

Institut Teknologi Bandung, Departemen Teknik Informatika

Penyempurnaan Intelegensa Buatan Mode AI ++ Ragnapolis dengan Langkah Penjualan Kota Berbasiskan Algoritma Greedy

Strategi Algoritma IF3051 2010/2011

Penyempurnaan Intelegensa Buatan Mode AI ++ Ragnapolis dengan Langkah Penjualan Kota Berbasiskan Algoritma Greedy

Robert Gunawan - 13508038
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
archangel_gabriel46@yahoo.com ; if18038@students.if.itb.ac.id

Abstrak

Algoritma Greedy merupakan algoritma optimasi yang paling mendasar dan memiliki banyak terapan dalam pembuatan program-program khususnya program permainan berbasiskan desktop. Algoritma greedy memungkinkan terjadinya pembuatan daftar hal-hal yang harus dilakukan oleh intelegensa buatan berdasarkan rumus-rumus yang sudah diterapkan sebelumnya. Salah satu pemanfaatan algoritma Greedy dalam game adalah pembuatan sebuah Intelegensa Buatan dalam permainan monopoli. Dalam permainan monopoli Ragnapolis, greedy digunakan untuk menentukan kota-kota yang diincar oleh intelegensa buatan yang mengendalikan pemain yang ada.

Kata Kunci: Intelegensa Buatan, AI, Artificial Intelligence, Greedy, Algoritma, Optimasi, Ragnapolis

I. PENDAHULUAN

Ragnapolis adalah salah satu game monopoli buatan Indonesia yang dibuat oleh 3 orang mahasiswa ITB : Robert Gunawan, William Eka Putra, dan Muqtafi Akhmad. Pembuatan game ini menggunakan kakas Borland Delphi yang mengusung bahasa pemrograman Pascal dan mendukung *User Interface*. Dalam game Ragnapolis, terdapat fitur Intelegensa Buatan (AI) yang akan menggerakkan player CPU dengan 3 buah algoritma greedy yang ada, antara lain : Algoritma Greedy *Offensive* yang akan bermain dengan cepat dan brutal, Algoritma Greedy *Defensive* yang akan bermain dengan aman, dan Algoritma Greedy Optimized yang akan bermain dengan memperhatikan keseluruhan aspek permainan dan kota yang ada.

Ragnapolis menyediakan 2 macam permainan dengan masing-masing permainan terdiri dari 2 board permainan yang berukuran 24 kota dan 28 kota. 2 macam permainan yang ada di Ragnapolis adalah mode AI normal dan mode AI++. Perbedaan antara keduanya adalah mode AI normal hanya memungkinkan AI untuk membeli kota, namun

tidak dapat menjual kota yang sudah dimiliki, sedangkan mode AI ++, AI dapat menjual kota yang ada hanya jika kekurangan uang atau sudah terancam akan bangkrut (sebutan di game monopoli untuk pemain yang kalah).

Sistem AI yang ada, khususnya mode AI++ dinilai kurang pintar karena dia melakukan penjualan hanya dengan melakukan pencarian kota dari awal sampai akhir, dan mengambil kota pertama yang terpilih.

II. LANDASAN TEORI

Algoritma Greedy

Algoritma *Greedy* adalah salah satu metode pemecahan persoalan optimasi yang paling populer. Secara harafiah, greedy memiliki arti tamak atau rakus. Orang yang tamak akan mengambil sebanyak mungkin apa yang tersedia tanpa memikirkan konsekuensi ke depan. Algoritma greedy pun demikian, algoritma ini bersifat sederhana dan lempang dengan prinsip, "take what you can get now". Pada tiap langkah algoritma greedy dipilih pilihan optimum lokal, dengan harapan bahwa langkah sisanya mengarah ke solusi yang optimum global.

Untuk menentukan solusi algoritma greedy memiliki kendala (*constraint*) dan fungsi optimasi. Solusi yang memenuhi semua kendala disebut solusi layak, dan solusi layak yang mengoptimumkan fungsi optimasi disebut solusi optimum.

Persoalan optimasi dalam konterks algoritma greedy tersusun oleh elemen-elemen berikut

1. Himpunan kandidat, C
Merupakan himpunan yang berisi elemen-elemen pembentuk solusi. Pada setiap langkah, satu buah kandidat diambil dari himpunannya. Contohnya adalah himpunan simpul dalam graf untuk menentukan graf merentang minimum.
2. Himpunan solusi, S
Berisi kandidat-kandidat yang terpilih sebagai solusi persoalan. Dengan kata lain, himpunan solusi adalah himpunan bagian dari himpunan kandidat.
3. Fungsi seleksi

Fungsi yang pada setiap langkah memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya. Biasanya setiap kandidat, x , di-assign sebuah nilai numerik, dan fungsi seleksi memilih x yang mempunyai bilangan nilai terbesar atau memilih x yang memiliki nilai terkecil.

4. Fungsi kelayakan

Merupakan fungsi yang memeriksa apakah suatu kandidat yang dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala yang ada. Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi.

5. Fungsi objektif

Yaitu fungsi yang memaksimumkan atau meminimumkan nilai solusi.

Monopoli

Monopoli adalah salah satu permainan papan yang paling terkenal di dunia. Tujuan permainan ini adalah untuk menguasai semua petak di atas papan melalui pembelian, penyewaan dan pertukaran properti dalam sistem ekonomi yang disederhanakan.

Setiap pemain melemparkan dadu secara bergiliran untuk memindahkan bidaknya, dan apabila ia mendarat di petak yang belum dimiliki oleh pemain lain, ia dapat membeli petak itu sesuai harga yang tertera. Bila petak itu sudah dibeli pemain lain, ia harus membayar pemain itu uang sewa yang jumlahnya juga sudah ditetapkan.

Permainan Monopoli diciptakan pada tahun 1904 oleh Elizabeth (Lizzie) J. Magie Phillips dengan nama awal *The Landlord's Game*. Satu set permainan Monopoli terdiri dari berapa buah bidak yang mewakili pemain, dua buah dadu, setumpuk kartu kesempatan, setumpuk kartu dana umum, papan permainan, uang permainan, miniatur properti (hotel dan rumah) dan akte kepemilikan setiap daerah yang tersedia. Sesuai dengan namanya, permainan Monopoli mengharuskan kita untuk membuat bangkrut pemain lain dengan cara menguasai setiap aspek bisnis yang ditawarkan oleh permainan. Para pemain Monopoli awam biasanya melakukan satu kesalahan dalam bermain Monopoli. Kesalahan tersebut biasanya berkutat di seputar kebebasan membeli tanah yang ditempati saat melangkah. Pada aturan aslinya setiap pemain yang melangkah ke kawasan tertentu diwajibkan untuk membeli tanah tersebut atau melelangnya saat tanah tersebut tidak ingin dibeli atau pemain yang berada di atas tanah tersebut tidak memiliki uang.

III. TIPE BENTUKAN

Berikut ini adalah tipe bentuk yang ada di dalam Ragnapolis

1. Type City

```

type City
<
    KotaID : integer;           {ID
kota}
    Penjara : boolean;         {
penanda apakah penjara atau bukan }
    NamaKota : string;         {nama
kota}
    HBeli : integer;           {harga
beli kota}
    HJual : integer;           {harga
jual kota}
    Sewa : integer;            {sewa
per kunjungan}
    Peluang : real;            {peluang
kota dikunjungi}
    IsAvailable : boolean;     {apakah
kota masih tersedia untuk dibeli}
    PID : integer;             {ID dari
player yang memiliki kota}
>

```

2. Type SortKota

```

type SortKota
<
    KotaID : integer; {ID kota}
    Bobot : real; {bobot untuk kota
tersebut}
>

```

3. Type Field

```

type Field
<
    Kota : array [1..28] of city;
{menyatakan kota-kota yang tersedia}
    JumlahKota : integer;
{menyatakan jumlah maks kota yang
dipakai}
    Value : array [1..3] of array
[1..28] of SortKota; { untuk algoritma
greedy}
    JumlahKotaAvailable : integer;
{menyatakan jumlah kota yang masih
available}
>

```

4. Type Player

```

type Player
<
    PlayerID : integer; {ID pemain}
    Nama : string; {nama pemain}
    PMode : integer; {mode pemain,
untuk menunjukkan apakah pemain adalah
user atau komputer}
    Uang : integer; {jumlah uang
yang dimiliki oleh pemain}
    IsWin : boolean; {penanda apakah
pemain mendapatkan kemenangan}
>

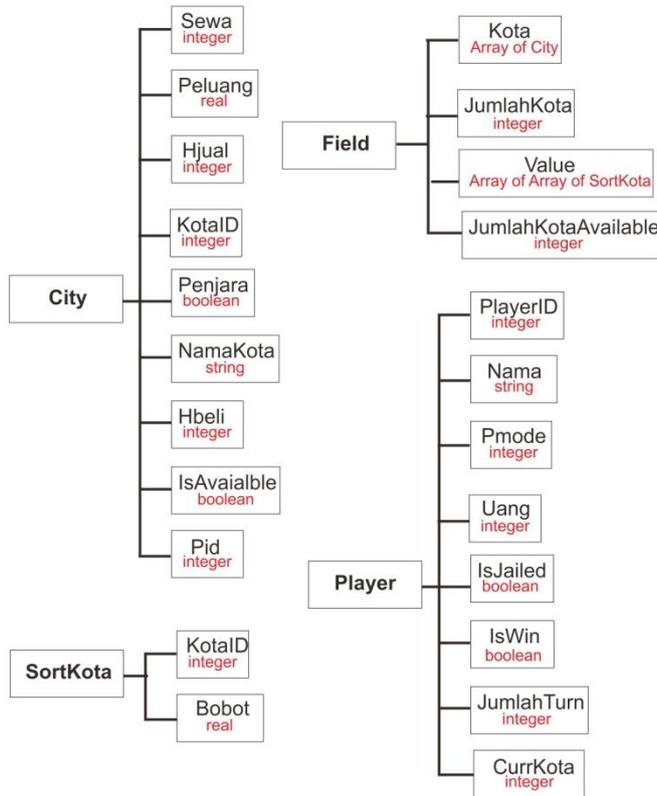
```

```

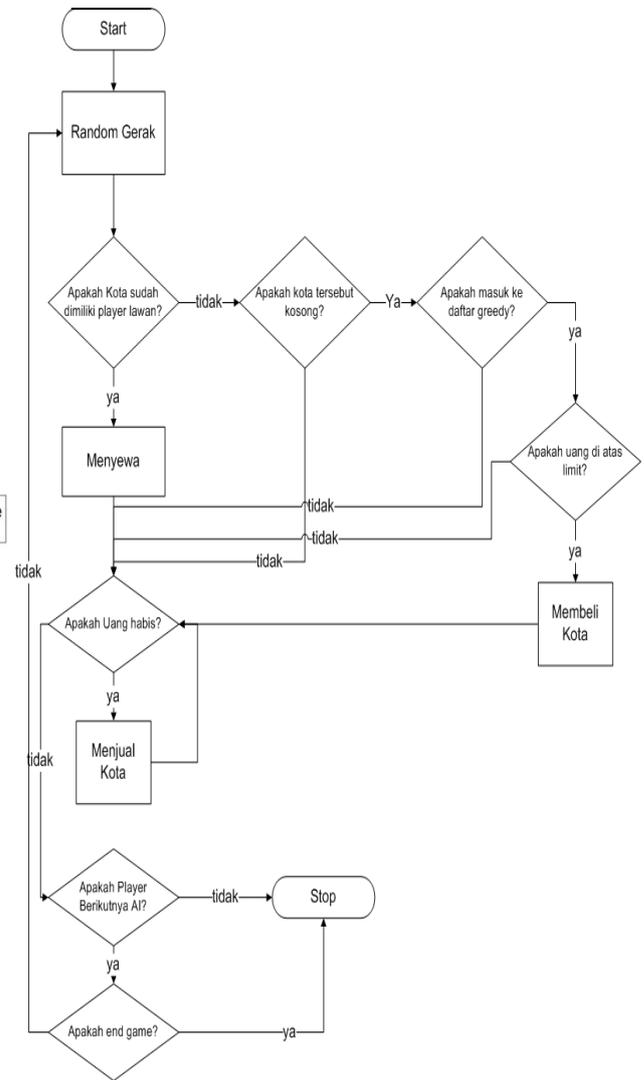
IsJailed : boolean; {penanda
apakah pemain masuk ke penjara}
JumlahTurn : integer; {jumlah
giliran pemain}
CurrKota : integer; {kota tempat
pemain sekarang berada}
>

```

Struktur data secara keseluruhan dapat digambar sebagai berikut



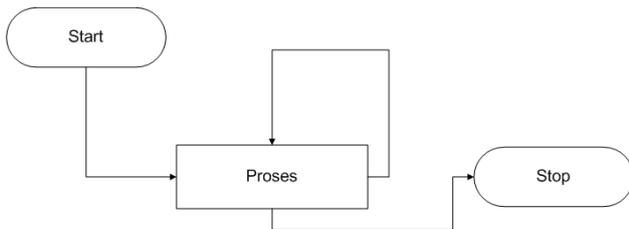
Gambar 1 Tipe Bentuk Ragnapolis



Gambar 3 AI Mode ++

IV. INTELEGENSA BUATAN YANG DIGUNAKAN DALAM RAGNAPOLIS MODE AI++

Dalam pembangunan Ragnapolis, penulis mengimplementasikan AI yang sederhana dalam programnya. AI tersebut dapat dilihat dalam gambar di bawah ini:



Gambar 2 AI Sederhana

Diagram AI lengkap (tanpa proses penjualan) dapat dilihat di bawah ini:

Berikut ini adalah algoritma AI yang digunakan

```

Procedure Berjalan ()
Kamus Lokal
Algoritma
{Kalau Player saat ini adalah CPU}
{Kalau tiba di kota yang sudah ada pemiliknya dan bukan milik player tsb}
{menyewa}
{selain itu}
{jika uang diatas limit}
{membeli}
{jika uang dibawah limit}
{tampilkan pesan kurang uang}
{cek uang, jika dibawah limit, lakukan penjualan}

```

```

{cek player berikutnya, apakah
CPU atau bukan}
{jika player berikutnya CPU}
  Berjalan()
{jika bukan CPU (manusia)}
  {Do nothing}

```

Potongan kode yang berisi proses penjualan dapat dilihat di algoritma di bawah ini

```

//...
if (TabPlayer[CurrPlayer].Uang <=0)
then //uang currplayer habis
begin
  if (TabPlayer[CurrPlayer].PMode
<> 1) then //ini mode AI ++ nya
begin
  tempjual := 0;
  for iterator := 1 to
TabField.JumlahKota do
begin
  if
(TabField.Kota[iterator].PID =
CurrPlayer) then //saat ditemukan
sebuah kota
tempjual := iterator;
end;
if (tempjual <> 0) then //jika
ditemukan ada kota yang bisa dijual
Menjual(tempjual);
end;
if (TabPlayer[CurrPlayer].Uang
<=0) then //uang currplayer masih
habis (akomodasi mode AI ++ sekaligus
player human)
begin
  TabPlayer[CurrPlayer].IsWin
:= false;
  TabPlayer[(CurrPlayer mod
2)+ 1].IsWin := true; //player lawan
menang
  EndGame := true;
end;
end;
//...

```

V. IDE PERBAIKAN

Dari Algoritma yang sudah diberikan di bab IV, terlihat bahwa proses penjualan di mode AI++ cukup sederhana. Komputer hanya perlu melakukan perulangan dari kota pertama hingga kota terakhir untuk mencari kota pertama yang ditemui dan akan menjual kota tersebut (Lihat lingkaran merah di potongan kode di atas). Disamping itu, proses penjualan hanya dilakukan sekali, jadi seandainya setelah menjual pun, uang player lawan masih kurang dari 0, maka pemain AI akan langsung kalah.

Ide untuk memperbaiki proses penjualan adalah sebagai berikut:

1. Cek Apakah uang player dibawah 0
2. Jika uang player dibawah 0, lakukan perulangan

sampai uang player tersebut sudah diatas 0 atau kota player habis

- a. Cari kota yang dimiliki oleh player berdasarkan greedy yang dipakai dan memiliki nilai paling kecil (Algoritma Greedy untuk menahan kota-kota dengan value tinggi), jual
- b. Cek apakah uang player sudah diatas 0
 - i. Jika belum dan kota habis, maka player kalah
 - ii. Jika belum dan kota masih ada, tidak melakukan apapun (hanya menunggu perulangan saja)

3. Akhir dari perulangan.

Dalam ide ini, penulis menyempurnakan 2 hal

1. Proses penjualan yang buruk, langsung menjual kota yang pertama ditemukan. Hal ini dapat digantikan dengan sistem penjualan dengan mencari kota dengan nilai terkecil untuk dijual agar kota-kota dengan nilai tinggi tetap dipegang di tangan.
2. Penanganan untuk kasus uang masih dibawah 0 saat sudah menjual. Dengan menangani hal ini, penulis dapat membuat AI lebih pintar dan dapat melanjutkan permainan lebih lama daripada AI sebelumnya. Mengingat bahwa AI adalah pengganti manusia dalam game, maka diperlukan AI yang lebih pintar..

VI. IMPLEMENTASI

Berikut ini adalah implementasi yang dilakukan

```

//...
if (TabPlayer[CurrPlayer].Uang <=0)
then //uang currplayer habis
begin
  //hitung kota yang tersedia dulu
  TempKota := 0; //inisiasi awal
  for j:= 1 to TabField.JumlahKota
do
begin
  if (TabField.Kota[i].PID =
CurrPlayer) then
begin
  TempKota++; //iterasi
end;
end;
//mulai proses penjualan
tempjual = 0; //inisiasi indeks
penjualan
MinValue = 9999; //inisiasi
indeks min kota untuk greedy minimum
repeat //perulangan sampai kota
habis, atau uang player sudah di atas
0
begin
  for j :=
(TabField.JumlahKotaAvailable+1) to

```

```

TabField.JumlahKota do //coba lakukan
pencarian dari tabel greedy yang ada
begin

if (TabField.Value[TabPlayer[CurrPlayer
].PMode - 1][j].Bobot < MinValue) then
begin
MinValue :=
TabField.Value[TabPlayer[CurrPlayer].P
Mode - 1][j].Bobot;
tempjual := j;
end;
end;
Menjual(tempjual);
end;
until ((TempKota = 0) or
(TabPlayer[CurrPlayer].Uang >0));
if (TabPlayer[CurrPlayer].Uang <=0)
then //uang currplayer masih habis
(akomodasi mode AI ++ sekaligus player
human)
begin
TabPlayer[CurrPlayer].IsWin
:= false;
TabPlayer[(CurrPlayer mod
2)+ 1].IsWin := true; //player lawan
menang
EndGame := true;
end;

end;

//...

```

VII. ANALISA

Dari percobaan bermain dengan komputer, terbukti komputer menjadi lebih pintar. Komputer akan menjual kota-kota yang dianggap tidak terlalu berharga bagi dirinya sesuai dengan mode greedy yang sudah ditentukan sebelumnya jika dia sudah kehabisan uang. AI akan lebih bertahan lama untuk kebanyakan kondisi permainan yang ada mengingat pelemparan dadu adalah sesuatu yang random. Sangat kecil kemungkinan 2 peristiwa terjadi berulang-ulang.

VIII. KESIMPULAN DAN SARAN

Kesimpulan

Algoritma Greedy mampu menyelesaikan kebanyakan persoalan pemilihan yang ada.

Saran

Pengembangan Ragnapolis masih cukup sederhana dengan algoritma yang cukup sederhana. Untuk pengembangan algoritma AI yang lebih baik masih terbuka lebar.

IX. UCAPAN TERIMA KASIH

Penulis mengucapkan terimakasih kepada Tuhan YME yang sudah melindungi penulis dalam pembuatan makalah ini. Penulis juga berterimakasih kepada orang tua dan

keluarga yang selalu mendukung. Tak lupa, kepada seluruh civitas akademik Informatika ITB, khususnya kepada Bapak Ir. Rinaldi Munir M.T. yang sudah banyak mendukung, William Eka Putra dan Muqtafi Akhmad yang sudah membantu pengembangan Ragnapolis, beserta kawan-kawan semuanya.

Akhir kata, terimakasih

DAFTAR REFERENSI

- [1] www.informatika/~rinaldi
- [2] <http://www.kotakgame.com/feature/detail.php?page=2&page2=2&id=75>
- [3] Laporan Tugas Besar 1 Strategi Algoritma tahun 2010 – StrAlgo1-08038

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, 30 November 2010

Ttd



Robert Gunawan - 13508038