

Algoritma Greedy untuk Pengambilan Keputusan dalam Permainan Kartu Hearts

Kanisius Kenneth Halim (13515008)

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganeca No. 10, Bandung, Jawa Barat

13515008@std.stei.itb.ac.id

Abstraksi—Permasalahan optimasi adalah permasalahan yang sering dihadapi dalam kehidupan sehari-hari kita. Algoritma Greedy adalah algoritma yang cukup mampu merepresentasikan strategi untuk pengambilan keputusan dalam kehidupan sehari-hari, dikarenakan sifatnya yang *irreversible*, seperti layaknya jalannya waktu di dunia nyata. Makalah akan membahas bagaimana algoritma Greedy dapat digunakan untuk pengambilan keputusan dalam permainan kartu Hearts ataupun dapat diimplementasikan menjadi lawan bermain.

Keywords—*Greedy, Himpunan, Solusi, Objektif, Optimal*

I. PENDAHULUAN

Di era teknologi ini, sebuah permainan yang dirancang untuk dimainkan oleh banyak orang dapat dimainkan sendiri dengan kemampuan komputer. Permainan berbasis kartu adalah sebuah permainan yang pada dasarnya menggunakan kartu sebagai media untuk bermain, permainan kartu cukup sering dimainkan oleh kalangan mahasiswa dan dewasa karena dapat dimainkan bersama-sama. Namun, permainan kartu juga turut mengikuti perkembangan teknologi, sekarang setiap orang dapat bermain kartu di dalam ponsel pintar, baik secara *online* bersama orang lain, atau secara *offline* dengan lawannya adalah bot buatan. Permainan Kartu Hearts adalah sebuah permainan yang cukup populer karena permainan ini disediakan oleh sistem operasi Windows dari versi 3.1. Permainan ini merupakan keluarga dari *Bridge*, permainan kartu yang sangat populer di dunia karena permainan ini tidak hanya mengandalkan keuntungan, namun juga diperlukan kemampuan berpikir untuk memenangkan permainan.



Gambar 1 Ilustrasi Permainan Kartu Hearts

Sumber: https://upload.wikimedia.org/wikipedia/en/7/72/Hearts_7.png

II. DASAR TEORI

A. Permainan Kartu Hearts

Permainan kartu Hearts adalah permainan yang cukup sederhana. Seperti yang sebelumnya telah ditulis, permainan ini masih masuk ke dalam keluarga permainan *Bridge*, dimana permainan merupakan *trick-taking game*. *Trick* atau trik adalah satu ronde dimana setiap pemain mengeluarkan satu kartu, lalu pada akhir ronde ditentukan siapa yang akan mengambil trik tersebut.



Gambar 2 Ilustrasi Trik

Sumber: <https://en.wikipedia.org/wiki/Hearts>

Poin dalam permainan kartu *Hearts* adalah suatu hal yang harus dihindari, berbeda dengan kartu *Bridge*. Poin didapatkan dari jumlah kartu hati yang didapatkan oleh pemain dari trik yang dimainkan.

B. Cara Bermain

1. Pembagian Kartu

Dilakukan dengan membagi kartu sama rata terhadap semua pemain, diasumsikan pemain yang ikut bermain 4 orang, sehingga setiap orang mendapatkan 13 kartu.

2. Penukaran Kartu

Sebuah sifat khusus dari permainan Hearts adalah pemain dapat menukar 3 kartu apa saja pada saat awal ronde, pada ronde 1 kartu tukar akan diputar searah dengan jarum jam, ronde 2 kartu tukar akan diputar berlawanan dengan arah jarum jam, ronde 3 kartu akan ditukar dengan pemain yang berhadapan, dan ronde ke 4 tidak ada penukaran kartu.

3. Aturan pada setiap ronde

Permainan dimulai oleh seorang *Lead* yang mengeluarkan kartu pertama pada setiap rondonya. *Lead* bebas mengeluarkan kartu apa saja yang dimilikinya, kecuali kartu dengan simbol ♠, jika belum ada yang mengeluarkan kartu dengan simbol ♠ sebagai kartu yang bukan kartu *Lead*, lalu diikuti oleh orang yang berada di sebelah kirinya, berputar searah jarum jam. Kartu yang dikeluarkan oleh pemain selain *Lead* pada ronde tersebut harus mengeluarkan kartu dengan simbol yang sama dengan kartu *Lead*, dengan pengecualian jika tidak ada kartu dengan simbol yang sama dimiliki oleh pemain tersebut, dalam hal itu pemain bebas mengeluarkan kartu apa saja.

Pemain yang mengambil trik tersebut adalah pemain yang memiliki kartu dengan nilai paling besar dan simbol yang sama dengan kartu *Lead*, kartu dengan simbol berbeda, nilainya diabaikan.

Urutan nilai kartu dimulai dari yang paling tinggi adalah: A K Q J 10 9 8 7 6 5 4 3 2. Tidak ada simbol yang memiliki nilai yang lebih besar dibanding simbol lainnya, alias *no-trump*.

4. Memulai ronde

Setiap ronde dimulai oleh *Lead*, pemain yang menjadi *Lead* pada ronde tersebut adalah pemain yang mengambil trik di ronde sebelumnya, pengecualian untuk ronde pertama, yang menjadi *lead* adalah pemain yang memegang kartu 2♣, dan harus mengeluarkan kartu tersebut.

5. Perhitungan Poin

Poin dihitung dari jumlah kartu dengan simbol ♠ yang diambil pada trik, 1 kartu bersimbol ♠ bernilai 1 poin. Sedangkan kartu Q♠ bernilai 13 poin.

6. Shoot the Moon

Shoot the Moon adalah sebuah kejadian khusus dalam permainan Hearts. Hal ini terjadi ketika ada pemain yang mengambil seluruh kartu bersimbol ♠, dan kartu Q♠. Jika ini terjadi maka pemain tersebut tidak mendapatkan poin, dan seluruh pemain lainnya mendapatkan 26 poin. Namun hal ini sangat terjadi oleh karena itu, Algoritma yang dibahas dalam makalah ini tidak akan membahas tentang kemungkinan Shoot the Moon.

C. Algoritma Greedy

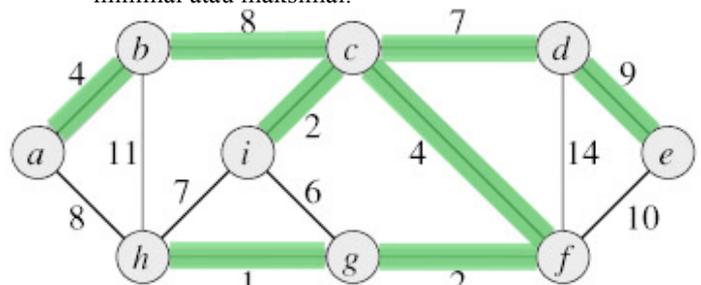
Algoritma Greedy adalah algoritma yang umum digunakan untuk menyelesaikan masalah optimasi, baik itu untuk mencair nilai maksimal atau untuk mencari nilai minimal. Sifat dari algoritma greedy adalah irreversible, jadi ketika langkah sudah diambil, maka tidak mungkin untuk mengembalikan ke state sebelumnya.

Prinsip utama algoritma greedy adalah "take what you can get now", yang berarti mengambil apa yang bias diambil

sekarang, sifat "rakus" ini merupakan ciri utama dari algoritma greedy yang mengakibatkan algoritma ini bias berjalan dengan sangat cepat dibanding algoritma *Exhaustive Search*, namun juga dapat mengakibatkan solusi terjebak dalam optimum lokal yang bukan optimum global

Algoritma Greedy memiliki lima komponen, yaitu:

1. Himpunan Kandidat, himpunan yang berisikan kemungkinan pilihan yang akan diambil.
2. Himpunan Solusi, himpunan yang berisikan solusi yang terpilih dari algoritma greedy.
3. Fungsi Kelayakan, menentukan apakah kandidat tersebut layak menjadi solusi
4. Fungsi Seleksi, fungsi yang menentukan kandidat mana saja yang menjadi bagian dari himpunan solusi
5. Fungsi Obyektif, tujuan dari algoritma greedy tersebut, fungsi obyektif pasti untuk mencari nilai minimal atau maksimal.



Gambar 3 Ilustrasi Algoritma Greedy dalam Mencari Minimum Spanning Tree

Sumber: <http://www.catonmat.net/blog/wp-content/uploads/2008/12/minimum-spanning-tree.jpg>

III. PEMBAHASAN

A. Algoritma Greedy untuk Menentukan Kartu Tukar

Seperti yang dijelaskan sebelumnya, dalam permainan kartu Hearts terjadi penukaran kartu. Berikut adalah analisis Greedy dari fase penukaran kartu

1. Himpunan Kandidat: Seluruh kartu yang dimiliki oleh pemain
2. Himpunan Solusi: 3 kartu yang akan ditukar dengan pemain lain
3. Fungsi Kelayakan: tidak dibutuhkan dalam analisis ini, karena semua kartu layak untuk ditukar
4. Fungsi Seleksi: Memilih kartu dengan nilai paling besar.
5. Fungsi Obyektif: Mendapatkan himpunan kartu dengan nilai yang kecil sehingga meminimalisir kemungkinan untuk mengambil trick.



Gambar 4 Permainan Hearts Windows pada Saat Penukaran Kartu

Sumber: <https://i.ytimg.com/vi/9gEuEnSSR2k/maxresdefault.jpg>

Dengan Strategi greedy demikian pilihan terbaik adalah A♣, K♣ dan A♦ yang ditukar. Dengan melakukan penukaran ini, diharapkan bahwa kartu yang kita miliki memiliki kecenderungan yang lebih kecil untuk mendapatkan poin

B. Algoritma Greedy dalam Permainan

1. Komponen Greedy Umum

Berikut adalah analisis dari komponen greedy dari permainan kartu Hearts

- Himpunan kandidat: Kartu yang dimiliki oleh pemain
- Himpunan Solusi: Kartu yang akan dimainkan pada ronde
- Fungsi Objektif: meminimalisir kemungkinan untuk mengambil trik
- Fungsi Seleksi: terdapat beberapa kasus dalam fungsi seleksi yang akan dijelaskan lebih lanjut dalam makalah ini
- Fungsi Kelayakan: jika terdapat kartu yang dengan simbol sama dengan kartu lead maka harus memainkan kartu dengan simbol yang sama, jika tidak maka semua kartu layak.

2. Strategi saat bermain sebagai Lead

Saat menjadi lead pemain bebas mengeluarkan kartu apa saja selain kartu bersimbol ♥ dan kartu Q♠, kecuali sudah ada yang mengeluarkannya sebelumnya, oleh karena itu fungsi kelayakan berubah tergantung sudah adakah pemain yang mengeluarkan kartu ♥ atau belum.

Ketika bermain menjadi lead maka pemain harus bermain aman, harus ada kartu lawan yang lebih besar dari kartu yang kita keluarkan dengan simbol yang sama. Sebuah kartu dapat dikatakan aman untuk dikeluarkan jika,

1. Kartu tersebut bukan kartu terbesar untuk simbol tertentu dilihat dari himpunan kartu pemain lainnya.
2. Sudah ada pemain yang tidak memiliki kartu dengan simbol tersebut. Misal permainan pada ronde sebelumnya urutan keluar kartu sebagai berikut: 4♦ 6♦ 7♦ 9♥, maka jangan menggunakan kartu ♦ sebagai Lead karena akan dengan mudah disisipkan kartu dengan poin.

3. Jumlah kartu dengan simbol sama pada himpunan kartu pemain lainnya lebih banyak dari 3, 3 dipilih karena 3 merupakan angka batas minimal kartu dengan simbol sama dimiliki oleh pemain lain, namun pada batas hal ini cukup riskan karena kemungkinan salah satu pemain tidak memiliki kartu dengan simbol sama cukup besar. Atau jika jumlah kartu lebih kecil dari 3 namun semua kartu yang dimiliki oleh pemain lain bernilai lebih besar

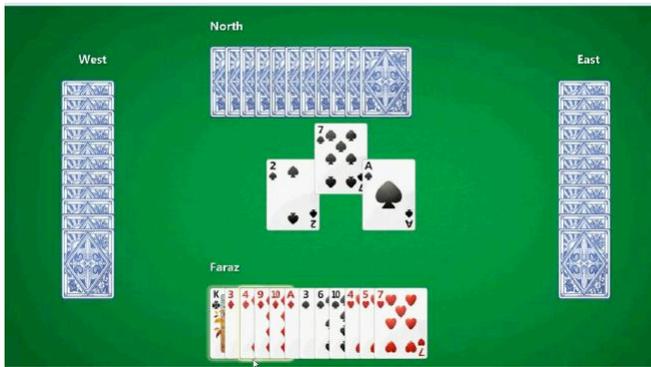
Berikut adalah urutan prioritas yang digunakan untuk menentukan kartu yang akan dikeluarkan sebagai kartu Lead, dengan pertimbangan bahwa setiap kartu yang akan dikeluarkan harus memenuhi ketiga syarat diatas.

1. Kartu dengan simbol yang jumlah kartunya sangat sedikit, hal ini diprioritaskan agar menghabiskan suatu simbol tertentu agar pemain dapat dengan bebas mengeluarkan simbol lainnya.
2. Kartu yang nilainya besar namun masih aman, karena dengan mengeluarkan kartu bernilai besar memperkecil kemungkinan untuk tersudut yaitu kondisi ketika pemain harus mengeluarkan kartu yang tidak aman.
3. Strategi saat pemain menjadi pemain terakhir dalam ronde

Strategi saat menjadi pemain terakhir sangat sederhana, ada 2 kondisi yang terjadi ketika bermain menjadi pemain terakhir, apakah trik tersebut aman untuk diambil atau tidak.

Kondisi aman untuk diambil adalah ketika poin yang terkandung dalam trik tersebut sangat minimal, mendekati nol atau nol, dan pemain memiliki kartu yang aman sesuai dengan standar kartu Lead, karena jika trik diambil maka pemain akan menjadi pemain lead pada ronde selanjutnya.

- Jika aman maka sebaiknya pemain mengeluarkan kartu terbesar untuk simbol yang sama dengan lead, hal ini dilakukan karena kartu bernilai besar harus dibuang dengan cepat.
- Jika tidak aman untuk diambil, maka pemain sebaiknya mengeluarkan kartu dengan nilai terbesar yang lebih kecil dari kartu dengan nilai terbesar pada trik yang sedang dimainkan, jika tidak ada kartu yang lebih kecil dari kartu terbesar dalam trik maka pemain terpaksa mengambil trik dengan memainkan kartu terbesar.



Gambar 5 Kondisi Sebagai Pemain terakhir

Langkah optimal yang harus dilakukan pemain dalam ilustrasi di atas adalah mengeluarkan $10\spadesuit$. Terlihat bahwa trik sebenarnya aman untuk diambil, namun tidak ada kartu yang dapat mengakibatkan pengambilan trik, maka pemain sebaiknya mengeluarkan kartu terbesar yang bias dikeluarkan.

4. Strategi ketika pemain bukan merupakan pemain terakhir dan bukan merupakan Lead

Pengambilan keputusan sebagai pemain tengah adalah strategi pengambilan keputusan yang paling berat, banyak pertimbangan yang dibutuhkan dalam mengeluarkan kartu. Terdapat dua kondisi utama ketika bermain, yaitu ketika pemain memiliki simbol yang sama dengan lead, atau kondisi sebaliknya dimana pemain tidak memiliki kartu dengan simbol yang sama.

- Jika pemain tidak memiliki simbol yang sama maka pemain bebas untuk mengeluarkan kartu apa saja, prioritas dalam pengeluaran kartu sebagai berikut:

1. Kartu $Q\spadesuit$, kartu yang sangat berbahaya jika dimiliki oleh pemain, sehingga jika kartu ini bias dibuang dengan aman, maka kartu ini harus dikeluarkan
2. Jika ada simbol yang jumlah kartu dimiliki oleh pemain hanya satu maka kartu tersebut sebaiknya dikeluarkan karena semakin sedikit simbol yang dimiliki oleh pemain, semakin bebas pemain mengeluarkan kartu
3. Kartu yang bernilai tinggi, kartu dengan nilai tinggi sebaiknya dikeluarkan karena memegang kartu bernilai tinggi cukup berbahaya bagi pemain
4. Kartu bersimbol \heartsuit tertinggi, dengan mengeluarkan kartu bersimbol \heartsuit maka terdapat dua keuntungan bagi pemain, yaitu memberi poin pada lawan bermain, dan menghindari kondisi dimana kartu lead sebuah ronde adalah kartu bersimbol \heartsuit dan pemain hanya memiliki kartu bersimbol \heartsuit dengan nilai besar, sehingga pemain harus mengambil trik yang dapat mengandung poin hingga 4 poin.

- Jika pemain memiliki kartu dengan simbol yang sama dengan kartu lead maka pemain harus mengeluarkan kartu dengan simbol yang sama, maka pemain harus mempertimbangkan keamanan dalam mengeluarkan kartu

tersebut, dalam kondisi ini, kartu dianggap aman ketika memenuhi salah satu kondisi di bawah ini:

1. Kartu bukan kartu tertinggi dalam ronde
2. Kartu yang tersisa dalam himpunan kartu pemain lain bernilai lebih besar dibandingkan kartu yang akan dikeluarkan, hal ini cukup berbahaya karena ada kemungkinan dimana pemain selanjutnya tidak memiliki kartu dengan simbol yang sama.

Kartu yang akan dikeluarkan harus bersifat aman jika mungkin. Kartu yang dikeluarkan harus merupakan kartu dengan nilai tertinggi yang masih tergolong aman.

IV. STRUKTUR DATA

Permainan kartu Hearts seperti pada permainan kartu yang mirip dengan Bridge lainnya, cukup membutuhkan kemampuan pemain untuk mengingat kartu. Dalam analisa kali ini struktur data digunakan untuk menyimpan informasi kartu apa saja yang dimiliki oleh kita sebagai pemain, kartu apa saja yang telah dimainkan, dan kartu apa saja yang dimiliki oleh pemain lainnya. Berikut adalah struktur data yang dibutuhkan dalam melakukan analisis dengan algoritma greedy yang dibahas di makalah ini.

A. SetOfCard

Sebuah Abstract Data Type(ADT) yang digunakan untuk menyimpan seluruh kartu. ADT dibuat menggunakan larik(array) dua dimensi yang berisikan nilai Boolean.

```
typedef bool SetOfCard[4][14];
```

Contoh implementasi tipe data dalam bahasa C++, tanpa menggunakan prinsip pemrograman berorientasi objek.

Pemetaan Nilai Kartu sebenarnya dengan struktur data yang digunakan (SetOfCard[i][j]):

i adalah simbol dari kartu tersebut, dengan pemetaan:

- $\spadesuit \leftrightarrow 0$
- $\heartsuit \leftrightarrow 1$
- $\clubsuit \leftrightarrow 2$
- $\diamond \leftrightarrow 3$

j adalah nilai dari kartu tersebut. Pemetaannya adalah A menjadi 13, K menjadi 12, Q menjadi 11, dan J menjadi 1, selebihnya nilai pada kartu sama dengan nilai pada matriks.

j\i	0	1	2	3
2	1	0	0	0
3	0	0	0	0
4	0	1	0	0
5	0	1	1	0
6	0	0	1	0
7	0	0	0	0

8	0	0	1	0
9	0	1	0	0
10	0	0	0	1
11	0	1	0	0
12	0	1	0	1
13	1	0	0	0
14	0	0	1	0

Contoh data yang menyimpan kartu yang dimiliki oleh pemain. Digit 1 menyatakan punya dan digit 0 menyatakan tidak punya

B. Card

Sebuah tipe data yang dengan atribut Simbol dan Value yang direpresentasikan dengan cara yang sama dengan SetOfCard.

```
typedef struct _Card {
    int Symbol;
    int Value;
}Card;
```

C. Trick

Trick adalah sebuah tipe data yang mengandung informasi kartu apa saja yang dimainkan pada ronde tertentu, berikut adalah contoh implementasi

```
typedef struct {
    Card Lead;
    SetOfCard PlayedCard;
}Trick;
```

V. RANCANGAN IMPLEMENTASI

Makalah ini membahas kemungkinan untuk pengembangan intelegensi buatan (AI) sebagai lawan bermain, sehingga berikut adalah rancangan fungsi yang mungkin akan dibutuhkan ketika program AI dibuat.

A. Fungsi MaxInSet

Fungsi mencari nilai maksimum dari sebuah himpunan kartu. Dengan pseudocode sebagai berikut

B. Fungsi MinInSet

Fungsi yang serupa dengan MaxInSet dengan mencari nilai minimum dalam himpunan kartu.

C. Fungsi CountCardSymbol

Fungsi untuk menghitung jumlah kartu dengan simbol terdefinisi, pada sebuah himpunan kartu

D. Fungsi IsLeadSafe

Fungsi untuk menentukan apakah kartu aman sebagai Lead dalam sebuah ronde.

E. Fungsi IsCardSafe

Fungsi untuk menentukan apakah kartu aman sebagai kartu yang akan dimainkan.

VI. HASIL DAN ANALISIS

A. Hasil Permainan

Hasil dari permainan kartu hearts pada laman Website <https://cardgames.io/hearts/> dengan menggunakan algoritma greedy yang dijelaskan dalam makalah ini, 11 dari 15 ronde menang dan selebihnya kalah, kondisi menang dalam hal ini adalah pemain mendapat nilai terkecil dibanding seluruh pemain lainnya, sedangkan kalah adalah kondisi dimana kondisi menang tidak terpenuhi.

B. Analisis Strategi Pengambilan Keputusan

Strategi pengambilan keputusan (fungsi seleksi) adalah dari algoritma greedy, jika strategi yang digunakan baik maka solusi akan cenderung mendekati kondisi optimum global.

Strategi yang digunakan dalam makalah ini terbukti cukup baik untuk memenangkan mayoritas dari permainan kartu Hearts. Namun, terdapat beberapa kelemahan dari strategi algoritma yang digunakan, seperti:

1. Tidak dapat memprediksi langkah selanjutnya, hal ini biasanya dimiliki oleh AI lainnya, yaitu melakukan komputasi terhadap kemungkinan dan menentukan pilihan terbaik
2. Masih mungkin terjadinya kelalaian dalam permainan yang menyebabkan pemain mendapatkan poin, hal ini pada umumnya karena kekuranganantisipasi terhadap kemungkinan dimana pemain lawan tidak memiliki simbol yang sama dengan kartu lead.
3. Mungkin menyebabkan kejadian dimana pemain hanya memiliki himpunan kartu dengan 1 simbol, dan tidak ada pemain lain yang memiliki simbol yang sama dan pemain menjadi pemain Lead, sehingga akan terjebak menjadi pemain Lead sampai akhir permainan dan harus mengambil semua trik yang dimainkan.
4. Tidak mengantisipasi Shoot the Moon, meskipun jarang terjadi, namun Shoot the Moon mungkin terjadi dan strategi yang digunakan tidak mengantisipasi hal ini.
5. Tidak dapat melakukan pembelajaran mesin, strategi tidak dapat berubah, algoritma greedy tidak mendukung *Machine Learning* sehingga algoritma mudah tertebak oleh lawan main.

Selain itu strategi yang digunakan juga memiliki beberapa keunggulan:

1. Strategi yang digunakan cukup mampu untuk mendapatkan nilai seminimal mungkin karena analisis kondisi dan pengambilan keputusan cukup teliti.
2. Mudah diimplementasikan, dalam pembelajaran membuat AI, terkadang implementasi yang susah cukup menyusahkannya, sedangkan dengan algoritma ini implementasi menjadi kode program dapat dilakukan dengan mudah

3. Cepat dan tidak membutuhkan komputasi berat, jika sebuah AI dibuat untuk berjalan dalam platform *mobile* maka baterai dan penggunaan CPU menjadi hal yang penting, dengan menggunakan algoritma ini, tidak banyak *resource* CPU yang digunakan dan mengakibatkan lebih sedikit menggunakan tenaga baterai.

C. Analisis Algoritma Greedy Dibanding Algoritma Optimasi Lainnya

Algoritma Greedy adalah algoritma yang digunakan untuk mencari kondisi optimal dari suatu permasalahan, permasalahan dari algoritma greedy adalah kemungkinan terjadinya kondisi dimana optimum lokal tidak sama dengan optimum global, hal ini terjadi ketika sebuah strategi greedy tidak dirancang dengan baik atau memang ada kondisi khusus yang menyebabkan algoritma greedy tidak dapat mencapai optimum global, kondisi dimana seharusnya bukan kondisi terbaik yang diambil, namun penyelesaian masalah seperti ini tidak dimungkinkan hanya dengan menggunakan algoritma greedy, dibutuhkan algoritma lainnya untuk menyelesaikan persoalan seperti ini.

Pada umumnya pengguna algoritma greedy menyadari kelemahan algoritma ini dengan sifatnya yang irreversible dan tidak mampu memprediksi kejadian yang ada di depan, namun algoritma greedy adalah algoritma yang sangat banyak digunakan lantaran kelemahannya telah diketahui. Mengapa? Algoritma greedy adalah algoritma yang relatif mudah untuk diimplementasikan dan dapat berjalan dengan waktu singkat, algoritma lainnya yang umum digunakan untuk membuat intelegensi buatan (AI) sederhana adalah algoritma branch and bound dengan pengembangannya seperti A* untuk pencarian rute, atau algoritma pemrograman dinamis mungkin dapat menghasilkan solusi optimum global, tapi pada dasarnya perancangan dan implementasi algoritma relatif lebih susah dibanding algoritma greedy, selain itu juga, algoritma branch and bound adalah algoritma yang membutuhkan *backtrack* atau runut balik, begitu juga dengan algoritma pemrograman dinamis, hal ini tidak selalu mungkin karena pada implementasi dunia nyata, waktu tidak dapat berjalan mundur dan langkah yang sudah diambil tidak dapat dikembalikan, sehingga greedy tetap menjadi salah satu pilihan.

Selain itu, pada algoritma greedy umumnya tidak dipedulikan jumlah aktor yang terlibat dalam sebuah persoalan, tidak seperti algoritma branch and bound yang perhitungan batas (*bound*) harus dapat diprediksi, sehingga jika dalam kasus ini algoritma branch and bound harus melakukan komputasi terhadap setiap kemungkinan yang mungkin terjadi, setiap langkah dari pemain lawan, contoh dalam permainan kartu Hearts ini, jika algoritma branch and bound digunakan maka algoritma harus dapat mencari kemungkinan kartu yang akan dikeluarkan oleh pemain lawan, dimana kondisi ini tidak

diketahui kartu pemain lawan sehingga harus mencoba seluruh kemungkinan kartu yang layak pada masing masing pemain, asumsikan setiap pemain memiliki simbol yang sama dengan kartu yang akan kita keluarkan, maka sebelum mengeluarkan kartu algoritma perlu menghitung sebanyak $13 \times 12 \times 11 \times 10$ kemungkinan jika pemain merupakan lead dalam ronde, sehingga algoritma greedy masih menjadi pilihan untuk membuat AI untuk permainan ini.

KESIMPULAN

Algoritma Greedy adalah algoritma yang populer dan terbukti cukup baik dalam prakteknya. Meskipun tidak selalu menghasilkan solusi yang optimal, terkadang permasalahan optimasi cukup diselesaikan dengan solusi optimum lokal. Algoritma greedy adalah algoritma yang dapat diimplementasikan dalam dunia nyata, kemampuan algoritma greedy yang mungkin belum menyetarai kemampuan otak manusia dikarenakan kemampuan otak yang mampu belajar dari pengalaman dapat diimplementasikan untuk membantu manusia dalam menentukan pilihan melalui komputasi terhadap kondisi. Algoritma greedy merupakan algoritma yang cukup optimal untuk menyelesaikan permainan ini.

REFERENSI

- [1] CB, Bond, "How To Play Hearts", Online, Available: <https://www.familyeducation.com/fun/card-games/how-play-hearts>
- [2] EA, Arneson, "Hearts – Card Game Rule", Online, Available: <https://www.thespruce.com/hearts-complete-card-game-rules-411730>
- [3] AK, Klappenecker, "Theory of Greedy Algorithm", Online, Available: <https://parasol.tamu.edu/~welch/teaching/411.f08/greedy.pdf>
- [4] R. Munir, "Diktat Kuliah IF2211 Strategi Algoritma", Institut Teknologi Bandung, 2007

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, 29 April 2012



Kanisius Kenneth Halim, 13515008