PERBANDINGAN ALGORITMA RUNUT-BALIK DENGAN ALGORITMA TANPA RUNUT-BALIK (*FAST-SWAP*) DALAM PERSOALAN PERPINDAHAN GUNDU

Yudi Rizkiadi[1] , Eko Adi Wicaksono[2] , Yandri Rahmat Fadli[3]

Departemen Teknik Informatika
Fakultas Teknologi Industri, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung

E-mail : if13073@students.if.itb.ac.id[1] , if13091@students.if.itb.ac.id[2] , if13103@students.if.itb.ac.id[3]

Abstraksi

Dalam dunia pemrograman, istilah algoritma sudah tidak asing lagi. Dalam menyusun sebuah program, maka seorang *programmer* akan membuat algoritma terlebih dahulu sebelum membuat program. Algoritma yang digunakan oleh *programmer* tersebut harus sesuai dengan program yang akan dibuat. Hal ini dilakukan agar program yang dibuat menjadi lebih mangkus. Ada berbagai jenis algoritma pemrograman, diantaranya adalah algoritma *BruteForce*, algoritma *Greedy*, dan algoritma Runut-Balik (*backtracking*). Masing-masing algoritma tersebut memiliki perbedaan satu sama lain, diantaranya adalah perbedaan dalam strategi pemecahan masalah dan waktu yang dibutuhkan untuk memecahkan masalah. Karena perbedaan tersebut, maka untuk menyelesaikan suatu masalah secara mangkus dibutuhkan algoritma yang benar-benar sesuai dengan masalah yang akan diselesaikan. Makalah ini akan membahas mangkus atau tidaknya algoritma *backtracking* dalam menyelesaikan masalah perpindahan gundu.

Kata kunci : *algoritma, runut-balik, program*

I. Pendahuluan

Runut-balik(*Backtracking*) adalah algoritma yang berbasis pada DFS untuk mencari solusi persoalan secara lebih mangkus. Runut-balik, yang merupakan perbaikan dari algoritma brute-force, secara sistematis mencari solusi persoalan di antara semua kemungkinan solusi yang ada. Dengan metode ini, kita tidak perlu memeriksa semua kemungkinan solusi yang ada. Hanya pencarian yang mengarah ke solusi saja yang selalu dipertimbangkan. Akibatnya, waktu pencarian dapat dihemat. Runut-balik lebih alami dinyatakan dengan algoritma rekursif.Kadang-kadang disebutkan pula bahwa runut-balik merupakan bentuk tipikal dari algoritma rekursif.

Isitlah runut-balik pertama kali diperkenalkan oleh D.H. Lehmer pada tahun 1950. Selanjutnya, R.J. Walke, Golomb, dan Baumert menyajikan uraian umum tentang runut-balik dan penerapannya pada berbagai persoalan. Saat ini, algoritma runut-balik banyak diterapkan untuk program persoalan dan masalah- masalah pada bidang kecerdasan buatan.

Karena algoritma runut balik banyak diterapkan untuk program persoalan, maka kami mencoba menganalisis apakah algoritma runut-balik dapat diterapkan pada persoalan pepindahan gundu dan menghasilkan waktu pencarian yang hemat.

II. Tujuan

1. Menentukan apakah algoritma runut-balik dapat diterapkan pada persoalan perpindahan gundu.
2. Menganalisis rentang waktu yang dibutuhkan untuk pencarian solusi dalam persoalan perpindahan gundu dengan menggunakan algoritma runut-balik.
3. Menganalisis langkah – langkah pencarian solusi dalam persoalan perpindahan gundu dengan menggunakan algoritma runut-balik.
4. Menentukan algoritma terbaik dalam pencarian solusi pada persoalan perpindahan gundu dengan membandingkan algoritma runut-balik dan algoritma lain tanpa runut-balik.

III. Skema Umum Algoritma Runut-Balik dan *Fast-Swap*

Dengan menggunakan algoritma runut-balik, maka algoritma yang dihasilkan adalah sebagai berikut

```
function TMarbleSet.Swap() → boolean;

{ Fungsi yang digunakan untuk menukar
gundu dengan mengunakan algoritma runut
balik}

Deklarasi

  i: integer

Algoritma

  If MarbleSolved then
    Swap ← true;
```

```
function TmarbleSet.Swap() -> boolean
{ Fungsi yang digunakan untuk menukar
gundu dengan menggunakan algoritma runut
balik }

Deklarasi
 i : integer

Algoritma

if MarbleSolved then
  return true
else
  for i<-1 to 4 do
    move(i)
    if Swap then
      return true
    else
      unmove(i)
    endif
  endfor
  return false
endif
```

Sedangkan tanpa algoritma runut-balik, algoritma yang dihasilkan adalah sebagai berikut :

```
function TmarbleSet.FastSwap() -> boolean
{ Fungsi yang digunakan untuk menukar
gundu tanpa menggunakan algoritma runut
balik }

Deklarasi
 i : integer

Algoritma

while not solved do {
  while move(2) do endwhile
  move(1)
  while move(4) do endwhile
  move(2)
endwhile
```

Dari kedua algoritma tersebut, manakah yang lebih mangkus? Apakah algoritma runut-balik memberikan hasil yang lebih mangkus dibandingkan dengan algoritma Fast-swap? Kita akan mencoba menganalisisnya.

IV. Ruang Lingkup

Makalah ini berisi berbagai pemikiran, usulan dan konsep dalam menganalisis pencarian solusi pada persoalan perpindahan gundu.
Algoritma yang diaplikasikan adalah algoritma runut-balik (*backtrack*) dan tanpa runut-balik (*fast-swap*)
Penilaian terhadap penggunaan suatu algoritma dalam pencarian solusi dianalisis dari rentang waktu dan jumlah langkah pencarian solusi yang didapat dari data eksperimen.

V. Deskripsi Persoalan

Diberikan suatu persoalan sebagai berikut :

Terdapat *n* buah gundu yang berwarna hitam dan *n* buah gundu yang berwarna putih, dan sebuah papan persoalan yang berbentuk lajur dan terdiri dari 2*n* + 1 ruang untuk meletakkan gundu di dalamnya.

Semua gundu hitam diletakkan pada satu sisi (bagian kiri), sedangkan semua gundu putih diletakkan pada bagian yang lain (bagian kanan), dan satu buah ruang kosong di antara kedua sisi tersebut.

Tujuan persoalan ini adalah untuk membuat semua gundu yang berwarna hitam berada di sebelah kanan dan semua gundu yang berwarna putih berada di sebelah kiri. Ketentuan perpindahan adalah sebagai berikut :

1. Gundu hitam hanya dapat digerakkan ke kanan, sedangkan gundu putih hanya dapat digerakkan ke kiri (tidak dapat berbalik arah).
2. Pada setiap perpindahan, setiap gundu dapat :
   - maju satu posisi jika ruang didepannya kosong, atau
   - Melompati tepat satu buah gundu yang berbeda warna jika ruang di depan gundu yang dilompati tersebut kosong.

Persoalan ini selalu mempunyai solusi yang dapat diterapkan. Permasalahannya adalah apakah dengan menggunakan algoritma runut-balik waktu yang dibutuhkan dalam pencarian solusi menjadi lebih cepat dan langkah yang dibutuhkan menjadi lebih sedikit, ataukah ada algoritma lain yang lebih mangkus dalam pencarian solusi.

VI. Hasil Analisis

Dalam melakukan analisis terhadap permasalahan persoalan perpindahan gundu, gundu yang digunakan adalah gundu dengan jumlah 1,2,4,dan 6 buah gundu.

Spesifikasi komputer yang digunakan dalam pengujian ini adalah sebagai berikut :

System : Microsoft Windows XP Pro SP2
Computer :
AMD Sempron(tm) XP2500+
1746.8 MHz
512 MB DDR

Pengujian dilakukan dengan mematikan fitur animasi. Setelah dilakukan pengujian, maka didapat hasil sebagai berikut :

Dengan menggunakan algoritma runut-balik

Jumlah Gundu	Banyak Langkah	Waktu (ms)	Runut-balik
1	3	0	0
2	14	15	6
4	90	78	66
8	1799	1984	1719

Tabel 6.1 Penyelesaian dengan algoritma runut balik

Tanpa menggunakan algoritma runut-balik

Jumlah Gundu	Banyak Langkah	Waktu (ms)	Runut-balik
1	3	0	0
2	8	0	0
4	24	16	0
8	80	47	0

Tabel 6.2 Penyelesaian tanpa algoritma runut balik

Selisih pergerakan (langkah + runut balik) yang dilakukan antara penerapan dengan algoritma runut-balik dan tanpa algoritma runut balik

Jumlah Gundu	Selisih Pergerakan
1	0
2	11
4	116
8	3391

Tabel 6.3

Jika dilihat dari selisih pergerakan, maka dapat disimpulkan bahwa perbedaan yang mendasar pada saat menggunakan algoritma runut-balik dan tanpa penggunaan algoritma runuk balik terdapat pada jumlah runut-balik yang dilakukan.

Dengan menggunakan algoritma runut-balik, maka pencarian yang mengarah ke solusi akan dicoba dan jika tidak mengarah ke pencarian solusi, maka akan dilakukan runut-balik ke simpul sebelumnya.

Sedangkan tanpa algoritma runut-balik, maka runut-balik tidak akan dilakukan sehingga akan menghasilkan langkah penyelesaian yang lebih singkat dibandingkan dengan penyelesaian dengan algoritma runut-balik. Secara otomatis, waktu yang diperlukan pun menjadi lebih sedikit. Hal ini dapat dilihat pada tabel 4.1 dan 4.2.

Dengan melihat tabel penyelesaian dengan metode tanpa runut-balik, dapat disimpulkan bahwa banyaknya langkah efektif untuk mencapai solusi akhir adalah:

$$f(n) = n^2 + 2n$$

n = Banyaknya gundu pada satu sisi

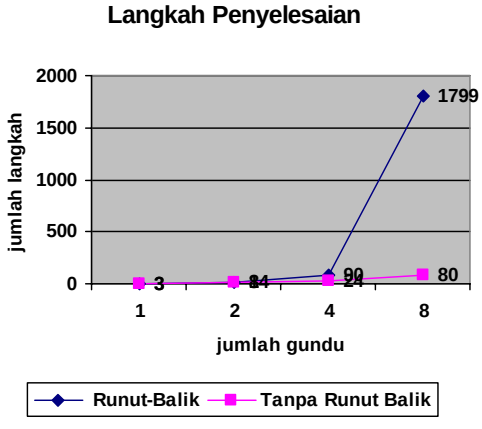
Selisih waktu yang diperlukan dalam pencapaian solusi dalam persoalan perpindahan gundu adalah sebagai berikut

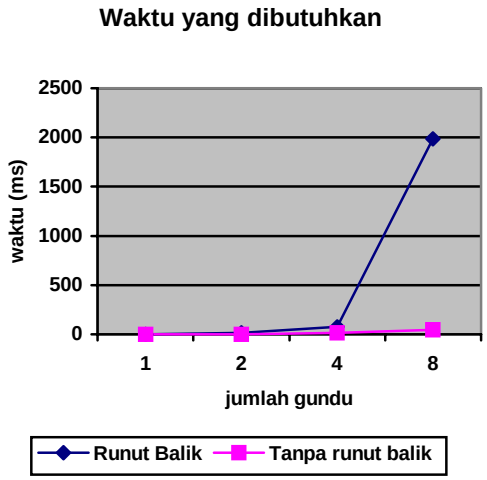
Jumlah Gundu	Selisih Waktu (ms)
1	0
2	15
4	62
8	1937

Tabel 6.4

Dilihat dari tabel 4.4, maka semakin banyak gundu yang diuji, maka selisih waktu akan semakin besar. Hal ini dapat membuktikan bahwa algoritma runut-balik jika diterapkan pada persoalan perpindahan gundu tidak dapat menghasilkan waktu yang mangkus.

Grafik berikut memperlihatkan perbedaan yang sangat besar dalam menggunakan algoritma runut-balik dan tanpa algoritma runut-balik dalam penyelesaian persoalan perpindahan gundu.





Dari kedua grafik di atas, terlihat jelas bahwa laju pertumbuhan dalam banyaknya langkah penyelesaian dan jumlah waktu yang dibutuhkan terlihat lebih cepat jika menggunakan algoritma runut-balik.

Jika menggunakan jumlah gundu yang sangat besar, penggunaan algoritma runut-balik dapat menyebabkan jumlah langkah penyelesaian yang sangat banyak dan rentang waktu penyelesaian yang lama pula.

VII. Kesimpulan

Berdasarkan data dari hasil analisis di atas, maka dapat disimpulkan bahwa penggunaan algoritma runut-balik pada persoalan perpindahan gundu kurang sesuai, karena membutuhkan waktu yang lama dan jumlah langkah penyelesaian yang banyak dalam suatu pencarian solusi.
Di samping itu, algoritma *fast-swap* tanpa runut-balik yang telah diimplementasikan lebih mangkus daripada algoritma runut-balik karena membutuhkan rentang waktu yang lebih singkat dan jumlah langkah penyelesaian yang lebih sedikit.

VIII. Referensi

1. R. Munir, *Diktat Kuliah IF2251 Strategi Algoritmik*, Departemen Teknik Informatika ITB, Bandung, 2005.