

# Penerapan Algoritma Greedy pada game *Toraburu Hanafuda Travel Journal*

Iqrar Aminullah / 135616126

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

13516126@std.stei.itb.ac.id

**Abstrak**—Hanafuda adalah sebuah permainan kartu tradisional yang berasal dari Jepang, dimana pemain berusaha untuk mengumpulkan pasangan – pasangan kartu tertentu untuk mengakhiri ronde dan mengumpulkan poin, untuk meraih kemenangan di akhir permainan. Permainan kartu ini merupakan permainan yang cukup diminati di Jepang dan di dunia, hingga memiliki berbagai macam variasi dari pengembangan permainannya. Salah satu variasi pengembangan permainannya adalah *Toraburu Hanafuda Travel Journal*. Pada *Toraburu Hanafuda Travel Journal*, permainan hanafuda dikombinasikan dengan menambahkan karakter – karakter dari serial populer dari Jepang, “Fate/” dan kemampuan – kemampuan karakternya ke dalam permainan hanafuda. Makalah ini akan menjelaskan pemanfaatan algoritma greedy dalam pemilihan kartu yang digunakan beserta kemampuan – kemampuan karakter untuk meraih kemenangan dalam *Toraburu Hanafuda Travel Journal*

**Kata Kunci**—Hanafuda; Fate; Kartu; Fate/Hollow Ataraxia; Mini game

## I. PENDAHULUAN

Jepang merupakan sebuah Negara yang memiliki budaya – budaya yang sangat unik. Salah satu budaya unik yang dimiliki oleh Negara ini adalah permainan tradisionalnya. Permainan – permainan tradisional yang berasal dari Jepang memiliki semantik permainan yang sangat indah, serta tingkat kompleksitas permainan yang sangat tinggi, bahkan dibandingkan dengan permainan dari daerah lain yang memiliki bentuk permainan yang sejenis, sehingga menarik perhatian banyak ahli di seluruh dunia untuk mendalami strategi – strategi yang dapat membantu para pemain untuk memenangkan permainan tersebut.

Salah satu permainan tradisional yang berasal dari Jepang adalah hanafuda. Hanafuda, secara bahasa memiliki arti “Kartu Bunga”. Seperti namanya, hanafuda merupakan permainan mengumpulkan kartu yang memiliki gambar bunga – bunga dari berbagai jenis dan musim yang ada di Jepang. Permainan ini dimainkan oleh dua orang pemain yang saling berlawanan. Tujuan dari permainan ini adalah untuk mengumpulkan poin sebanyak – banyaknya dan mengalahkan lawan dengan mengumpulkan poin lebih banyak dari lawan dalam ronde/waktu yang disediakan dalam permainan, atau terlebih

dahulu mencapai batas poin yang ditentukan di awal permainan.

Permainan ini cukup dinikmati tidak hanya di Jepang, namun juga di beberapa Negara dan daerah seperti Korea dan Hawaii. Istilah dan Peraturan – peraturan yang digunakan di setiap tempat juga bervariasi. Kartu hanafuda sendiri juga memiliki berbagai macam variasi permainan yang dapat dimainkan. Setiap permainan memiliki mekanisme yang berbeda pula.

Pada setiap giliran permainan, pemain dapat mengambil kartu yang tersedia dengan mencocokkannya dengan kartu – kartu yang dipegang di tangan. Setiap kartu memiliki nilai masing – masing, dan juga memiliki sebuah nilai ketika dikombinasikan dengan kartu – kartu lainnya yang dimiliki. Ketika seorang pemain sudah memiliki suatu kombinasi kartu yang memiliki sebuah nilai tertentu, pemain dapat memutuskan untuk mengakhiri ronde dan menyimpan nilai yang sudah dikumpulkan, atau mengambil resiko untuk mengumpulkan kombinasi lain untuk menambah nilai yang bisa disimpan pada ronde tersebut.

Permainan ini kemudian diadaptasi oleh sebuah franchise yang sangat terkenal dari Jepang yaitu serial Fate/. Pada salah satu game dari franchise ini yang berjudul Fate/Hollow Ataraxia, terdapat sebuah mini game yang mengadaptasi permainan hanafuda jenis *Koi-Koi* dan menyatukannya dengan elemen – elemen yang terdapat dalam franchise Fate/ seperti cerita, karakter, dan kemampuan – kemampuannya sehingga melahirkan sebuah permainan baru yang berjudul *Toraburu Hanafuda Travel Journal*.



Gambar 1.1. Game *Toraburu Hanafuda Travel Journal*.  
Sumber : *in-game screenshot* penulis

Algoritma *greedy* dipilih dalam usaha untuk membangun kecerdasan buatan yang dapat memenangkan permainan *Toraburu Hanafuda Travel Journal* ini karena permainan ini dimainkan dengan cara mengambil pilihan pada setiap gilirannya. Bobot dari masing – masing pilihan tersebut nantinya akan diperhitungkan melalui penentuan nilai bobot tindakan. Dari pertimbangan tersebut, akan dipilih sebuah tindakan dengan bobot paling baik. Penentuan nilai bobot setiap pilihan membutuhkan banyak pertimbangan. Pertimbangan – pertimbangan tersebut antara lain adalah kemampuan yang dimiliki karakter pemain, kartu yang dipegang di tangan pemain, kartu – kartu yang tersedia untuk diambil, kartu – kartu yang sudah diambil, poin – poin dari berbagai kombinasi kartu yang dapat diperoleh, dan lain – lain. Pada penentuan bobot pada makalah ini dibatasi oleh nilai kombinasi terdekat yang dapat dibentuk, kemampuan dari karakter pemain,serta kartu – kartu yang sudah diambil oleh kedua pemain.

## II. DASAR TEORI

### A. Definsi Algoritma Greedy

Algoritma *greedy* merupakan suatu algoritma yang dilakukan untuk membentuk solusi tiap langkah. Pada setiap langkahnya terdapat banyak pilihan yang harus dieksplorasi dan ditentukan mana pilihan terbaiknya. Keputusan yang telah diambil tersebut tidak akan dapat diubah lagi karena setelah itu langsung berpindah pada langkah selanjutnya. Contoh algoritma *greedy* misalkan untuk menempatkan komponen di atas papan sirukuit, jika suatu komponen sudah diletakkan di papan, komponen tersebut tidak dapat dipindahkan lagi.

Algoritma *greedy* menggunakan pendekatan dengan membuat pilihan-pilihan kandidat yang dapat memberikan nilai terbaik, yaitu dengan membuat pilihan optimum lokal pada setiap langkahnya. Dengan dipilihnya masing-masing optimum

lokal pada tiap tahap tersebut, diharapkan optimum global dapat dicapai.

Salah satu contoh permasalahan yang dapat diselesaikan dengan algoritma *greedy* adalah masalah penukaran uang. Misalkan terdapat uang dengan nilai  $X$  serta sekumpulan uang koin dengan berbagai nominal, ditanyakan jumlah minimum koin yang diperlukan untuk menukarkan uang senilai  $X$  tersebut. Dengan menggunakan algoritma *greedy*, pada tiap tahap dicari nilai koin maksimum dengan nilai yang lebih kecil dari  $X$ . Kemudian, nilai  $X$  dikurangi dengan nilai koin terpilih pada tahap tersebut, sehingga menjadi nilai maksimum koin yang akan dipilih pada tahap selanjutnya. Tahap tersebut diulangi sampai  $X$  bernilai 0 atau sudah ditukarkan semua.

Pada penukaran uang tersebut, algoritma *greedy* belum tentu menghasilkan nilai optimum global. Algoritma *greedy* tersebut hanya dapat menghasilkan nilai yang cukup baik, namun apabila dicari dengan menggunakan Makalah IF2211 Strategi Algoritma – Sem. II Tahun 2016/2017 algoritma brute force, akan ditemukan solusi yang lebih optimal. Untuk masalah penukaran uang Euro, dollar AS, dan crown Swedia, algoritma penukaran uang dengan *greedy* ini pasti menghasilkan nilai yang optimum.

Disimpulkan bahwa algoritma *greedy* dapat didefinisikan sebagai algoritma yang memecakan masalah langkah per langkah dengan pada tiap langkahnya dipilih suatu pilihan terbaik yang dapat diperoleh pada saat itu juga tanpa memperhatikan konsekuensi ke depan. Dengan memilih pilihan terbaik pada tiap langkah, diharapkan solusi optimum global dapat dicapai pada akhirnya.

### B. Elemen – Elemen Algoritma Greedy

#### 1. Himpunan kandidat (C)

Himpunan kandidat ini terdiri atas elemen-elemen yang mungkin membentuk solusi. Contoh dari himpunan kandidat ini misalnya himpunan koin, himpunan job yang akan dikerjakan, himpunan simpul di dalam graf, dan lain-lain bergantung pada persoalannya.

#### 2. Himpunan solusi (S)

Himpunan solusi ini terdiri atas kandidat-kandidat yang terpilih sebagai solusi persoalan. Himpunan solusi ini merupakan himpunan dari kandidat.

#### 3. Fungsi seleksi

Fungsi yang pada setiap langkah memilih satu elemen dari himpunan kandidat sebagai satu solusi yang memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi apakah kandidat tersebut benar-benar solusi optimalnya atau bukan. Fungsi seleksi ini biasanya memberikan suatu bobot pada masing-masing elemen pada himpunan kandidat lalu memilih suatu elemen dengan bobot paling besar atau paling kecil, bergantung pada persoalan.

#### 4. Fungsi kelayakan (feasible)

Fungsi kelayakan ini memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak. Solusi layak ini berarti bersamaan dengan himpunan solusi yang terbentuk tidak melanggar aturan atau constraint yang ada. Kandidat yang layak dimasukkan ke himpunan solusi dan tidak dipertimbangkan lagi.

### 5. Fungsi obyektif

Fungsi obyektif ini merupakan fungsi yang memaksimalkan atau meminimalkan nilai solusi.

## III. PERMAINAN TORABURU HANAFUDA TRAVEL JOURNAL

Permainan *Toraburu Hanafuda Travel Journal* mengadaptasi permainan kartu Hanafuda dengan melakukan ekspansi terhadap aturan Hanafuda Jepang atau yang biasa disebut dengan Koi-Koi. Kondisi untuk memenangkan permainan adalah untuk menghabiskan poin yang dimiliki musuh dengan cara mengurangkannya dengan poin yang dikumpulkan dalam tiap ronde. Ekspansi mekanisme permainan yang dilakukan adalah penambahan elemen karakter pemain beserta kemampuan dari karakter tersebut yang disebut *noble phantasm*.

### A. Kartu Hanafuda

Kartu hanafuda yang digunakan pada *Toraburu Hanafuda Travel Journal* adalah kartu – kartu yang digunakan pada Hanafuda Koi-Koi. Didalam tumpukan kartu pada permainan, terdapat 48 kartu yang berbeda, yang terdiri dari 12 Jenis kartu, dimana setiap jenis kartu terdiri dari 4 kartu yang berbeda. Setiap kartu melambangkan sebuah bunga atau tanaman, dan setiap jenis kartu melambangkan salah satu dari 12 bulan yang ada dalam satu tahun.

Kartu – kartu yang ada dibagi menjadi 4 tingkatan yaitu *Light*, *Animal*, *Ribbons*, dan *dreg*. Setiap jenis kartu memiliki 1 kartu *light* atau 1 kartu *animal*, 1 kartu *ribbons*, dan 2 kartu *dreg*, kecuali untuk kartu bulan November yang memiliki 1 kartu *light*, 1 kartu *animal*, 1 kartu *ribbons*, dan 1 kartu *dreg*, dan kartu bulan desember yang memiliki 1 kartu *light* dan 3 kartu *dreg*.



Gambar 3.1. Daftar kartu hanafuda dalam *Toraburu Hanafuda Travel Journal*. Sumber : *in-game screenshot* penulis

### B. Aturan Umum Permainan Hanafuda Koi-Koi

Urutan giliran permainan ditentukan dengan cara masing – masing pemain mengambil kartu secara acak dari tumpukan kartu, kemudian membandingkan kedua kartu yang diambil. Pemain yang mendapatkan giliran pertama adalah pemain yang memiliki kartu bulan lebih awal berdasarkan urutannya dalam satu tahun. Misalkan dibandingkan kartu april dan maret, maka pemain dengan kartu maret mendapatkan giliran pertama.

Pada awal permainan, setiap pemain memiliki 8 Kartu di tangan. Kemudian diletakkan 8 kartu diantara kedua pemain. Untuk 8 kartu di tengah kedua pemain. Untuk kartu – kartu di tengah, apabila terdapat 3 kartu yang sejenis sama, maka ketiga kartu tersebut ditumpuk dan jika kartu – kartu tersebut diambil, maka kartu – kartu yang ditumpuk juga akan diambil bersamaan.

Setiap giliran, pemain dapat mengambil kartu yang ada di tengah dengan mencocokkan jenisnya dengan kartu yang ada di tangan. Kemudian kedua kartu yang dicocokkan disimpan pemain yang mengambil. Selain mengambil kartu, pemain juga dapat meletakkan kartu dari tangannya ke tengah. Setelah mengambil atau meletakkan kartu, pemain tersebut membuka kartu teratas dari tumpukan kartu. Jika kartu yang diambil dapat dicocokkan dengan kartu yang ada di tengah, maka pemain tersebut harus mengambil salah satu kartu di tengah yang memiliki jenis yang sama dengan kartu yang ditarik. Jika tidak ada yang sejenis, maka kartu yang ditarik diletakkan di tengah, kemudian giliran pemain tersebut berakhir dan diganti oleh giliran pemain lawannya.

Jika setelah mengambil kartu dari tumpukan kartu, kartu – kartu yang dimiliki pemain tersebut membentuk sebuah kombinasi yang disebut *yaku*, maka pemain tersebut dapat mengakhiri ronde dan mengambil poin *yaku* yang dimiliki, atau mendeklarasikan *Koi-Koi*, dimana permainan akan dilanjutkan, hingga salah satu pemain mendapatkan *yaku* baru, atau mencapai kondisi ronde berakhir. Ketika seorang pemain mendeklarasikan *Koi-Koi*, pemain tersebut harus menambah nilai *yaku* yang dimilikinya untuk menyimpan poin yang pemain tersebut miliki. Ronde berakhir ketika salah satu pemain memiliki *yaku* dan memutuskan untuk menyimpannya, atau ketika kartu di tangan kedua pemain habis.

*Koi – koi* sendiri memiliki banyak variasi seperti *soft koi*, *hard koi*, dan lain – lain. Aturan untuk *koi – koi* dapat divariasikan untuk menghasilkan tantangan yang berbeda pula. Misalkan pada *soft koi*, ketika mendeklarasikan *koi*, kemudian berhasil menambah nilai *yaku* yang didapat, maka nilai *yaku* yang didapat akan dihitung menjadi dua kali lipatnya.

### C. Yaku

*Yaku* adalah kombinasi kartu – kartu yang memiliki nilai tertentu. Terdapat 12 Jenis *yaku* yang terdapat dalam permainan. Setiap *yaku* memiliki poinnya masing – masing, dan dapat bertambah dengan kondisi tertentu atau ketika digabungkan dengan *yaku* lain. Beberapa *yaku* dapat diatur untuk tidak dimasukkan kedalam permainan.



Gambar 3.2. Daftar *yaku*, kartu pembentuk *yaku*, dan deskripsi *yaku* dan poinnya dalam *Toraburu Hanafuda Travel Journal*. Sumber : *in-game screenshot* penulis

### D. Teknis permainan *Toraburu Hanafuda Travel Journal*

Dalam *Toraburu Hanafuda Travel Journal*, setiap pemain memiliki *Points*, *Mana Points (MP)*, dan karakter masing – masing. *Points* adalah jumlah poin yang dibutuhkan pemain lawan untuk mengalahkan pemain tersebut. Setiap pemain memiliki 12 *Points* pada awal permainan. *Mana Points* atau *MP* adalah poin yang dapat digunakan pemain untuk mengaktifkan *noble phantasm* karakter pemain. Karakter pemain menentukan jumlah *Mana Points* yang dimiliki pemain, beserta *noble phantasm* yang dimiliki pemain. *Noble phantasm* adalah kemampuan yang dimiliki oleh karakter pemain, yang dapat digunakan dalam permainan untuk membantu pemain.

Setiap karakter memiliki sejumlah *noble phantasm* yang berbeda – beda. *Noble phantasm* terbagi menjadi dua kategori yaitu pasif dan aktif. *Noble phantasm* pasif adalah *noble phantasm* yang selalu bekerja sejak awal permainan dimulai, hingga permainan berakhir. *Noble phantasm* aktif adalah *noble phantasm* yang hanya dapat digunakan pada saat – saat tertentu. Untuk mengaktifkan *noble phantasm* aktif, pemain harus membayar *mana points* sesuai dengan *noble phantasm* yang ingin diaktifkan, pada saat dimana *noble phantasm* tersebut bisa digunakan.

Untuk menambah jumlah *mana points* yang dimiliki pemain, terdapat dua cara. Cara pertama adalah mengambil kartu tingkat *light* dan menyimpannya. Cara kedua adalah dengan mengumpulkan kartu – kartu tertentu sesuai dengan karakter yang dimiliki.

## IV. PENGAPLIKASIAN ALGORITMA GREEDY

### A. Variabel Statis

Variabel statis ini disimpan untuk membantu penentuan nilai bobot dari masing-masing pilihan aksi. Variabelvariabel yang perlu disimpan antara lain:

#### 1. myNoblePhantasm

Variabel ini merupakan variabel yang menyimpan data dari *noble phantasm* yang dimiliki beserta jenis dan *MP* yang dibutuhkan untuk mengaktifkan *noble phantasm* tersebut. Untuk setiap *noble phantasm* diberi nilai -1 jika *noble phantasm* tersebut merupakan *noble phantasm* pasif, nilai 0 untuk *noble phantasm* yang bersifat dapat dipakai pada giliran sendiri dalam giliran, nilai 1 untuk *noble phantasm* yang digunakan untuk menambah poin yang diberikan kepada lawan, dan nilai 2 untuk *noble phantasm* yang digunakan ketika giliran lawan atau ketika menerima poin dari lawan. Juga terdapat nilai untuk menyatakan apakah *noble phantasm* tersebut bisa digunakan atau tidak pada saat itu. *Noble phantasm* yang aktif secara otomatis ketika terjadi suatu kejadian dianggap *noble phantasm* pasif.

#### 2. isInYaku

Variabel ini menyimpan Boolean apakah lawan sedang memiliki *yaku*, namun sedang dalam masa *koi-koi* atau tidak. Jika bernilai true, maka lawan sedang dalam *koi-koi* dan akan mengubah strategi untuk menyelesaikan *yaku* yang sudah mulai terbentuk

#### 3. yakuProgress

Variabel ini menyimpan progress untuk membentuk setiap *yaku* yang ada. Setiap *yaku* akan memiliki nilai progressnya masing – masing. Untuk *yaku* yang tidak mungkin terbentuk lagi akan bernilai -1, *yaku* yang masih bisa terbentuk tapi belum terbentuk akan bernilai 0, dan *yaku* yang sudah terbentuk akan memiliki jumlah kartu yang sudah disimpan untuk mendapatkan *yaku* tersebut.

#### 4. listYaku

Variabel ini menyimpan nilai yang dibutuhkan untuk setiap *yaku* terbentuk. Misalkan untuk *yaku* Ribbons memiliki nilai 5.

#### 5. enemyYaku

Variabel ini melakukan hal yang sama dengan *progressYaku*, tetapi digunakan untuk memeriksa progress dari *yaku* musuh.

#### 6. canKoi

Variabel ini akan menyimpan Boolean yang digunakan untuk memeriksa apakah aman dilakukan *koi* pada saat ini. Jika bernilai true, maka ketika pemain akan mendeklarasikan *koi* ketika bisa. Nilai

dari Boolean diperoleh dengan meninjau progressYaku dan enemyYaku.

### B. Himpunan Kandidat

Himpunan Kandidat dari permainan *Toraburu Hanafuda Travel Journal* ini adalah kartu yang tersedia di tangan pemain sebagai aksi yang mungkin dilakukan. Himpunan kandidat ini nantinya akan bercabang lagi ke pemilihan kartu yang diambil dari tengah. Masing – masing cabang dari himpunan kandidat ini akan dihitung bobotnya dengan suatu rumus tertentu.

Cabang elemen kandidat pada tiap giliran berupa kartu yang tersedia di tengah, yang memiliki jenis yang sama dengan satu atau lebih kartu yang disimpan di tangan, dan *yaku* yang sedang dibentuk. Kartu yang diambil adalah kartu yang paling dekat untuk membuat sebuah *yaku* yang masih dapat dibuat oleh pemain. Setiap kartu memiliki cabang sesuai dengan *yaku* yang dapat dibentuk oleh kartu tersebut dan kartu yang tersedia di tengah.

### C. Implementasi Greedy

Penentuan bobot dilakukan setiap kali akan mengambil kartu yang ada di tengah. Penentuan bobot dari setiap kartu yang akan diambil adalah sebagai berikut.

Setiap kartu memiliki nilai masing – masing berdasarkan tingkatan kartu tersebut. Kartu *light* bernilai 20, *Animal* bernilai 10, *ribbon* bernilai 5, dan *dreg* bernilai 1. Untuk setiap nilai progress *yaku* yang mengandung kartu tersebut, nilai kartu tersebut bertambah 1. Misalkan progress untuk membuat *yaku inoshikacho* adalah sudah mengumpulkan 1 kartu babi dan satu kartu kupu – kupu. Maka progress dari *yaku* tersebut adalah 67% dan memberikan nilai 6 poin. Sehingga nilai dari kartu rusa, yang akan menyelesaikan *yaku* tersebut jika diambil akan menjadi  $10 + 6 = 16$  Poin.

Ketika suatu *yaku* menjadi tidak dapat diperoleh lagi oleh pemain, maka nilai dari *yaku* tersebut tidak akan mempengaruhi nilai dari langkah – langkah yang bisa diambil. Sehingga membuat pemain tidak akan mencoba untuk mengarah ke *yaku* tersebut lagi.

Untuk melakukan *koi-koi*, maka akan dilakukan perhitungan keuntungan untuk melakukan *koi-koi*. Untuk memeriksa apakah pemain sedang dalam posisi aman untuk melakukan *koi-koi* digunakan prosedur dengan *pseudocode* I sebagai berikut :

```

Procedure canKoi()
  Int margin = NumCardsInHand*2
  If (Yaku.yakuProgressSum()>enemyYaku.sum()+margin)
  then
    Koikoi()
  Else
    //do not koi
  endif

```

dengan numCardsInHand adalah jumlah kartu yang tersisa di tangan lawan.

Untuk menggunakan *noble phantasm*, dilakukan dengan memperhatikan jenis dari *noble phantasm* dan jumlah *MP* yang sedang dimiliki oleh player. *Noble phantasm* akan selalu digunakan ketika *noble phantasm* tersebut dapat digunakan. *Pseudocode* prosedur untuk pemakaian *noble phantasm* adalah sebagai berikut:

```

Procedure noblePhantasm()
  If (NoblePhantasm.getMP()>=Player.getMP() and
  NoblePhantasm.isUsable) then
    If (NoblePhantasm.Type=0) then
      Use()
    Else if (NoblePhantasm.Type=1)then
      If(Enemy.Point<NoblePhantasm.calcDamage()) then
        Use()
      Else if (NoblePhantasm.Type=2) then
        Use()
      endif
    else
      //do nothing
    endif

```

*noble phantasm* bertipe 2 yaitu *noble phantasm* yang aktif saat giliran lawan atau saat menerima poin dari lawan tidak akan bisa digunakan ketika tidak sedang dalam giliran lawan (isUsable bernilai false) sehingga tidak akan digunakan. *Noble Phantasm* beripe 0 dan 1 yaitu *noble phantasm* yang dapat digunakan pada giliran sendiri dan *noble phantasm* yang akan memberikan bonus poin yang diberikan ke lawan tidak akan bisa digunakan ketika tidak sedang dalam giliran sendiri.

Ketika lawan sedang berada status *koi – koi*, maka pemain akan merubah prioritas, yaitu dari ke mengumpulkan *yaku* dengan poin terbesar, menjadi ke menyelesaikan *yaku* secepatnya untuk menghentikan lawan dari menambah poin yang diberikan. Prosedur perubahan sifat dilakukan dengan *pseudocode* sebagai berikut :

```

Yaku searchClosestYaku()
Yaku closest
For (int i=0; i<11;i++) do
  If(Yaku[i].yakuProgress()!=-1 and closest.yakuProgress <
  Yaku[i].yakuProgress) then
    Closest=Yaku[i]
  End if
Return closest
Procedure enemyKoi()
Yaku closestYaku=searchClosestYaku()
For(int i=0;i<11;i++) do
  If(Yaku[i]!=closestYaku)then

```

Yaku[i].yakuProgress=-1

Endif

Prosedur diatas membuat pemain tidak akan mengejar *yaku* lain selain *yaku* yang paling dekat, dengan harapan bisa menghentikan lawan sebelum ia menyelesaikan *yakunya* yang baru

## V. KESIMPULAN

Algoritma greedy cocok dilakukan untuk pemilihan keputusan optimal yang terbagi atas tahap-tahap dan tidak dibutuhkan pertimbangan atas pemilihan keputusan selanjutnya ataupun sebelumnya. Permainan *Toraburu Hanafuda Travel Journal* adalah salah satu permainan yang dapat diperkirakan solusi optimalnya apabila tidak memperhatikan faktor – faktor yang tidak bisa dikendalikan seperti keberuntungan kartu.

## UCAPAN TERIMA KASIH

Penulis mengucapkan rasa syukur kepada Allah SWT. yang atas rahmat dan kuasanya makalah ini bisa selesai tepat pada waktunya. Penulis mengucapkan terima kasih kepada Bapak Rinaldi Munir, Ibu Ulfa Maulidevi dan Ibu Masayu sebagai dosen pengampu mata kuliah IF2211 Strategi Algoritma atas bimbingannya. Penulis juga berterima kasih

kepada orangtua, keluarga dan teman-teman yang selalu mendukung penulis melalui doa.

## REFERENSI

- [1] Rinaldi Munir, Diktat Kuliah IF2211: Strategi Algoritma. Bandung: Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, 2009. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.
- [2] Hanafuda-Rama!. <http://www.sloperama.com/hanafuda/> diakses pada tanggal 13 Mei 2018 pukul 21.19

## 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, 14 Mei 2018



Iqrar Aminullah 13516126