

Pemanfaatan Algoritma BFS dalam Menyelesaikan Permasalahan *Knight Moves*

Faza Thirafi - 13514033
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13514033@std.stei.itb.ac.id

Abstrak—Permasalahan *Knight Moves* merupakan salah satu dari berbagai permasalahan pada di luar permainan catur yang cukup menarik. Bagaimana sebuah bidak kuda dapat berpindah dari suatu petak tertentu kepada petak yang lain.

Pada makalah ini, akan dicari solusi untuk permasalahan *Knight's Moves* dengan menggunakan program. Yang akan dicari adalah jumlah langkah yang dapat ditempuh oleh sebuah bidak kuda. Di sini, akan digunakan salah satu pendekatan pencarian solusi permasalahan graf atau pohon, yaitu BFS (*Breadth First Search*).

Disini akan terlihat bagaimana algoritma BFS dapat menyelesaikan suatu permasalahan. Juga akan ditampilkan hasil program yang menunjukkan bahwa BFS tetap bisa mencapai solusi, walaupun kompleksitasnya sangat tinggi. Namun karena permasalahan ini tidak terlalu besar, maka BFS cukup ampuh untuk mendapatkan solusinya.

Kata kunci—algoritma, *Knight's Moves*, BFS, bidak, papan catur

I. PENDAHULUAN

Catur adalah permainan klasik yang sudah ada sejak abad ke-6 di India (menurut sumber www.amazine.co). Dari sana, permainan catur tersebar ke seluruh belahan dunia hingga zaman modern. Maka catur adalah salah satu permainan yang banyak digemari di kalangan masyarakat.

Pada permainan catur biasa, terdapat dua pemain yang bermain pada papan catur yang sama. Satu pemain menggerakkan bidak berwarna putih sedangkan lawannya menggerakkan bidak berwarna hitam. Satu pemain memiliki “armada” berupa 8 bidak pion (*pawn*), 2 bidak benteng (*rook*), 2 bidak kuda (*knight*), 2 bidak menteri (*bishop*), 1 bidak ratu (*queen*), dan 1 bidak raja (*king*). Untuk memenangkan pertandingan, salah satu pemain harus melumpuhkan bidak raja lawannya. Karena permainan ini membutuhkan strategi yang khusus dan memacu kerja otak, maka permainan catur tergolong

sebagai sebuah olahraga.



Gambar 1 Bidak Kuda
(Diktat Matematika Diskrit)

Selain permainan biasa, ada beberapa “tantangan” yang biasa dibuat memanfaatkan bidak-bidak catur. Seperti 8 Queen problem, Knight’s Tour problem, dan Knight Moves. Pada makalah ini, akan dibahas tentang penyelesaian persoalan “*knigh’s moves*”.

Knight moves adalah permasalahan dalam catur dengan detail sebagai berikut. Jadi, pada papan catur berukuran 8 x 8, kita pilih dua buah petak tertentu. Maka dari 2 petak tersebut, bagaimana cara untuk sebuah kuda yang diletakkan pada salah satu petak untuk dapat berpindah dari salah satu petak ke petak yang lain, dengan gerakan “L” nya bidak kuda.

Permasalahan Knight Moves tersebut dapat diselesaikan dengan berbagai cara. Sebenarnya, permasalahan ini mirip dengan perkara “the shortest path” yang sering kali dibahas dalam pemrograman. Pada kesempatan kali ini, saya memakai algoritma BFS (*breadth first search*). BFS seperti yang kita ketahui merupakan pencarian melebar untuk mencari solusi sebuah perjalanan. Kita menganggap petak papan catur sebagai simpul dalam BFS dan sisinya

adalah penghubung antar petak. Simpul yang dibangun ialah petak yang dilewati oleh bidak kuda. Maka dengan BFS kita

Maka dalam makalah ini, kita akan mengetahui bagaimanakah algoritma BFS menyelesaikan permasalahan knight's move. Dengan kompleksitas yang dimiliki algoritma BFS, kita akan mengetahui seberapa mangkus jika kita memakai BFS dalam permasalahan ini.

II. DASAR TEORI

1. Catur

a. Sejarah

Seperti yang disampaikan sebelumnya, permainan ini mulai dikenalkan sejak abad ke-6, di India. Permainan ini disebut caturāṅga saat dahulu dimainkan pada masa-masa kerajaan Gupta. Kemudian, catur menyebar ke Persia dan disebut sebagai shatranj di Sassanid. Lalu masuk ke Spanyol, Yunani, hingga keseluruhan Eropa.

Aturan pada permainan catur sudah ada sejak dahulu, namun aturan yang berlaku sekarang didefinisikan pada abad ke-19 di Eropa.

Beberapa pemain catur ternama di era modern diantaranya adalah Paul Morphy, Wilhelm Steinitz, Johannes Zukertort, Emanuel Lasker, José Raúl Capablanca, Gary Kasparov, Anatoly Karpov, dan Viswanathan Anand. Untuk *Grand Master* dari Indonesia, Utut Adianto adalah salah satu yang ternama.

b. Permainan

Catur merupakan salah satu permainan yang mengasah otak dengan strategi, taktik, kecerdasan, dan keterampilan untuk mengalahkan pihak lawan. Untuk memenangkan pertandingan, pemain harus membuat raja lawan tidak bergerak lagi selagi sedang "skak" atau "check". Dalam permainannya, ada dua pihak pemain. Ada yang memegang bidak putih, lawannya memegang bidak hitam. Untuk pemain dengan bidak putih, memiliki hak untuk memulai pertandingan.

Seperti yang sudah disampaikan pula, setiap pemain mendapatkan 8 bidak pion, 2 bidak benteng, 2 bidak kuda, 2 bidak menteri, 1 bidak ratu dan 1 bidak raja. Setiap bidak memiliki keistimewaan sebagai berikut:

- Pion
Berjalan dengan melangkah satu per satu ke arah depan. Dapat memakan bidak lawan saat salah satu bidak berada pada diagonal depan pion tersebut. Jika mencapai wilayah lawan paling dalam, dapat "promosi" menjadi bidak benteng, kuda, menteri, atau ratu.
- Benteng
Berjalan lurus ke kanan, kiri, depan, atau belakang dengan jarak yang dibatasi papan catur, atau bidak sendiri. Dapat memakan bidak lawan

jika berada pada depan, belakang, kiri, atau kanan.

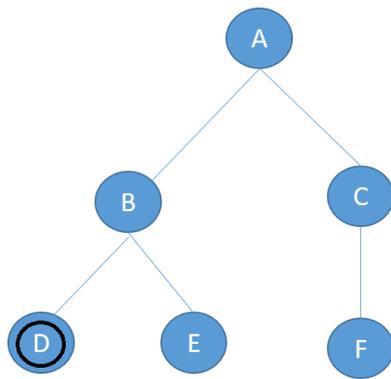
- Kuda
Berjalan dengan pola "L", yaitu 2 petak horizontal dengan 1 petak vertikal atau sebaliknya. Bidak ini dapat "melompati" bidak-bidak lainnya dengan bebas. Untuk memakan bidak lawan, jika bidaknya berada pada titik-titik pergerakan kuda.
- Menteri
Berjalan mendiagonal ke kanan depan, kiri depan, kanan belakang, atau kiri belakang dengan jarak yang dibatasi papan catur, atau bidak sendiri. Dapat memakan bidak lawan jika berada pada *range* ruang gerak.
- Ratu
Merupakan kombinasi gerakan antara benteng dan menteri. Biasanya, bidak ini merupakan andalan dari setiap pemain catur yang terus dipertahankan.
- Raja
Untuk bergerak, raja hanya dapat bergerak satu langkah dengan arah manapun (petak-petak disekitarnya). Raja adalah kunci dari permainan catur yang menentukan kemenangan seorang pemain. Dikatakan "skak" atau "check" bila bidak raja pemain berada pada zona dimana ada bidak lawan yang bisa memakan raja tersebut.

2. Algoritma BFS (Breadth First Search)

Algoritma *Breadth First Search* merupakan algoritma pencarian solusi pada graf dengan konsep 'melebar'. Algoritma ini akan memeriksa semua kemungkinan dahulu pada suatu status ke arah selanjutnya. Misal dari suatu level 0, BFS akan melihat semua anaknya pada level 1 dahulu. Setelah semua anak pada level 1 ditemukan, maka pencarian akan dilanjutkan pada semua anak pada simpul pertama level 1. Sebelum berpindah kepada keturunannya lagi, BFS harus menyelesaikan dahulu semua anak node pada level 2. Begitu seterusnya hingga solusi didapat.

Untuk mengatasi masalah belum adanya graf (kita mengkhususkan graf berbentuk pohon), maka harus dibangkitkan dahulu anak-anak dari simpul akar (graf dinamis). Pembangkitannya ini mirip langkah-langkahnya jika dibandingkan dengan graf yang sudah terbentuk (graf statis). Nantinya, solusi akan didapat bersamaan dengan dibangkitkannya simpul solusi tersebut.

Lebih detilnya, untuk menyelesaikan sebuah masalah graf secara BFS, dibutuhkan struktur data *queue* (antrian) dalam urutan pengunjungan / pembangkitan simpul. Lalu, untuk semua simpul, dibuat sebuah *array of Boolean* yang jumlah elemennya sebanyak simpul. Larik ini digunakan untuk mengetahui apakah sebuah simpul sudah dikunjungi atau belum. Misalkan ada pohon sebagai berikut:



Gambar 2 Contoh pohon merentang (sumber: pribadi)

Urutan pengunjungan:

A _ _ _ -> kunjungi semua anak A dan masukkan di belakang, hapus A

B C _ _ _ -> kunjungi semua anak B dan masukkan di belakang, hapus B

C D E _ -> kunjungi semua anak C dan masukkan di belakang, hapus C

D E F -> solusi ditemukan: D

Kompleksitas secara umum:

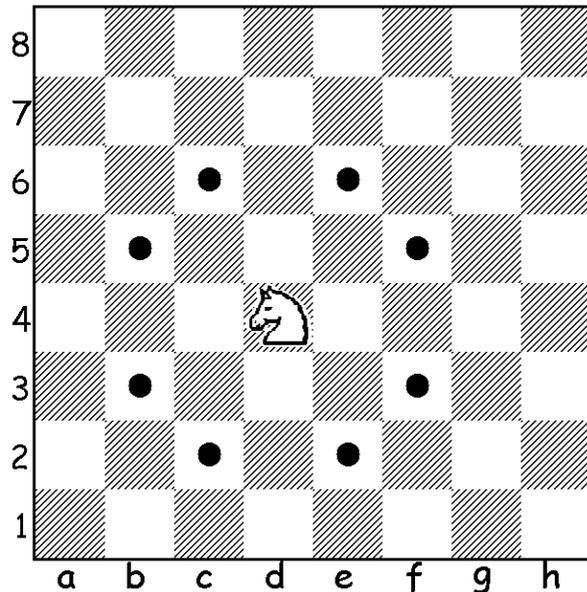
- Kompleksitas waktu: $1+b+b^2+b^3+\dots+b^d = O(b^d)$
- Kompleksitas ruang: $O(b^d)$

III. PEMBAHASAN

A. Pemaparan Masalah

Seperti yang sudah diungkapkan sebelumnya, bahwa makalah ini membahas tentang *Knight's Moves*. Permasalahan ini membahas bagaimanakah bidak kuda dapat berpindah dari satu petak tertentu ke petak yang lain dengan karakteristik pergerakan bidak kuda. Juga, sudah diungkapkan bahwa algoritma yang akan digunakan adalah algoritma BFS yang meng-*expand* simpul anaknya dengan cara melebar.

Dari petak-petak papan catur yang ada, kita namakan semuanya agar unik dan dapat diperiksa oleh program. Jadi untuk sisi papan "horizontal" dinamakan a, b, c, hingga h; dan untuk sisi papan "vertical" dinamakan 1, 2, 3, hingga 8. Contohnya, untuk kolom ketiga dan baris ketiga bernama petak c3, kolom ketujuh dan baris kedua bernama b7, dan seterusnya.



Gambar 3 Knight Move

(sumber:<http://www.spoj.com/problems/NAKANJ/>)

B. Penyelesaian

Dalam mengatur algoritma BFS nya, kita harus menentukan komponen-komponen yang menjadi penyusun BFS. Pertama, untuk simpul-simpul pohon merentang BFS menyatakan petak-petak. Satu petak akan memiliki anak sebanyak 2 hingga 4. Pendefinisian anak dari suatu simpul ialah, semua simpul yang mungkin dapat dicapai gerakan bidak kuda sekali langkah. Gerakan tersebut dapat dipandang sebagai dx dan dy yang merupakan pasangan dari:

```

dx[] = { -1, -1, 1, 1, -2, -2, 2, 2 };
dy[] = { 2, -2, 2, -2, 1, -1, 1, -1 };
  
```

Lalu, disini kita dapat menggunakan graf statis maupun dinamis untuk membentuk pohon ruang status petak-petak yang dapat diambil. Untuk graf statis, berarti semua kemungkinan petak yang ditempuh di mulai dari petak awal hingga semua petak telah dibangkitkan. Untuk graf dinamis, selagi mencari petak yang dapat dicapai bidak kuda, simpul-simpul dibangkitkan.

Disini, idenya adalah menyusuri petak-petak dari petak asal, sehingga mendapatkan simpul dengan nama sama dengan petak tujuan. Karena memakai BFS, maka pencarian petaknya dilakukan secara melebar dengan menyelesaikan semua anak-anak dari suatu simpul terlebih dahulu, baru berpindah ke keturunan salah satu anak dan seterusnya.

Maka, dengan algoritma BFS, kita dapat mencari solusi dengan *psedocode* di bawah ini:

```
function KnightMoveSol(input a : Petak, b :
Petak) → integer
{ mengembalikan jarak gerakan kuda yang
ditempuh dari PAwal ke PAKhir, menggunakan
algoritma BFS}
```

Deklarasi:

```
a, b : character
P : Array [1..8][1..8] of
integer
Q : PairQueue (karakter dan
angka)
C : Petak
X, Y : integer
dx : Array [1..8] of integer
dy : Array [1..8] of integer
Sx, Sy, Ex, Ey : integer
```

Algoritma:

```
dx
Sx ← a.karakter - 'a'
Sy ← a.angka - 1
Ey ← b.karakter - 'b'
Ey ← b.angka - 1
P[Sx][Sy] ← 0
Q.masukkan(Sx,Sy)
while (Q tidak kosong)
    C ← Q.elemendapan
    X ← C.karakter
    Y ← C.angka
    if (X = Ex and Y = Ey) then
        keluarWhile
    endif
    Q.keluarkan()
    for i=1 to 8 do
        if ((X+dx[i] >= 1) and
(X+dx[i]<=8)
        and (Y+dy[i] >= 1) and
(Y+dy[i]<=8))
            Q.masukkan((X+dx[i],Y+dy[i]))
            P[X+dx[i]][Y+dy[i]] ← P[X][Y]
        + 1
    endif
endfor
endwhile

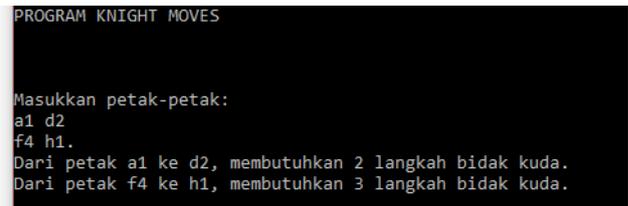
return P[Ex][Ey]
```

credits to:

UVa 439: Knight moves, Sai Cheemalapati

keterangan: "Petak" adalah sebuah tipe yang komponennya karakter (char) dan angka (int).

Setelah *pseudocode* tersebut diimplementasi ke program berbahasa pemrograman C++ dan dieksekusi pada *Command Prompt*, hasil yang ditampilkan adalah sebagai berikut:



Gambar 4 Hasil Eksekusi
(sumber:pribadi)

C. Alternatif Lain

Sebenarnya, ada juga metode lain dalam menyelesaikan permasalahan sejenis. Karena ini mengenai permasalahan graf, kita dapat mengubah algoritmanya ke A*, DFS, dan sebagainya. Untuk algoritma A*, maka harus dicari heuristiknya juga dengan mempertimbangkan sisi riil nya. Untuk DFS, maka pencarian petak akan dilakukan secara mendalam. Setiap langkah akan dicari keturunannya terlebih dahulu. Hal ini memungkinkan terjadinya *infinite search*, kecuali akan ada batasan pembangkitan simpul.

IV. KESIMPULAN

Knight's Moves adalah contoh dari berbagai persoalan di dunia nyata (lebih khusus lagi permainan catur) yang dapat ditemukan solusinya dengan pola pikir komputasi dan algoritmik. Maka pola pikir seperti ini sangat diperlukan dalam menyelesaikan berbagai permasalahan. Tinggal bagaimana caranya kita memastikan bahwa solusi tersebut tidak akan menimbulkan kerugian bagi kita. Ada beberapa kasus dimana program yang dibuat harus terhindar dari kecacatan dan harus memiliki akurasi yang tinggi karena terkadang permasalahan tersebut menimbulkan risiko kerugian bahkan kematian.

Kembali ke masalah *Knight's Moves*, BFS adalah salah satu alternatif algoritma yang dapat dipakai dalam menyelesaikan permasalahan graf atau pohon seperti ini. Pada permasalahan *Knight's Moves* ini, BFS akan mencari petak disekitarnya terlebih dahulu. Walaupun masih memiliki kompleksitas yang tinggi dalam teorinya, BFS tetap dapat diandalkan dalam penyelesaian masalah ini.

Namun tidak hanya menggunakan BFS, kita dapat memakai algoritma lain yang berhubungan dengan permasalahan graf. Di luar sana, sangat banyak algoritma yang ditemukan untuk menyelesaikannya seperti DFS, A*, UCS, dan sebagainya. Semua algoritma tersebut memiliki kompleksitas, kelebihan, dan kekurangan masing-masing. Maka kita dapat mengeksplor lebih jauh terkait dengan algoritma yang dipakai.

Harapannya, pembaca tidak hanya terpaku pada isi makalah ini, dapat juga mengeksplorasi algoritma-algoritma lain yang dapat dipakai untuk menyelesaikan persoalan sejenis ini. Semoga makalah ini dapat bermanfaat, khususnya bagi penulis dan umumnya untuk pembaca sekalian.

V. UCAPAN TERIMA KASIH

Dengan terbentuknya makalah ini, saya mengucapkan terima kasih, pertama dan utama kepada Allah SWT yang telah mencurahkan ilmu-Nya dan atas kehendak-Nya makalah ini dapat selesai. Lalu kepada orang tua dan keluarga saya yang selalu mendukung studi saya di Teknik Informatika ini. Lalu kepada dosen-dosen saya yang saya banggakan, Bapak Rinaldi Munir dan Ibu Nur Ulfa Maulidevi yang telah membimbing kuliah Strategi Algoritma pada semester ini dengan penuh dedikasi. Kemudian, kepada teman-teman seperjuangan, Teknik Informatika ITB 2014 yang mendukung dan menyemangati penulis dalam penyelesaian makalah ini.

Terakhir, penulis mengucapkan terima kasih kepada seluruh pembaca makalah ini, dari mana pun asalnya. Makin banyak makalah ini tersebar mudah-mudahan dapat menjadi pemacu bagi para pembaca untuk bisa menggali lagi dari isi makalah ini. Mohon maaf apabila terdapat kekurangan, Terima kasih.

REFERENSI

- [1] Munir, Rinaldi. "Diktat Kuliah IF2211 Stratego Algoritma", Informatika, Bandung: 2009
- [2] Anonymous. "Sejarah Permainan Catur, Mulai India Kuno hingga Era Modern" from: <http://www.amazine.co/39266/sejarah-permainan-catur-mulai-india-kuno-hingga-era-modern/>, diakses tanggal 8 Mei 2016, pukul 13.53
- [3] Kovshenin, Konstantin. "Bipartite Matching, an Application" from: <https://saicheems.wordpress.com/2013/11/13/uva-439-knight-moves/>, diakses tanggal 8 Mei 2016, pukul 14.10

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, 6 Mei 2015



Faza Thirafi / 13514033