjumlahan semua profit pada kedua strategi di atas.

IV. HASIL DAN PEMBAHASAN

Sebagai salah satu contoh kasus, berikut merupakan perbandingan rekomendasi langkah menurut ketiga strategi. Kondisi berikut diperoleh dari pemilihan secara acak untuk langkah pemain (merah) dan lawan (biru). Seluruh Gobbler masih terlihat (belum ada Gobbler yang di-*gobble*) untuk mempermudah ilustrasi. Langkah selanjutnya merupakan giliran merah. Perhitungan tidak dicantumkan demi estetika.

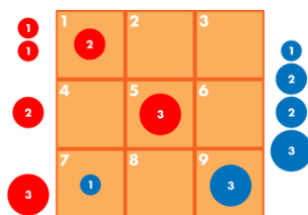


Fig. 8. Ilustrasi Kondisi Suatu Permainan. (Sumber: Arsip Penulis).

TABLE VI. PERBANDINGAN PERHITUNGAN PROFIT GREEDY

Kondisi	Fork	Gobbles	Formula
Place Gobbler ukuran 1 ke sel 2	9	4	13
Place Gobbler ukuran 1 ke sel 3	21	4	25
Place Gobbler ukuran 1 ke sel 4	5	4	9
Place Gobbler ukuran 1 ke sel 6	3	4	7
Place Gobbler ukuran 1 ke sel 8	-1	4	3
Place Gobbler ukuran 2 ke sel 2	3	0	3
Place Gobbler ukuran 2 ke sel 3	15	0	15
Place Gobbler ukuran 2 ke sel 4	-1	0	-1
Place Gobbler ukuran 2 ke sel 6	-3	0	-3
Place Gobbler ukuran 2 ke Gobbler pada sel 7 (<i>gobble</i>)	19	0	19
Place Gobbler ukuran 2 ke sel 8	-7	0	-7
Move Gobbler dari sel 1 ke Gobbler pada sel 7 (<i>gobble</i>)	3	0	3
Place Gobbler ukuran 3 ke sel 2	12	8	20
Place Gobbler ukuran 3 ke sel 3	24	8	32
Place Gobbler ukuran 3 ke sel 4	8	8	16
Place Gobbler ukuran 3 ke sel 6	6	8	14
Place Gobbler ukuran 3 ke Gobbler pada sel 7 (<i>gobble</i>)	28	11	39
Place Gobbler ukuran 3 ke sel 8	6	28	34
Move Gobbler dari sel 5 ke Gobbler pada sel 7 (<i>gobble</i>)	12	3	15
Move Gobbler dari sel 1 ke Gobbler pada sel 2	3	0	3
Move Gobbler dari sel 1 ke Gobbler pada sel 3	-1	0	-1
Move Gobbler dari sel 1 ke Gobbler pada sel 4	-1	0	-1
Move Gobbler dari sel 1 ke Gobbler pada sel 6	-3	0	-3
Move Gobbler dari sel 1 ke Gobbler pada sel 8	-7	0	-7
Move Gobbler dari sel 5 ke Gobbler pada sel 1 (<i>gobble</i>)	8	0	8
Move Gobbler dari sel 5 ke Gobbler pada sel 2	12	0	12
Move Gobbler dari sel 5 ke Gobbler pada sel 3	8	0	8
Move Gobbler dari sel 5 ke Gobbler pada sel 4	8	0	8
Move Gobbler dari sel 5 ke Gobbler pada sel 6	6	0	6
Move Gobbler dari sel 5 ke Gobbler pada sel 7	12	0	12
Move Gobbler dari sel 5 ke Gobbler pada sel 8	2	20	22
Profit Terbesar	28	28	39

Sebagai wawasan dari tabel di atas, perhatikan bahwa Greedy by Fork tidak menghargai penempatan Gobbler ke sel 8 (sebagai langkah bertahan), tetapi justru menilai penempatan Gobbler ke sel 7 lebih tinggi karena membentuk fork yang strategis (sebagai langkah menyerang). Sebaliknya, Greedy by

Gobbles menganggap sel 8 rawan tersusun tiga bidak berderet lawan, sehingga algoritma memilih untuk menempatkan Gobbler terbesar di sana. Namun, bagi algoritma *balanced*, menge-*place* Gobbler ukuran besar (3) untuk meng-*gobble* Gobbler pada sel 7 lebih menguntungkan, karena tidak hanya membentuk *fork* (agresif), tetapi juga menutup langkah kemenangan lawan dan meng-*gobble* Gobbler lawan (defensif).

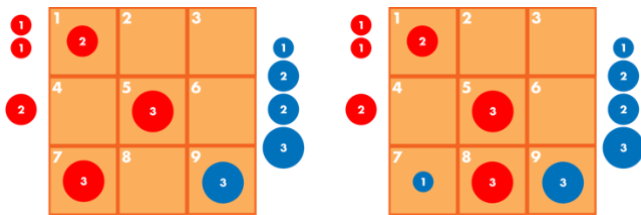


Fig. 9. Ilustrasi Kondisi Permainan Setelah Langkah Greedy by Fork dan Formula (kiri) dan Greedy by Gobbles (kanan). (Sumber: Arsip Penulis).

Untuk menguji kesanggupan (efisiensi) dari strategi Greedy yang telah penulis formulasi, penulis menuliskan program dalam bahasa Python dengan paradigma berorientasi objek. Program ini memang hanya didesain untuk dijalankan melalui Command Line Interface (CLI) karena hanya ditujukan menjadi Minimum Viable Product (MVP) untuk kepentingan pegujian algoritma dalam permainan sesungguhnya.

Sebagai gambaran besar saja, program terdiri atas 8 kelas, yaitu Board, Player, Piece, PiecesOnBoard, PiecesOnHand, Bot, Computer, dan GobbletGobblers. Alur dari programnya ialah pengguna dapat memilih peran sebagai pemain pertama atau kedua, lalu permainan akan berjalan. Pada setiap giliran, program akan memberikan rekomendasi langkah berdasarkan strategi Greedy pilihan (Fork, Gobbles, atau Formula).

```

Output
[redacted]
Current Board:
-----
| | | | |
-----
| | | | |
-----
| | | | |
-----

Pieces on Hand:
1. .
2. .
3. o
4. o
5. 0
6. 0

Active Pieces on Board:
-

Recommendation: piece of index 5 to cell 5
[redacted]

```

Dalam rangka mendukung kemangkusan (efektivitas) pengujian pula, penulis juga menyediakan bot dengan level yang didefinisikan oleh Ostermiller sebagai pemain pemula. Dengan

kata lain, program akan melangkah secara acak, dengan strategi kurang lebih seperti dalam algoritma berikut.

- (1) Pilih salah satu Gobbler secara acak dari Gobbler di tangan dan Gobbler aktif pada papan permainan.
- (2) Pilih salah satu sel secara acak. Apabila langkah tidak sah (Gobbler tidak bisa di-*gobble* oleh Gobbler pilihan), ulangi langkah (2). Apabila langkah tidak sah untuk seluruh kemungkinan sel, ulangi langkah (1).

Satu hal yang perlu diketahui lagi ialah bahwa ketiga strategi Greedy di atas hanya dimaksudkan sebagai panduan penyusunan strategi, artinya perhitungan profit dapat dikalibrasi lagi untuk menghasilkan peluang kemenangan terbesar. Terdapat pula tambahan parameter lain yang bisa diperhitungkan, misalnya perbandingan profit antara langkah *move* suatu Gobbler dengan tidak *move* Gobbler tersebut dan melakukan langkah lain.

Algoritma Greedy ini belum tentu merupakan solusi optimal karena belum dibuktikan secara matematis pada makalah ini. Sebagai pengembangan dari makalah ini, perlu dianalisis kemangkusan algoritma lain, terutama yang melibatkan pohon ruang status, untuk menemukan solusi optimal dari permainan Gobblet Gobblers.

KESIMPULAN

Algoritma Greedy dapat diimplementasikan sebagai salah satu strategi permainan Gobblet Gobblers. Setelah sejumlah alur dan hasil permainan dianalisis, diajukan tiga fungsi seleksi, yakni Greedy by Fork (berdasarkan *fork* terbanyak) Greedy by Gobbles (berdasarkan *gobble* dan *block* terbanyak), dan Greedy by Formula (berdasarkan keduanya). Akan tetapi, algoritma tersebut belum tentu solusi optimal karena perlu adanya pembuktian secara matematis atau pembuktian *counter-example* dari algoritma lainnya.

LINK VIDEO YOUTUBE

Sebagai suplemen penjelasan dalam bentuk verbal, penulis menyediakan video yang dapat diakses pada tautan berikut: <https://youtu.be/5jz51x4TcEk>

PENUTUP

Penulis mengucapkan puji syukur kepada Tuhan Yang Maha Esa karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan makalah ini. Penulis juga tak lupa berterima kasih kepada semua pihak yang terlibat dalam penyusunan makalah ini, baik secara langsung maupun tidak. Penulis mengucapkan terima kasih kepada Ibu Nur Ulfa Maulidevi selaku dosen pengampu mata kuliah IF2211 Strategi Algoritma Kelas 02 Semester II Tahun 2021/2022 yang telah membimbing penulis, serta kepada seluruh keluarga dan teman penulis yang telah mendukung penulis.

Penulis menyadari bahwa masih terdapat banyak kekurangan dalam penyusunan makalah ini. Oleh karena itu, penulis memohon maaf dan akan menerima secara terbuka setiap umpan balik yang membangun. Penulis juga membuka ruang diskusi terkait topik ini melalui pos-el penulis. Akhir kata, penulis berharap makalah ini dapat mendatangkan manfaat bagi banyak orang.

REFERENSI

- [1] Blue Orange Games (n.d.). Gobblet Gobblers. Diakses pada 19 Mei 2022, dari <https://www.blueorangegames.com/index.php/games/gobbletgobblers/>.
- [2] Board Game Geek. (n.d.). Diakses pada 19 Mei 2022, dari <https://boardgamegeek.com/boardgame/13230/gobblet-gobblers/forums/0>
- [3] Munir, R. (2021). Algoritma Greedy (Bag. 1) – Bahan Kuliah IF2211 Strategi Algoritma [slide powerpoint]. Diakses pada 19 Mei 2022, dari [http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-\(2021\)-Bag1.pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag1.pdf)
- [4] Ostermiller, S. (n.d.). Diakses pada 19 Mei 2022, pada <https://blog.ostermiller.org/tic-tac-toe-strategy/>
- [5] Ultra BoardGames (n.d.). Gobblet Gobblers Game Rules. Diakses pada 19 Mei 2022, dari <https://www.ultraboardgames.com/gobblet/gobblers.php>
- [6] xkcd (n.d.). Tic-Tac-Toe. Diakses pada 19 Mei 2022, dari https://xkcd.com/832_large/
- [7] (n.d.). Gobblet Gobblers Board Game Online Version. Diakses pada 19 Mei 2022, dari <https://gobblet-gobblers.netlify.app/>

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, 23 Mei 2022



Damianus Clairvoyance Diva Putra – 13520035