

ANALISA PENGGUNAAN ALGORITMA GREEDY PADA PERMAINAN “FIVE-LINK”

Joelian Samuel

Jurusan Informatika, Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jalan Ganesa 10 Bandung

ifi15092@students.if.itb.ac.id

ABSTRAK

Komputer telah menjadi alat bantu manusia yang sangat berguna dalam beberapa dekade ke belakang. Penggunaan komputer pun telah meluas tidak hanya sebagai alat komputasi atau perhitungan.

Salah satu kegunaan dari komputer pada masa sekarang adalah untuk membuat aplikasi permainan. Game dalam komputer telah menjadi sebuah kebutuhan di saat-saat sekarang ini, dimana seseorang yang sangat sibuk dan tidak memiliki waktu luang yang banyak dapat mendapatkan sedikit penyegaran dari bermain game di komputer. Permainan yang terdapat di komputer pun sudah sangat banyak macamnya dari yang sederhana sampai yang memakai algoritma yang sangat susah. Penulis pada makalah ini berusaha menerapkan algoritma greedy pada permainan five-link atau lebih dikenal tic-tac-toe.

Game five-link sendiri merupakan game logika yang telah dikenal banyak orang dan merupakan permainan tradisional yang diangkat menjadi permainan komputer. Game ini merupakan game yang memakai papan sebagai arena permainan layaknya catur. Game ini dilakukan oleh dua orang pemain dengan tujuan untuk membuat bidak pemain menjadi satu garis lurus secara horisontal, vertikal, ataupun diagonal dan menghalangi pemain lain dapat melakukan hal itu.

Aplikasi algoritma greedy di sini, dipakai untuk menentukan jalan komputer menaruh bidaknya pada papan permainan.

Kata kunci: Algoritma greedy, Five-Link, Game, Permainan.

1. PENDAHULUAN

Komputer telah menjadi alat bantu manusia yang paling sering dipakai dalam beberapa dekade ke belakang. Penggunaan komputer pun telah meluas ke berbagai

bidang. Semakin banyak pula masalah yang dapat diselesaikan menggunakan komputer.

Salah satu penggunaan komputer yang paling sering diaplikasikan adalah pembuatan game atau permainan. Permainan pada komputer telah menjadi salah satu perangkat lunak yang wajib ada pada setiap komputer yang dimiliki seseorang. Permainan atau game tersebut pun sangat banyak jenisnya. Salah satu aplikasi permainan ini adalah aplikasi permainan tradisional. Permainan tradisional adalah permainan yang sudah ada di masyarakat seperti permainan kartu, catur, ataupun cabang-cabang olahraga. Permainan tradisional ini yang semula dimainkan secara ‘real’ di masyarakat, sekarang dapat dilakukan dengan melawan komputer dengan simulasi.

Dalam simulasi inilah, strategi algoritmik dipakai untuk membuat semacam Intelejensi Buatan bagi komputer untuk menentukan langkah-langkah dalam suatu permainan. Strategi yang telah dikenal sudah banyak jenisnya seperti algoritma greedy, brute force, dan sebagainya.

Permainan Five-Link sendiri sudah dikenal masyarakat secara luas, namun mungkin saja dengan nama yang berbeda, seperti catur-jawa, ataupun tic tac toe. Permainan ini dilakukan dalam papan berukuran seperti papan catur dan dimainkan oleh dua orang. Tujuan dari permainan ini adalah untuk menempatkan bidak masing-masing pemain dalam suatu garis lurus yang dapat berupa garis horisontal, vertikal, ataupun diagonal.

Dalam makalah ini akan dijelaskan bagaimana penggunaan algoritma greedy dalam membantu komputer untuk menentukan langkah dalam menaruh bidak.



Gambar 1. Permainan Five-link (dalam versi berbeda) sedang dimainkan

2. DASAR TEORI

2.1 Definisi Umum Algoritma Greedy

Algoritma greedy adalah algoritma yang memecahkan masalah langkah per langkah yang dalam setiap langkahnya mengambil pilihan terbaik yang terdapat pada saat itu tanpa memedulikan konsekuensi ke depannya. Hal ini disebut sebagai optimum lokal. Harapan dari algoritma ini dapat membentuk suatu penyelesaian atau optimum global suatu masalah dari optimum lokalnya.

2.2 Skema Umum Algoritma Greedy

Persoalan optimasi suatu masalah menggunakan algoritma greedy dapat dipilah-pilah menjadi bagian-bagian sebagai berikut :

1. Himpunan Kandidat

Merupakan Himpunan yang berisi elemen-elemen penyelesaian persoalan

2. Himpunan Solusi

Merupakan Himpunan yang telah berisi solusi-solusi yang telah diterima sebagai langkah penyelesaian atau optimum lokal.

3. Fungsi Seleksi

Merupakan fungsi pembatas yang mensortir himpunan kandidat menjadi calon himpunan solusi

4. Fungsi Kelayakan

Merupakan fungsi pembatas bagi himpunan-himpunan kandidat bilamana sudah tidak layak lagi berada di dalam himpunan kandidat

5. Fungsi Objektif

Merupakan fungsi yang bertujuan memaksimalkan atau meminimalkan nilai solusi tergantung konteks permasalahannya.

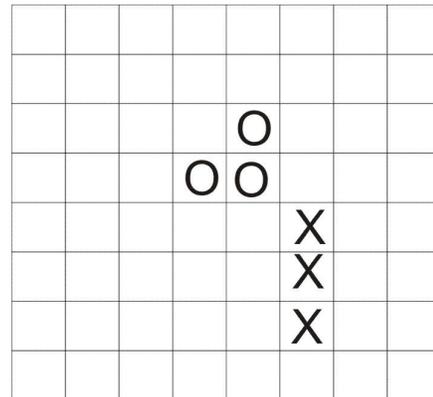
Algoritma greedy banyak dipakai dalam permasalahan seperti knapsack, penjadwalan job, dan sebagainya.

3. METODE PENYELESAIAN

3.1 Analisis Masalah

Dalam pencarian solusi untuk menentukan langkah komputer pada permainan five-link, kita harus meninjau terlebih dahulu beberapa permasalahan yang mungkin terjadi di dalam permainan.

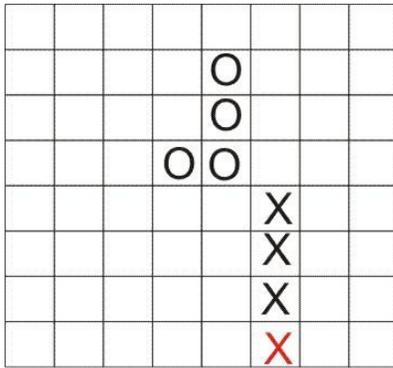
Permasalahan pertama yang dapat terjadi adalah ketika bidak pemain dan bidak komputer tidak ada yang mendekati jumlah lima bidak dalam satu rangkaian. Komputer dapat dengan mudah menentukkan peletakkan bidak pada papan permainan, seperti terlihat pada gambar. Pada kondisi ini kita dapat dengan langsung menerapkan algoritma greedy dengan memilih rangkaian bidak komputer yang memiliki jumlah paling mendekati lima.



X : Bidak Komputer
O : Bidak Pemain

Gambar 2. Analisis Masalah Pertama

Permasalahan kedua yang dapat terjadi dan paling sering terjadi adalah bidak lawan sudah ada yang memiliki rangkaian yang jumlahnya mendekati angka lima. Jika terdapat rangkaian tersebut, komputer harus memprioritaskan jalannya untuk menghalangi bidak lawan agar tidak dapat membentuk suatu rangkain lima bidak. Pada kasus ini, algoritma greedy tidak secara murni dipakai, karena ada kondisi yang menyebabkan semua kemungkinan dalam himpunan kandidat tidak dapat diterapkan. Dapat dilihat pada gambar di bawah contoh kasus seperti ini.



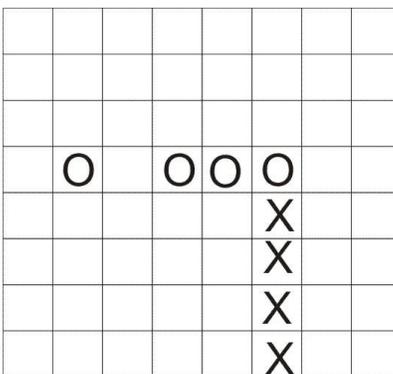
X : Bidak Komputer
O : Bidak Pemain

Gambar 3. Analisis Masalah Kedua

Pada gambar terlihat jumlah tiga bidak dalam satu rangkaian bidak lawan sudah dianggap mendekati lima rangkaian karena bila lawan menaruh bidak satu lagi yang membuat rangkaian itu menjadi empat –rangkaian, maka lawan memiliki dua kesempatan dari dua arah yang berbeda untuk membuat rangkaian itu menjadi lima-rangkaian.

Akan tetapi, pada masalah ini algoritma greedy dapat diterapkan apabila komputer telah lebih dahulu memiliki empat-rangkaian dan memiliki kesempatan memenangkan permainan dalam langkah berikutnya atau dengan kata lain lawan tidak melakukan pencegahan terhadap bidak komputer. Dapat dilihat pada gambar jika komputer memiliki bidak pada posisi X merah maka komputer tidak harus menghalangi jalan lawan lagi.

Kasus ketiga yang mungkin dapat terjadi adalah bidak lawan tidak dalam satu ‘rangkaian’ akan tetapi dapat membentuk lima-rangkaian. Dapat dilihat pada gambar



X : Bidak Komputer
O : Bidak Pemain

Gambar 4. Analisis Kasus Ketiga

Komputer harus dapat menghalangi jalan lawan karena terdapat kemungkinan lawan dapat memenangi permainan walaupun tidak dalam satu rangkaian.

3.2 Algoritma Penyelesaian Masalah

Dari analisis masalah di atas, penulis mencoba menerapkan algoritma greedy pada masalah ini. Pada masalah ini yang menjadi himpunan kandidat adalah semua posisi yang belum terisi oleh bidak lawan dan komputer pada papan permainan, dan yang menjadi himpunan solusi adalah semua posisi penaruhan bidak oleh komputer untuk memenangkan permainan ataupun untuk menghalangi lawan memenangkan permainan. Fungsi seleksi pada kasus ini adalah fungsi yang menyeleksi semua kemungkinan posisi kosong pada papan permainan yang menyebabkan rangkaian bidak yang dimiliki komputer bertambah panjang atau mendekati lima-rangkaian. Sedangkan untuk fungsi kelayakan pada algoritma penyelesaian masalah ini adalah penempatan posisi bidak komputer sehingga tidak menyebabkan lawan dapat memenangkan permainan ini. Dan fungsi objektif dari permainan ini adalah untuk membuat rangkaian bidak komputer menjadi lima-rangkaian.

Sehingga di dapat algoritma sebagai berikut

```

Procedure Cek_Lawan (input P : Papan,
output found : boolean, idxr : int,
idxc : int, jenis : int, max : int )
{Prosedur ini melakukan pengecekan pada
papan permainan mengenai posisi bidak
lawan, jika terdapat rangkaian tiga
bidak atau lebih maka akan
mengembalikan nilai true dan idxr, idxc
terisi posisi bidak pertama dalam
rangkaian bidak lawan dan jenis akan
mengembalikan apakah rangkaian terdapat
dalam satu kolom, baris, diagonal
utama, atau diagonal lainnya. Prosedur
ini akan berhenti pada penemuan pertama
}

```

```

Procedure Cek_Kawan (input P : Papan,
output idxr : int, idxc : int, jenis :
int, max : int)
{prosedur ini mencari rangkaian bidak
komputer terpanjang pada papan
permainan, dan mengembalikan posisi
bidak pertama dalam rangkaian, dan
jenis rangkaian itu melintang apakah
sebaris, sekolom, atau sediagonal
}

```

```

Deklarasi
    i, j, max1, max2: integer;
Algoritma
For (i = 0 to 7)
  For (j = 0 to 7)
    Cek_baris (p[i][j], max1)
    Cek_kolom (p[i][j], max2)
    If (max1 > max2)
      If (Max < max1 )
        Max = max1
        Idxr = i
        Idxc = j
        Jenis = 1
      endif
    Else
      If (Max < max1 )
        Max = max2
        Idxr = i
        Idxc = j
        Jenis = 2
      Endif
    Endif
    Cek_diagonalUtama (p[i][j], max1)
    Cek_diagonalLain (p[i][j], max2)
    If (max1 > max2)
      If (Max < max1 )
        Max = max1
        Idxr = i
        Idxc = j
        Jenis = 3
      Endif
    Else
      If (Max < max1 )
        Max = max2
        Idxr = i
        Idxc = j
        Jenis = 4
      Endif
    Endif
  EndFor
EndFor

Procedure Get_Next (input p:papan,
jenis :int, banyak : int, idxraw : int,
idxcaw : int ; output idxr : int, idxc
: int)
{Mendapatkan langkah selanjutnya
komputer}

Procedure Greedy ()
{Proses Melangkah Komputer}
Deklarasi
    P : papan
    I, j, idxr, idxc, max1, max2, j1, j2 : int
    Found : Boolean

```

```

Algoritma
Cek_Lawan (P, Found, i, j, j1, max1)
Cek_kawan (P, idxr, idxc, j2, max2)
  If (Found)
    If (max2 == 4)
      Get_Next (P, j2, max, idxr, idxc, i, j)
      P[i][j] = bidak_komp;
    Else
      Get_Next (P, j1, max, i, j, idxr, idxc)
      P[idxr][idxc] = bidak_komp
    Endif
  Else
    Get_Next (P, j2, max, idxr, idxc, i, j)
    P[i][j] = bidak_komp;
  Endif

```

Pada *pseudo-code* di atas, program memiliki tiga prosedur pendukung dan satu prosedur utama yang mengimplementasikan ketiga prosedur pendukung. Prosedur pendukung pertama, yaitu `cek_lawan` digunakan sebagai fungsi kelayakan yang membuat program akan memprioritaskan pencegahan terhadap bidak lawan yang telah membuat tiga atau lebih rangkaian. Prosedur yang kedua adalah `cek_kawan`. Prosedur ini merupakan kebalikan dari prosedur `cek_lawan`. Prosedur ini mencari posisi rangkaian yang membentuk rangkaian bidak komputer yang paling panjang. Prosedur inilah yang menjadi fungsi seleksi sekaligus prosedur yang menerapkan algoritma *greedy*. Prosedur ini memiliki empat prosedur bantuan yang tidak penulis jelaskan pada makalah ini yaitu prosedur `cek_baris`, `cek_kolom`, `cek_diagonalUtama`, dan `cek_diagonalLain`. Pada intinya, keempat prosedur bantuan ini memiliki fungsi yang sama, yaitu membalikkan panjang rangkaian yang terbentuk dari suatu bidak komputer di dalam papan permainan sesuai arah di dalam nam prosedur, prosedur ini juga mengecek apakah rangkaian tersebut memiliki kotak kosong untuk menaruh bidak komputer selanjutnya, jika tidak ada kotak kosong maka akan mengembalikan nilai 0 pada panjang rangkaian. Dari nilai balikkan fungsi tersebut didapatkanlah rangkaian terpanjang dari bidak yang dimiliki komputer. Prosedur yang ketiga sebenarnya hanyalah pencari posisi penaruhan berikutnya dari komputer.

Pada prosedur utama, semua prosedur pendukung dipakai dalam menentukan langkah komputer. Hal pertama dilakukan adalah melakukan pengecekan bidak lawan dan pengecekan bidak sendiri. Setelah itu dilihat apakah ada posisi bidak lawan yang berbahaya, jika ada lihat apakah komputer memiliki kesempatan memenangi permainan atau tidak. Jika komputer tidak memiliki kesempatan memenangi permainan maka komputer akan menghalangi rangkaian bidak lawan. Jika tidak ada rangkaian bidak lawan yang berbahaya, maka komputer dapat menambahkan bidak komputer pada posisi di

rangkaian yang terpanjang pada papan permainan dan mungkin saja dapat memenangi permainan.

IV. KESIMPULAN

Algoritma greedy yang digunakan pada permainan ini sebenarnya sudah cukup efektif dalam menahan bidak lawan agar tidak membuat rangkaian lima-bidak. Akan tetapi, karena hal itu kadang komputer tidak dapat menambah jumlah rangkaian bidaknya karena harus mencegah lawan mendapatkan rangkaian bidak yang lebih panjang. Bidak komputer pada permainan ini akan terlihat acak-acakkan karena harus menahan bidak lawan terus menerus.

Penulis belum mencoba algoritma lain sebagai pembanding dari algoritma greedy pada permainan ini. Akan tetapi, untuk permainan sederhana seperti five-link ini penulis merasa algoritma greedy sudah cukup baik dalam menjalankan simulasi permainan ini.

REFERENSI

- [1] Munir, Rinaldi, “Diktat Kuliah IF2251 Strategi Algoritmik”, Program Studi Informatika Sekolah Teknik Elektro dan Informatika ITB , 2007.
- [2] Five-link
<http://www.linkfive.com/LinkFiveIntro.html>
Diakses tanggal 15 Mei 2007 pukul 21.30 WIB.