

# Pencarian Jalan Keluar Terpendek Labirin Berujung Banyak

Husein Azmi El Firdausi

Program Studi Teknik Informatika, Institut Teknologi Bandung  
Jalan Ganesha 10 Bandung  
e-mail: if15100@students.if.itb.ac.id

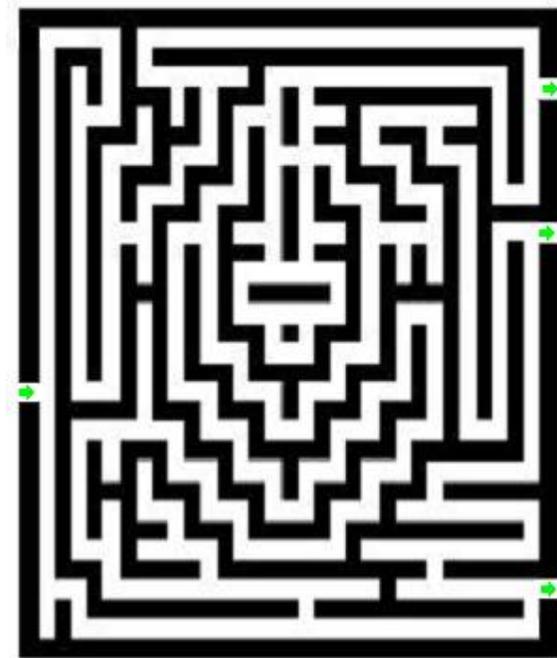
## ABSTRAK

Salah satu algoritma yang banyak diterapkan dalam mengambil keputusan adalah algoritma Program dinamis (*dynamic programming*). Dengan menggunakan algoritma ini, pengambilan suatu keputusan, yang cenderung melibatkan banyak kemungkinan, dapat diselesaikan dengan lebih sangkil. Hal ini dikarenakan, algoritma Program dinamis merupakan metode pengambilan keputusan dengan cara menguraikan tiap pilihan keputusan menjadi sekumpulan tahapan sedemikian sehingga keputusan yang diambil adalah keputusan terbaik yang didapat dari serangkaian tahapan yang berkaitan, bukan hanya didasarkan pada informasi lokal. Salah satu jenis pengambilan keputusan yang dapat diselesaikan dengan algoritma Program dinamis adalah pencarian jalan keluar dari suatu Labirin.

**Kata kunci:** Labirin, Program Dinamis, Keputusan, Optimal, Bobot, Tahap.

## 1. PENDAHULUAN

Labirin merupakan salah satu permasalahan yang cukup terkenal dalam sejarah kehidupan manusia. Pada zaman dahulu, banyak kerajaan yang menggunakan model Labirin untuk strategi pertahanan istana, menyembunyikan tempat rahasia, jalur pelarian, dan lain-lain, misalnya pada kerajaan di Yunani dan Mesir. Sedangkan Labirin masa kini lebih sering dinikmati sebagai sebuah permainan pemecahan masalah. Labirin biasanya berawal dari sebuah jalur, kemudian berakhir di beberapa jalur buntu, dan hanya satu jalur yang merupakan jalan keluar. Namun, untuk membuatnya menarik, Labirin dapat dibuat berujung banyak (ada banyak jalan keluar), sehingga misi penyelesaian masalah bertambah dari semula hanya mencari jalan keluar, menjadi mencari jalan keluar yang terpendek.



Gambar 1. Labirin berujung banyak

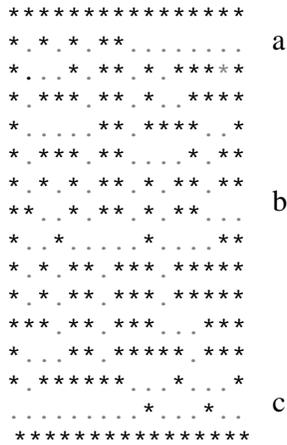
Masalah ini tentu akan sulit diselesaikan dengan algoritma runut balik, karena algoritma runut balik hanya mencari solusi yang mengarah ke satu solusi pertama. Padahal, akan ada banyak solusi pada permainan Labirin berujung banyak ini. Sehingga, penulis mencoba mengaplikasikan algoritma yang lain yaitu Program dinamis.

Proses pengambilan keputusan pada program dinamis dibuat dengan menggunakan Prinsip Optimalitas. [1] Prinsip ini berbunyi: jikasolusi total optimal, maka bagian solusi sampai tahap ke-k juga optimal. Prinsip Optimalitas berarti bahwa jika kita bekerja dari tahap k ke tahap k+1, kita dapat menggunakan hasil optimal dari tahap k tanpa harus kembali ke tahap awal. Jika pada setiap tahap kita menghitung bobot (dalam kasus Labirin = jarak tempuh, dalam kasus lain bisa berupa ongkos, dan sebagainya), maka dapat dirumuskan bahwa:

$$\text{bobot pada tahap } k + 1 = (\text{bobot yang dihasilkan pada tahap } k) + (\text{bobot dari tahap } k \text{ ke tahap } k+1) \quad (1)$$



Tiap pintu keluar kita tandai dengan huruf kecil a,b dan c. Simpul awal adalah 0. simpul berikutnya dari masing-masing simpul berbeda letaknya (namun bisa sama), tapi namanya sama.

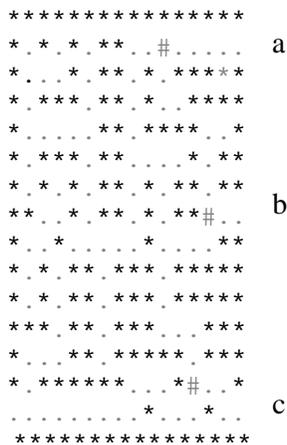


Gambar 3. Labirin pada keadaan awal

Tabel 1 Tahap pertama

| s | Solusi Optimum     |                    |                    |                |
|---|--------------------|--------------------|--------------------|----------------|
|   | f1(s) <sub>a</sub> | f1(s) <sub>b</sub> | f1(s) <sub>c</sub> | x <sub>1</sub> |
| 1 | 6                  | 3                  | 5                  | 0              |

Setelah didapat beberapa solusi dari tahap pertama (kebetulan pada saat pertama ini belum memilih jalur karena belum ada persimpangan). Kita letakkan penanda di tiap persimpangan yang ditemukan. Sehingga gambarnya berubah sebagai berikut:

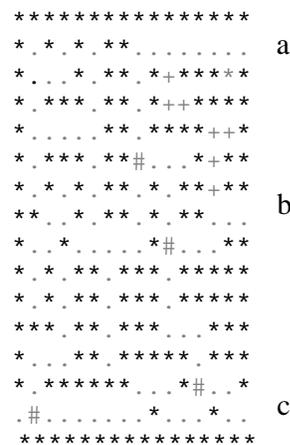


Gambar 4. Labirin pada keadaan 2

Tabel 2 Tahap kedua

| s | Solusi Optimum     |                    |                    |                 |                 |                 |
|---|--------------------|--------------------|--------------------|-----------------|-----------------|-----------------|
|   | f1(s) <sub>a</sub> | f1(s) <sub>b</sub> | f1(s) <sub>c</sub> | x <sub>2a</sub> | x <sub>2b</sub> | x <sub>2c</sub> |
| 2 | 12                 | 777                | 12                 | 12              | ~               | 12              |
| 3 | 777                | 7                  | 19                 | ~               | 7               | 19              |

Dari tabel 2 kita lihat bahwa telah terjadi beberapa pengeliminasi simpul, yang menunjukkan percabangan ke jalur yang buntu. Juga kita lihat bahwa simpul berikutnya dari x<sub>1b</sub> dan x<sub>1c</sub> sama untuk status yang berbeda, yaitu s<sub>3b</sub> dan s<sub>2c</sub>. Seperti pada tahap pertama, kita letakkan kembali penanda di tiap persimpangan yang ada, dan menghapus tanda di simpul persimpangan yang sudah tereliminasi. Sehingga gambarnya berubah menjadi demikian:

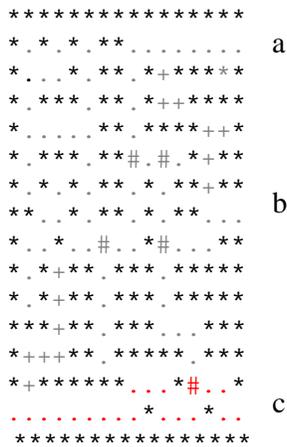


Gambar 5. Labirin pada keadaan 3

Tabel 3 Tahap ketiga

| s/s <sub>3</sub> | 2 <sub>a</sub> | 2 <sub>b</sub> | 2 <sub>c</sub> | 3 <sub>a</sub> | 3 <sub>b</sub> | 3 <sub>c</sub>  | Solusi Optimum     |                    |                    |                 |                 |                 |
|------------------|----------------|----------------|----------------|----------------|----------------|-----------------|--------------------|--------------------|--------------------|-----------------|-----------------|-----------------|
|                  |                |                |                |                |                |                 | f1(s) <sub>a</sub> | f1(s) <sub>b</sub> | f1(s) <sub>c</sub> | x <sub>3a</sub> | x <sub>3b</sub> | x <sub>3c</sub> |
| 4                | 18             | ~              | 777            | .              | .              | .               | 18                 | .                  | 15                 | 4               | .               | 5               |
| 5                | 21             | ~              | 15             | .              | .              | .               | .                  | 10                 | 20                 | .               | 6               | 7               |
| 6                | .              | .              | .              | ~              | 10             | 777             | .                  | 10                 | 20                 | .               | 6               | 7               |
| 7                | .              | .              | .              | ~              | 14             | 20 <sup>b</sup> | .                  | 10                 | 20                 | .               | 6               | 7               |

Dari tahap ketiga ini kita telah mendapatkan satu solusi pertama yaitu pada s(7) dari penelusuran simpul x<sub>3c</sub>. Kita belum dapat menetapkan jalur tersebut sebagai jalur solusi, karena pada jalur lain masih terdapat bobot yang lebih kecil daripada 20. Sehingga penelusuran dari jalur lain (yang bobotnya masih dibawah 20) tetap akan kita teruskan. Tidak lupa, kita beri tanda kembali untuk setiap simpul percabangan yang telah ditemukan. Sehingga gambarnya demikian:



Gambar 5. Labirin pada keadaan 3

Tabel 4 Tahap keempat

| s\X <sub>4</sub> | 4 <sub>a</sub> | 6 <sub>b</sub> | 7 <sub>b</sub> | Solusi Optimum     |                    |                 |                 |
|------------------|----------------|----------------|----------------|--------------------|--------------------|-----------------|-----------------|
|                  |                |                |                | f1(s) <sub>a</sub> | f1(s) <sub>b</sub> | X <sub>3a</sub> | X <sub>3b</sub> |
| 8                | 19             | 11             |                | 19                 | 11                 | 8               | 8               |
| 9                | 21             | 12             |                |                    |                    |                 |                 |
| 10               | .              | .              | 28             | .                  | 28                 | .               | 28              |

Dari tahap ke-empat ini kita dapat bahwa jalur solusi untuk  $x_b$ , ternyata telah ada yang melebihi jalur solusi total untuk  $x_c$ . Sehingga kita dapat mengeliminasi status yang bersangkutan dengan jalur tersebut. Demikian seterusnya sampai akhirnya kita temukan bahwa jalur solusi  $x_a$  dan  $x_b$  ternyata lebih panjang daripada jalur solusi  $x_c$ . Jadi kita baru dapat menetapkan jalan keluar terpendek dari Labirin berujung banyak diatas berjarak 20.

### 2.2.3 Kode Semu

```

procedure Jalan(Stack SJarak,Posisi
P,Labirin L)
{
    Pos = P;
    while
    (belumKetemuTandaSimpang(Pos,L))
    {
        Pos++; //posisi
        SJarak.Push(1); //jarak
        berpindah 1
        bertambah 1
    }
}

```

```

procedure Buntu(Posisi P, Labirin L)
{
    Pos = P;
    while
    (belumKetemuTandaSimpang(Pos,L))
    {
        Pos--;
    }
}

procedure cekJalan(Posisi P, Labirin L)
{
    if(adaSimpang(P,L))
    {
        Tandai(); //kasih tanda dan
        berhenti jalan
    }
    else if(adaJalan(P+1,L))
    {
        Jalan(P,L); //jalan terus
    }
    else //tidak ada jalan
    {
        Buntu(P,L); //kasih tanda
    }
}

buntu
}

```

### III. KESIMPULAN

Pengambilan keputusan untuk mencari jalan keluar terpendek dari Labirin berujung banyak dapat kita lakukan dengan optimal menggunakan algoritma Program dinamis. Pemilihan algoritma Program dinamis dalam menyelesaikan masalah pengambilan keputusan untuk mencari jalan keluar terpendek dari Labirin berujung banyak dapat dikatakan cukup baik jika dibandingkan dengan algoritma lain seperti algoritma *brute force* atau algoritma runut balik. Hal ini dikarenakan dalam Program dinamis, kita tidak mencari semua kemungkinan dan tidak pula mencari satu kemungkinan pertama yang didapat. Dengan menggunakan algoritma ini, pengambilan keputusan untuk mencari mencari jalan keluar terpendek dari Labirin berujung banyak dapat kita lakukan dengan optimal.

### REFERENSI

- [1] Munir Rinaldi, "Diktat Kuliah IF221 Strategi Algoritmik", Program Studi Teknik Informatika, 2007.
- [2] <http://forum.java.sun.com/forum.jspa?forumID=426&start=0>. Diakses pada hari Sabtu, 19 Mei 2007.