

Aplikasi Algoritma Runut Balik dalam Penentuan Peluang Kemenangan Permainan Kartu Bridge

Aliffiqri Agwar 13517107
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13517107@std.stei.itb.ac.id

Abstract – Aplikasi algoritma runut balik (*backtracing*) sudah sering kita jumpai dalam kehidupan sehari-hari. Biasanya kita jumpai dalam kasus-kasus yang berhubungan dengan *pathfinding* baik itu teka-teki maupun masalah lainnya. Bridge adalah permainan kartu yang mengandalkan strategi namun tidak luput dari probabilitas akan kemenangan. Dalam makalah ini, akan dijelaskan bahwa algoritma *backtrack* dapat menentukan jumlah kemenangan yang dapat diraih dari sekian banyak kemungkinan jalur permainan yang dapat dimainkan.

Kata kunci – Bridge, Runut Balik, Peluang

I. PENDAHULUAN

Permainan kartu merupakan permainan yang sangat umum untuk dijumpai. Tidak hanya kalangan muda, baik orang tua pun sering terlihat memainkan kartu. Permainan kartu memiliki banyak jenis dan aturan permainan. Contoh *capsa* yang menganggap kartu 2 adalah kartu tertinggi, atau *domino* yang mengharuskan pemainnya menghabiskan kartu di tangan lebih cepat daripada pemain lain. Permainan kartu dapat membuat pemainnya merasa terbawa alur permainannya dikarenakan keseruannya.

Salah satu permainan kartu yang cukup terkenal dan namanya dikenal kembali di Indonesia setelah ajang Asian Games 2018 adalah *Contract Bridge* (atau disebut *Bridge* saja). Dibalik permainan kartu yang mengandalkan keberuntungan dikarenakan komposisi kartu sangat random dan sulit untuk diprediksi, nyatanya *Bridge* merupakan salah satu permainan kartu dengan faktor keberuntungan yang rendah dan strategi tingkat tinggi dikarenakan proses penentuan kontrak yang mengandalkan komunikasi.



Gambar 1. Salah satu aplikasi Bridge Solver
Sumber :

<https://chrome.google.com/webstore/detail/bridge-solver/kokhaneonlmbgnlohbmkgeahbj>
(diakses pada 25 April 2019, 19.28 WIB)

Setelah melakukan sistem pelelangan dan penawaran, dilakukanlah permainan. Dari komposisi kartu dan kontrak yang telah ditentukan, kita dapat menentukan berapa peluang pemegang kontrak akan menang.

Makalah ini merupakan lanjutan dari makalah sebelumnya yang berjudul “Penggunaan Pohon Keputusan dalam Pelelangan dan Penawaran Kontrak Permainan Kartu Bridge – Pembukaan Sistem Presisi” dan bersifat eksperimental. Eksperimental dikarenakan penulis membuat meneliti sesuatu yang cukup sulit untuk dicari referensinya

II. DASAR TEORI

A. Algoritma Depth First Search (DFS)

Algoritma DFS adalah algoritma yang mengandalkan konsep rekursif yang biasa digunakan untuk suatu graf. Graf tersebut dimulai dari suatu titik dan akan mengunjungi suatu titik lainnya yang terhubung dengan titik mula-mula. Dari titik yang dikunjungi tersebut, akan mencari titik lain yang terhubung dengan titik tersebut.

Jika lajur yang dilalui sudah dikunjungi, maka akan dilakukan *backtrack* ke titik sebelumnya dan mencari titik lain yang belum dikunjungi dan tetap terhubung pada titik tersebut. Pencarian akan berakhir apabila seluruh kemungkinan jalur telah dikunjungi atau tidak ada lagi titik yang bisa dikunjungi.

B. Algoritma Runut Balik

Algoritma runut balik atau yang biasa disebut *backtracking* merupakan algoritma pengembangan dari *Depth First Search* yang dilakukan pemangkasan dikarenakan DFS memproses seluruh bagian lajur yang mungkin. Pada algoritma runut balik diberlakukan *pruning* (pemangkasan) sehingga solusi yang tidak mungkin dilakukan segera dihapus dari solusi penyelesaian.

Algoritma runut balik memiliki beberapa properti berupa:

a) Solusi persoalan

Merupakan kumpulan *node* yang menjadi solusi dari masalah. Dalam setiap masalah runut balik dapat memiliki satu atau lebih solusi persoalan. Bahkan terdapat masalah yang tidak memiliki solusi.

b) Fungsi pembangkit

Merupakan fungsi yang membuat suatu *node* pada solusi dapat digunakan kembali untuk pencarian solusi berikutnya.

c) Fungsi pembatas

Fungsi ini membatasi dan memutus pencarian solusi dikarenakan sudah tidak cocok dengan permasalahan yang diberikan.

C. Bridge

Bridge merupakan permainan kartu yang menggunakan satu dek kartu tanpa joker dan dimainkan oleh dua pasang pemain yang duduk dengan simbol arah mata angin dimana pemain utara berpasangan dengan pemain selatan, pemain timur berpasangan dengan pemain barat.

Bridge memiliki beberapa istilah. Namuin, pada makalah in, penulis akan memberikan info yang terkait pada konteks yang akan dibawa.

a) Truf (*Trump*)

Truf merupakan jenis kartu yang mengungguli jenis kartu lainnya. Jika kartu biasa diadukan dengan kartu truf, maka kartu truf pasti menang. Namun, apabila kartu truf saling beradu, maka nilai mana yang lebih tinggi akan menang.

b) Trik (*Trick*)

Trik merupakan hasil kemenangan dari satu perputaran. Jika yang menang dari perputaran tersebut adalah kubu utara-selatan, maka mereka mendapatkan 1 trik, begitu pula sebaliknya. Terdapat 13 trik yang dapat diperebutkan oleh kedua kubu.

c) Pelelangan (*Auction*)

Pelelangan merupakan proses penentuan kontrak dimana kontrak yang dihasilkan dipelelangan dijelaskan di poin f.

d) Pembuat kontrak dan Sang Boneka (*Declarer and Dummy*)

Pembuat kontrak merupakan orang yang melakukan penekanan kontrak pada pelelangan dan tidak bisa diganggu lagi oleh lawannya. Pembuat kontrak juga harus merupakan orang pertama dari kubu pemenang kontrak yang menjadi penentu jenis kartu yang dijadikan truf saat pelelangan. Sang boneka merupakan pasangan pembuat kontrak dimana dalam permainan akan dibuka kartunya setelah *defender* yang bertanggung jawab memulai permainan telah memainkan selemba kartu.

e) Penjatuh kontrak (*Defender*)

Penjatuh kontrak merupakan kubu lawan dari pembuat kontrak. Tugas mereka adalah menggagalkan kontrak yang dibuat oleh pembuat kontrak. Penjatuh kontrak sebelah kanan Sang Boneka juga bertanggung jawab untuk memainkan kartu pertama pada putaran pertama.

f) Kontrak (*Contract*)

Kontrak merupakan hasil pelelangan yang dimenangkan oleh *Declarer* dan tidak dibantahkan lagi oleh pasangan lain. Kontrak dapat berupa 1C, 1D, 1H, 1S, 1NT, ... 7C, 7D, 7H, 7S, 7NT. Dalam makalah ini tidak akan dijelaskan mengenai kontrak yang mengalami pengandaan nilai. Arti kontrak tersebut adalah [Jumlah trik yang harus dicapai - 6][Truf yang dimainkan]. Contohnya jika hasil pelelangan menjatuhkan kontrak pada 3H, maka *Declarer* harus mampu meraih 9 trik dengan hati sebagai trufnya.

g) Permainan (*play*)

Proses pencarian trik oleh seluruh kubu. Terdapat 13 ronde untuk setup play.

Bridge memiliki aturan umum sebagai berikut :

1. Masa persiapan dimulai. Empat pemain duduk di suatu tempat dan dinamai dengan arah mata angin. Pemain utara berpasangan dengan pemain selatan, pemain timur berpasangan dengan pemain barat.
2. Kartu dapat ditentukan oleh juri atau dikocok biasa oleh pemain. Kemudian setiap pemain mendapatkan 13 kartu.
3. Pelelangan dimulai. Salah satu pemain dapat memulai (tergantung kesepakatan atau aturan yang berlaku) untuk menyatakan kontrak atau melempar giliran ke pemain yang putaran searah jarum jam dia (selanjutnya disebut *pass*).
4. Jika seseorang telah menyebutkan suatu kontrak, maka kontrak yang lebih rendah dari itu tidak boleh disebutkan kembali di pelelangan tersebut. Hierarki dari kontrak dapat dilihat pada tabel 1 di bawah ini.

1C	1D →	1H →	1S →	1NT ↓
→ 2C →	2D →	2H →	2S →	2NT ↓
→ 3C →	3D →	3H →	3S →	3NT ↓
→ 4C →	4D →	4H →	4S →	4NT ↓
→ 5C →	5D →	5H →	5S →	5NT ↓
→ 6C →	6D →	6H →	6S →	6NT ↓
→ 7C →	7D →	7H →	7S →	7NT

Tabel 1. Hierarki urutan tawaran kontrak

III. STRUKTUR DATA DAN KELAS

Permasalahan ini dapat diselesaikan oleh program. Program tersebut memiliki beberapa kelas yang akan berfungsi sebagai *blueprint* dari objek yang dimainkan. Pada bab ini hanyalah struktur tanpa implementasi kode. Struktur kelas tersebut terdiri dari:

A. Kelas Kartu

```
class Card
{
    public int nilai;
    public char jenis;
    public Card(int nilai, char jenis);
    public bool isEqual(Card a);
}
```

B. Kelas Pemain

```
Class Player{
    Public char kursi; //arah mata angin
    Public List<Card> pegangan;
    //kartu yang dipegang

    Public Card lastPlay;
    //bernilai null apabila belum bermain pada suatu
round
    Public Player(char kursi);
    Public void addKartu(Card A);
    Public Card getKartu(int indeks);
    Public void removeKartu();
}
```

C. Kelas Permainan

```
Class Play{
    Public Player north,south,west,east;
    Public int [] trikNS;
    Public int [] trikWE;
    Public char declarer;
    Public char trump;
    //array dari 0-13 dari suatu permainan. Menyimpan
berapa banyak kejadian dengan kubu tersebut memenangkan
i trik. Dimana i adalah salah satu dari indeks array.

    Public void startGame(string fileinput);
    //bagian ini melakukan load file yang berisi
distribusi kartu dan dipastikan valid.
    Public void playGame(int round, Card first);
    Public char roundWinner(Card northCard, Card
southCard, Card eastCard, Card westCard, char
firstPlayer);
    //mengeluarkan arah mata angin pemenang ronde
}
```

Dalam prosesnya terdapat file yang dijadikan acuan untuk distribusi kartu. Struktur data yang tersimpan sebagai berikut:

```
North - Spade
North - Heart
North - Diamond
North - Clubs
East - Spade
East - Heart
East - Diamond
East - Clubs
South - Spade
South - Heart
South - Diamond
South - Clubs
West - Spade
West - Heart
West - Diamond
West - Clubs
```

5. Pelelangan dihentikan apabila terdapat 3 *pass* secara beruntun (atau 4 *pass* beruntun apabila pemain pertama yang mendapatkan kesempatan menawarkan kontrak melakukan *pass*). Kontrak merupakan tawaran terakhir yang disebutkan.
6. Penentu kontrak dan Sang Boneka ditentukan dari pemenang kontrak dan telah dijelaskan pada poin d).
7. Permainan dimulai, penjatuh kontrak yang berada di sebelah kanan Sang Boneka memainkan kartu pertama. Kemudian Sang Boneka memperlihatkan seluruh kartunya dan telah disusun berdasarkan jenisnya.
8. Kedua kubu akan terus berebut trik. Terdapat 13 trik yang diperebutkan. Setiap putaran, semua pemain wajib memainkan 1 kartu. Pemain yang menjadi pemenang tiap putaran akan memainkan kartu pertama untuk putaran berikutnya.
9. Jika pemain memiliki hak untuk memainkan kartu pertama pada suatu putaran, maka pemain tersebut boleh memainkan kartu apa saja termasuk truf. Seluruh pemain wajib mengikuti jenis kartu yang dimainkan oleh pemain kartu pertama.
10. Apabila kontrak tidak menentukan jenis tertentu sebagai truf, maka kartu yang berbeda jenis dari pemain kartu pertama tiap putaran akan dianggap kartu kalah. Begitu pula apabila jenis tersebut bukan truf pada kontrak truf.
11. Truf yang dimainkan oleh pemain selain pemain kartu pertama hanya dapat diturunkan apabila jenis kartu pemain kartu pertama sudah tidak dimiliki lagi atau memang pemain kartu pertama menurunkan trufnya.
12. Kartu yang sudah dimainkan pada suatu putaran tidak boleh dimainkan lagi pada putaran-putaran berikutnya.
13. Permainan berakhir apabila seluruh kartu telah dimainkan. Kemudian dilakukan penghitungan trik. Pembuat kontrak dan Sang Boneka dinyatakan menang apabila jumlah trik yang didapatkan sesuai atau lebih dari kontrak yang dijanjikan. Penjatuh kontrak menang apabila pembuat kontrak tidak dapat memenuhi kontrak tersebut.

D. Konsep Peluang

Dalam makalah ini, tidak dibahas mengenai macam-macam distribusi yang digunakan. Peluang yang digunakan pada kasus ini adalah peluang yang sederhana yang dirumuskan sebagai berikut:

$$P(x) = \frac{n_{\text{trik} \geq \text{kontrak}}}{n_{\text{total}}}$$

IV. METODE PENYELESAIAN MASALAH

A. Strategi Runut Balik pada Awal Putaran

Setiap pemain yang memainkan kartu pertama setiap putaran akan menentukan *node* akar setiap putaran. Terdapat n cara yang dapat dijalankan dimana n adalah jumlah kartu yang tersisa di tangan.

B. Strategi Runut-Balik pada Pertengahan Putaran

Pemain yang bukan pemain kartu pertama akan menentukan kartu berikutnya yang akan dimainkan. Terdapat m cara yang dapat dijalankan dimana m adalah kartu yang tersisa di jenis tersebut atau n cara apabila nilai m adalah 0 (pemain sudah tidak memiliki jenis kartu yang dimainkan pemain kartu pertama) dimana n adalah sisa kartu yang di tangan.

C. Pengecekan di Akhir Putaran dan Penentuan Pemain Kartu Pertama Putaran Selanjutnya

Apabila setiap pemain telah menurunkan 1 kartu di putaran tersebut, diadakan pengecekan kubu mana yang menang dan pemain manakah yang akan menjadi pemain kartu pertama di putaran selanjutnya.

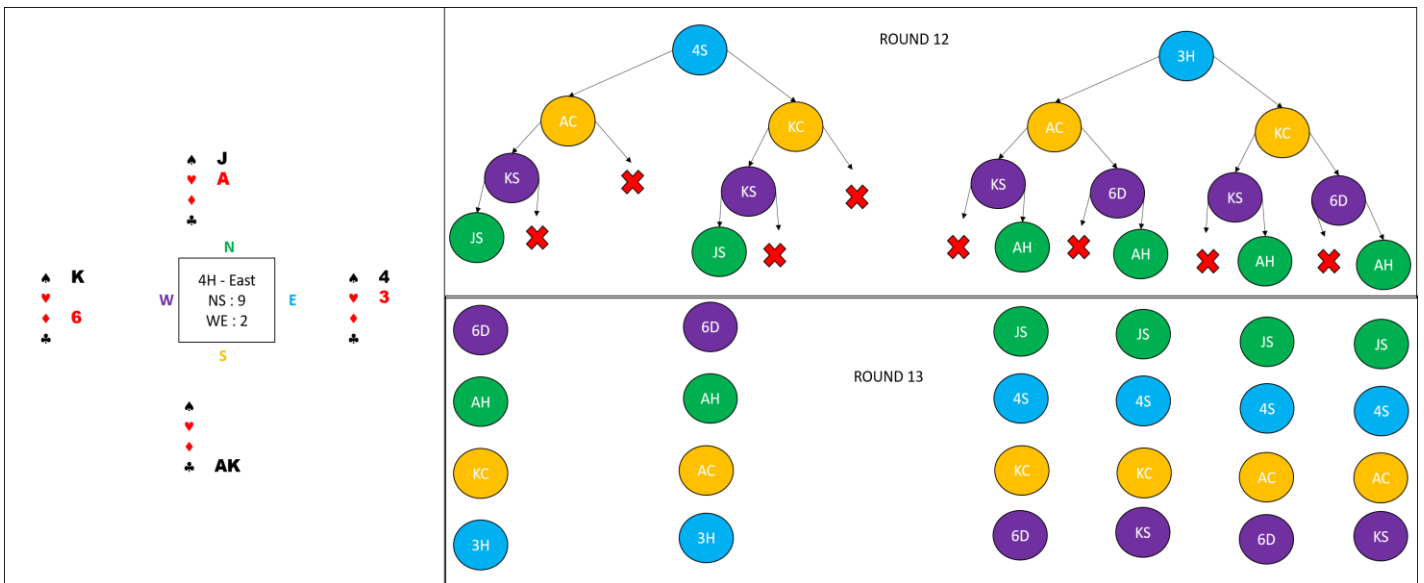
Bagian ini meliputi pemrosesan trif yang dimainkan dan penambahan trik bagi yang memenangkan putaran.

Apabila tidak ada kartu lagi yang dapat dimainkan, proses dihentikan dan dilakukan perhitungan trik yang didapatkan oleh declarer.

D. Perhitungan Akhir

Dari seluruh proses yang terjadi, didapatkan beberapa hasil. Hasil tersebut merupakan solusi permainan yang dapat terjadi. Dari solusi-solusi tersebut, kemudian dihitung banyak solusi yang menghasilkan keberhasilan kontrak oleh declarer. Kemudian dibagi dengan banyaknya solusi yang ada.

Struktur data dan kelas ini kemudian dilakukan *coding* sedemikian rupa agar sesuai dengan metode penyelesaian masalah. Salah satu skema kejadian yang terjadi dapat dilihat pada gambar di bawah ini.



Gambar 2. Contoh kasus yang terjadi pada ronde 12 dan 13 dengan kontrak 4H oleh Utara. (Contoh node : AC = As Club)
Sumber : Dokumentasi Penulis

Pada gambar 2 di atas, terdapat backtracking yang terjadi dikarenakan kartu tersebut tidak boleh dimainkan. Di gambar juga dapat dijelaskan bahwa pada bagian kiri, pemain barat pasti memenangkan ronde 12 sedangkan bagian kanan memenangkan pemain utara.

Dari permasalahan ini didapatkan beberapa fungsi yang digunakan untuk melakukan runut balik. Fungsi-fungsi tersebut merupakan :

1. Fungsi pembatas

Pembatas dari masalah ini adalah validnya kartu yang dimainkan ronde tersebut. Valid tidaknya sebuah kartu bergantung pada kartu pertama yang dimainkan pada ronde tersebut dan kartu yang saat ini pemain pegang.

Jika kartu yang dimainkan tidak cocok dengan kartu pertama di ronde tersebut, padahal masih ada kartu

dengan jenis yang sama dengan yang di tangan, maka dilakukanlah pruning.

Pada gambar 2, kita dapat melihat pada alur 4S ke AC tidak dapat melanjutkannya ke 6D dikarenakan masih terdapat KS yang dapat dimainkan.

2. Fungsi pembangkit

Fungsi pembangkit digunakan dengan cara rekursi sehingga kita tidak perlu melakukan cara *brute force* yang mengulang kembali dari awal pencarian.

Masalah ini memiliki kompleksitas yang sangat tinggi. Dapat dilihat pada awal ronde terdapat n kemungkinan kartu yang menjadi node awal dari backtrack ini.

Setelah n kartu yang dimainkan, terdapat n^3 kemungkinan kartu yang dapat dimainkan. Dengan backtracking, n^3 ini dapat direduksi menjadi $p \times q \times r$ dimana p adalah jumlah kartu sejenis dengan kartu awal permainan dari pemain urutan kedua, begitu pula dengan q untuk pemain ketiga dan r untuk pemain keempat. Namun, jika p (berlaku juga untuk q dan r) bernilai 0, maka harus diganti dengan n .

Setelah dilakukan pencarian solusi pada suatu ronde, masih ada pengecekan pemain mana yang mendapatkan kemenangan. Minimal terdapat $m!$ perbandingan dimana m adalah banyaknya pemain yang ikut.

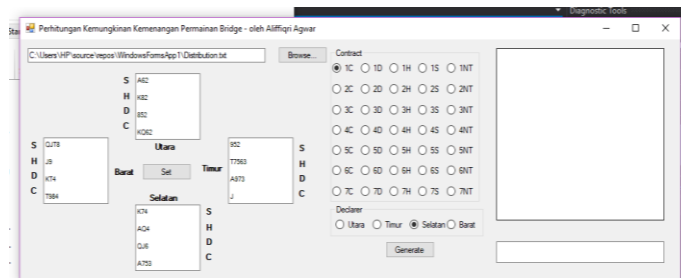
Dari penjelasan di atas dapat dirumuskan sebagai berikut:

$$\begin{aligned} \text{Satu Ronde :} \\ O(n) &= n \cdot f(p) \cdot f(q) \cdot f(r) \\ O(n) &= (n)^4 \\ f(p) &= \begin{cases} p, & p > 0 \\ n, & p = 0 \end{cases} \end{aligned}$$

Dikarenakan terdapat 13 ronde, maka kompleksitas yang terjadi dapat sebesar $O((n!)^4)$. Hal ini sangatlah besar mengingat n yang dipakai adalah 13. Sedangkan $13!$ adalah 6.227.020.800. sedangkan dipangkat 4 (walaupun ini adalah kasus terburuk) akan menjadi $3,87 \times 10^{19}$ kali perhitungan. Jika suatu komputer dapat mengerjakan 1.000.000 pengerjaan dalam 1 detik pun, akan membutuhkan 1244213 tahun untuk menyelesaikan perhitungan.

Kompleksitas sebesar ini awalnya membuat penulis tidak percaya dan ingin membuktikan sendiri apakah benar hingga sebesar itu. Penulis pun membuat suatu program untuk menguji kasus ini dan dijelaskan lebih lanjut di bagian V. Program dibentuk dengan bahasa C# dan kaskas .NET Framework

V. PERCOBAAN DAN ANALISIS

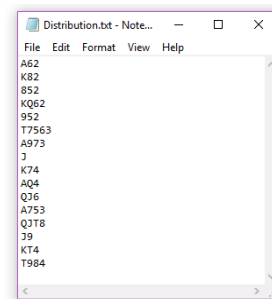


Gambar 3. Tampilan antarmuka aplikasi perhitungan peluang kemenangan dalam distribusi terbuka.
Sumber : Dokumentasi Penulis

Dalam pembuatan makalah ini, penulis sempat membuat program untuk menguji kebenaran kompleksitas sesuai perhitungan di bagian IV.

A. Proses Masukan

Program ini menerima input berupa file text yang formatnya terdiri atas 16 baris dan tersusun atas dek kartu yang valid. Dek kartu yang valid merupakan dek yang tidak ada kartu berulang. Pada 16 baris tersebut, masing-masing baris diwakili oleh data kartu pegangan yang sesuai dengan penjelasan pada bagian III struktur data input kartu.



Gambar 4. Contoh masukan distribusi kartu dari file txt
Sumber : Dokumentasi Penulis

Setelah melakukan input, dilakukan pemrosesan rekursif.

B. Proses Rekursif

Pemrosesan rekursif dilakukan dengan memanggil fungsi *playGame()* dari kelas *Play*. Hal ini melakukan pengecekan siapa yang sedang mendapat giliran bermain dan bagaimana komposisi kartu yang tersisa. Hal ini dilakukan berulang-ulang hingga kartu habis.

Dalam fungsi tersebut dicek apakah seluruh pemain telah bermain atau belum. Apabila seluruhnya telah bermain, akan dilakukan proses perbandingan akhir ronde. Jika belum, akan mengeluarkan salah satu kartu dari pemain yang mendapat giliran lalu melakukan pemanggilan fungsi rekursi.

C. Proses Perbandingan Akhir Ronde

Perbandingan akhir ronde ini melakukan 6 perbandingan dimana pemain utara dibandingkan dengan pemain barat, selatan, dan timur. Pemain timur dibandingkan dengan pemain selatan dan barat. Pemain selatan dibandingkan dengan pemain barat. Proses perbandingan ini beracu pada kartu acuan yang merupakan kartu pertama yang dimainkan.

Perbandingan ditentukan dengan aturan sebagai berikut :

1. Pemain A dan pemain B mengeluarkan truf. Maka pemain dengan truf tertinggi dinyatakan menang.
2. Pemain A mengeluarkan truf tapi pemain B tidak menggunakan truf. Maka pemain A dinyatakan menang. Begitu pula sebaliknya
3. Pemain A dan B tidak memiliki truf. Maka jika A memiliki kartu sejenis dengan kartu acuan dan lebih besar nilainya dari B, maka pemain A dinyatakan menang.

Apabila suatu pemain memiliki kondisi kemenangan lebih besar daripada pemain lain, dia akan menjadi pemenang dari ronde tersebut dan melanjutkan permainan.

D. Proses Output

Pada percobaan ini, tidak ada output yang berhasil dikeluarkan dikarenakan program meminta untuk berhenti dikarenakan telah bekerja lebih dari 1 menit. Sehingga dapat diketahui bahwa kompleksitas program ini sangatlah besar.

E. Algoritma lain yang dapat digunakan

Selain algoritma backtracking ini yang memakan waktu lama, terdapat beberapa algoritma lain yang umumnya digunakan untuk kasus bridge solving. Algoritma tersebut adalah alfa-beta, penggunaan *hash table* dan *ordering-pruning* yang merupakan algoritma rumit dan tidak cukup dijelaskan di makalah ini.

VI. KESIMPULAN

Algoritma backtracking secara teori dapat digunakan untuk menghitung berapa besar peluang yang dimiliki oleh pemain untuk memenangkan pertandingan. Namun, pemrosesan backtracking akan memakan waktu sangat lama dikarenakan kompleksitasnya yang menyentuh $O((n!)^4)$.

Terdapat algoritma lain yang dapat memproses hal semacam ini dengan lebih cepat. Algoritma seperti alfa-beta, penggunaan hash-table serta ordering-pruning dapat menjadi pilihan untuk membuat masalah ini menjadi lebih cepat dalam mendapat solusi.

VII. UCAPAN TERIMA KASIH

Ucapan terima kasih ini saya berikan kepada Unit Aktivitas Bridge yang banyak meminjamkan saya buku-buku untuk dijadikan referensi. Kemudian, permainan di unit ini dapat diamati dan dianalisis sehingga mempermudah saya dalam pengerjaan makalah ini.

REFERENSI

- [1] Chang, Ming-Seng. "Building a Fast Double-Dummy Bridge Solver". <https://pdfs.semanticscholar.org/eed7/025c0ab6e5f693dff53ed7cf2605aa10d768.pdf> diakses pada 18 April 2019, 15.00 WIB
- [2] [http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/BFS-dan-DFS-\(2019\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/BFS-dan-DFS-(2019).pdf). Diakses pada 24 April 2019, 02.53 WIB.
- [3] [http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Runut-balik-\(2018\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Runut-balik-(2018).pdf). Diakses pada 25 April 2019, 14.19 WIB.
- [4] Iskandar. 2014. "Menjadi Declarer Terbaik". Surabaya. Dapat diakses juga pada www.quicktrick.biz.
- [5] Kantar, Eddie dan Edwin. B. Kantar. 1999. "Advanced Bridge Defense". Diakses secara daring di https://books.google.co.id/books?id=rf_wV7GJxRsC&printsec=frontcover&hl=id#v=onepage&q&f=false pada 19 April 2019.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini merupakan tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain dan bukan plagiasi.

Bandung, 26 April 2019



Aliffiqri Agwar

13517107