

Penerapan Algoritma Runut-Balik untuk Menyelesaikan *Zebra Puzzle*

Catherine Almira/13515111

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

13515111@std.stei.itb.ac.id, almiratjann@gmail.com

Abstrak—*Zebra puzzle* atau yang lebih dikenal dengan *Einstein's Riddle* adalah sebuah teka-teki yang terdiri dari kumpulan pernyataan, dimana setiap pernyataan tersebut merupakan petunjuk yang mengarah pada solusi yang diharapkan. Kemudian dari pernyataan yang ada, akan diberikan suatu pertanyaan yang memerlukan analisis yang mendalam dan logika yang kuat untuk dapat menyelesaikannya. *Zebra puzzle* sering diklaim hanya dapat diselesaikan oleh 2% dari populasi dunia. Dalam makalah ini akan dibahas mengenai cara menyelesaikan *zebra puzzle* dengan menggunakan algoritma runut-balik (*backtracking*).

Kata kunci—algoritma runut-balik; analisis; logika; *zebra puzzle*

I. PENDAHULUAN

Puzzle atau teka-teki merupakan permainan yang memerlukan analisis dan logika untuk dapat menyelesaikannya. Permainan ini sudah ada sejak zaman dahulu dan terus-menerus mengalami perkembangan dari masa ke masa. Salah satu teka-teki menarik yang pernah diciptakan adalah *zebra puzzle* atau yang lebih dikenal dengan sebutan *Einstein's Riddle*. *Zebra puzzle* adalah sebuah teka-teki yang terdiri dari pernyataan-pernyataan sedemikian rupa sehingga dapat membentuk suatu kesimpulan yang berupa pasangan yang sesuai dengan yang dimaksud dari pernyataan-pernyataan yang diberikan. Pada awalnya, pertanyaan dari teka-teki ini adalah "Siapakah yang memiliki zebra?" sehingga diberi nama *zebra puzzle*. Namun, seiring dengan berjalannya waktu, banyak versi baru dari *zebra puzzle* ini sehingga pertanyaan yang diajukan dapat bervariasi. Salah satu versi paling terkenal dari *zebra puzzle* dan disajikan pada makalah ini diterbitkan di dalam Majalah *Life International* pada tanggal 17 Desember 1962.

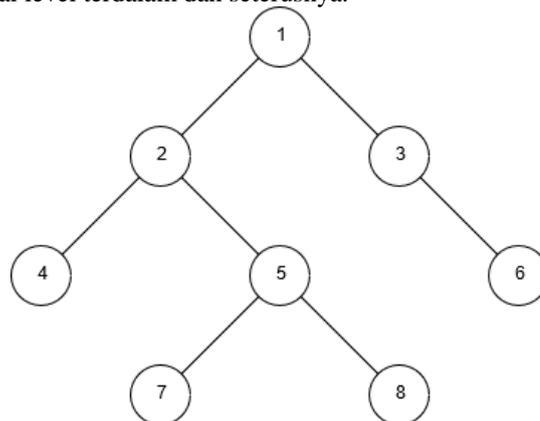
Zebra puzzle merupakan puzzle yang cukup menantang untuk diselesaikan karena puzzle ini diklaim hanya dapat dipecahkan oleh 2% penduduk di dunia [2]. Oleh karena itu, untuk mempermudah penyelesaian *zebra puzzle*, dapat digunakan algoritma runut-balik (*backtracking*). Pada makalah ini akan dibahas mengenai teori dasar algoritma runut-balik, teori dasar algoritma DFS (*depth-first search*) yang merupakan dasar dari algoritma runut-balik, penjelasan singkat mengenai *zebra puzzle*, dan cara menyelesaikan *zebra puzzle* dengan menggunakan algoritma runut-balik.

II. DASAR TEORI

A. Algoritma DFS (*Depth-First Search*)

Algoritma DFS adalah salah satu algoritma yang dapat digunakan untuk menelusuri graf. Pencarian solusi dengan DFS dicirikan oleh ekspansi simpul secara mendalam terlebih dahulu. Simpul ditelusuri dari simpul akar kemudian ke salah satu simpul anaknya (misalnya prioritas penelusuran berdasarkan simpul dengan nomor terkecil), maka penelusuran dilakukan terus melalui simpul anak pertama dari simpul anak pertama level sebelumnya hingga mencapai level terdalam.

Setelah sampai di level terdalam, penelusuran akan kembali ke satu level sebelumnya untuk menelusuri simpul anak kedua pada pohon biner lalu kembali ke langkah sebelumnya dengan menelusuri simpul anak pertama lagi sampai level terdalam dan seterusnya.



Gambar 1. Graf yang dikunjungi

Penelusuran dengan menggunakan algoritma DFS pada graf (gambar 1) akan menghasilkan urutan 1-2-4-5-7-8-3-6.

B. Algoritma Runut-Balik

Istilah runut-balik (*backtracking*) pertama kali diperkenalkan oleh D.H. Lehmer pada tahun 1950. Algoritma runut-balik merupakan perbaikan dari algoritma *brute force* dalam pencarian solusi terhadap suatu persoalan. Algoritma ini

berbasis pada algoritma DFS (*depth-first search*). Hanya pencarian yang mengarah pada solusi saja yang akan dipertimbangkan untuk dikembangkan pada pohon pencarian. Secara umum, algoritma runut-balik dinyatakan dengan menggunakan algoritma rekursif. Algoritma runut balik dapat diterapkan pada berbagai permainan seperti tic-tac-toe, labirin, dan masalah-masalah pada bidang kecerdasan buatan.

C. Properti Umum Metode Runut-Balik

Untuk menerapkan metode runut-balik, property berikut didefinisikan:

- Solusi persoalan

Solusi persoalan dinyatakan sebagai vektor dengan *n-tuple*: $X = (x_1, x_2, \dots, x_n)$, $x_i \in$ himpunan berhingga S_i . Mungkin saja $S_1 = S_2 = \dots = S_n$.

Contoh: $S_i = \{0, 1\}$, $x_i = 0$ atau 1.

- Fungsi pembangkit nilai x_k

Fungsi pembangkit nilai dinyatakan sebagai $T(k)$. $T(k)$ membangkitkan nilai untuk x_k , yang merupakan komponen vektor solusi.

- Fungsi pembatas

Fungsi pembatas atau fungsi kriteria dinyatakan sebagai $B(x_1, x_2, \dots, x_k)$. Fungsi pembatas menentukan apakah (x_1, x_2, \dots, x_k) mengarah ke solusi. Jika ya, maka pembangkitan nilai untuk x_{k+1} dilanjutkan, tetapi jika tidak, maka (x_1, x_2, \dots, x_k) dibuang dan tidak dipertimbangkan lagi dalam pencarian solusi. Fungsi pembatas tidak selalu dinyatakan sebagai fungsi matematis. Ia dapat dinyatakan sebagai predikat yang bernilai *true* atau *false*, atau dalam bentuk lain yang ekuivalen^[1].

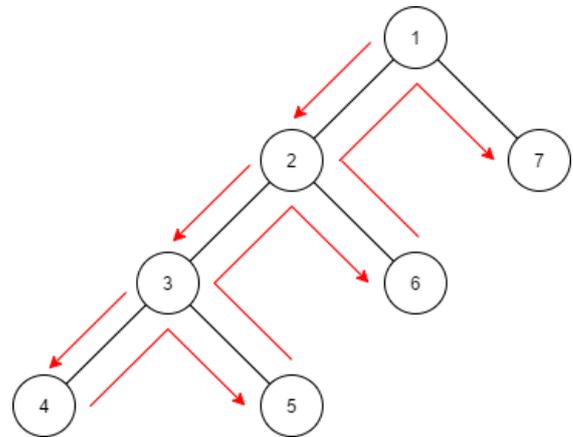
D. Prinsip Pencarian Solusi dengan Metode Runut-Balik

Langkah-langkah pencarian solusi pada pohon ruang status yang dibangun secara dinamis adalah sebagai berikut.

- Solusi dicari dengan membentuk lintasan dari akar ke daun. Aturan pembentukan yang dipakai yang dipakai mengikuti metode pencarian mendalam (DFS). Simpul-simpul yang sudah dilahirkan dinamakan simpul hidup. Simpul hidup yang sedang diperluas dinamakan simpul-E (*Expand-node*). Simpul dinomori dari atas ke bawah sesuai dengan urutan kelahirannya pada metode DFS.
- Setiap kali simpul-E diperluas, lintasan yang dibangun olehnya bertambah panjang. Jika lintasan yang sedang dibentuk tidak mengarah ke solusi, maka simpul-E tersebut “dibunuh” sehingga menjadi simpul mati. Fungsi pembatas digunakan untuk membatasi ekspansi yang dilakukan terhadap suatu simpul. Simpul yang sudah mati tidak akan diperluas lagi.
- Jika pembentukan lintasan berakhir dengan simpul mati, maka proses pencarian dilakukan dengan membangkitkan simpul anak yang lainnya. Bila tidak ada lagi simpul anak yang bisa dibangkitkan, maka pencarian solusi dilanjutkan dengan melakukan runut-

balik ke simpul hidup terdekat (simpul orangtua). Selanjutnya, simpul tersebut menjadi simpul-E yang baru. Lintasan baru dibangun kembali sampai lintasan tersebut membentuk solusi.

- Pencarian dihentikan apabila solusi telah ditemukan atau tidak ada lagi simpul hidup untuk runut-balik^[1].



Gambar 2. Prinsip pencarian solusi pada algoritma runut balik

E. Skema Umum Algoritma Runut-Balik

Algoritma runut-balik dapat diimplementasikan dalam dua versi algoritma, yaitu versi rekursif dan versi iteratif. Skema dalam versi rekursif lebih tepat karena algoritma runut-balik lebih alami dinyatakan dalam bentuk rekursi. Berikut ini adalah skema algoritma runut-balik versi rekursif.

```

procedure RunutBalik(input k : integer)
{Mencari semua solusi persoalan dengan metode runut-balik skema rekursif.
Masukan: k, yaitu indeks komponen vektor solusi, x[k]
Keluaran: solusi x = {x[1], x[2], ..., x[n]}

Algoritma

for tiap x[k] yang belum dicoba sedemikian sehingga (x[k] ← T[k]) and B(x[1], x[2], ..., x[k]) = true do
  if (x[1], x[2], ..., x[k]) adalah lintasan dari akar ke daun then
    for k ← 1 to n do
      write(x[k])
    endfor
  endif
  RunutBalik(k+1) {Tentukan nilai untuk x[k+1]}
endfor

```

III. ZEBRA PUZZLE

Zebra puzzle biasa disebut juga *Einstein's Riddle* karena dikatakan bahwa pencipta teka-teki tersebut adalah Albert Einstein^[2], tetapi teka-teki tersebut juga sering dikaitkan dengan Lewis Carroll^[3]. Namun, hingga saat ini belum ada bukti yang sah mengenai pengaruh Albert Einstein maupun

Lewis Carroll terhadap *zebra puzzle*. Teka-teki tersebut sering diklaim hanya dapat diselesaikan oleh 2% populasi dunia ^[4].

Terdapat berbagai versi dari *zebra puzzle*. Salah satu yang paling terkenal adalah versi yang diterbitkan di majalah *Life International* pada tanggal 17 Desember 1962. Berikut ini adalah teka-tekinya.

1. Terdapat lima buah rumah.
2. Orang Inggris tinggal di dalam rumah berwarna merah.
3. Orang Spanyol memelihara anjing.
4. Kopi diminum di dalam rumah berwarna hijau.
5. Orang Ukraina minum teh.
6. Rumah berwarna hijau berada tepat di sebelah kanan rumah berwarna putih.
7. Orang yang merokok Old Gold memelihara siput.
8. Orang yang tinggal di rumah berwarna kuning merokok Kools.
9. Orang yang tinggal di rumah yang berada di posisi tengah meminum susu.
10. Orang Norwegia tinggal di rumah pertama.
11. Orang yang merokok Chesterfields tinggal di sebelah orang yang memelihara serigala.
12. Orang yang merokok Kools tinggal di sebelah orang yang memelihara kuda.
13. Orang yang merokok Lucky Strike meminum jus jeruk.
14. Orang Jepang merokok Parliaments.
15. Orang Norwegia tinggal di sebelah rumah berwarna biru.

Dari 15 buah pernyataan tersebut, terdapat dua buah pertanyaan. Pertanyaan pertama adalah “Siapa yang meminum air tawar?” Pertanyaan kedua adalah ”Siapa yang memelihara zebra?” Dalam teka-teki ini, setiap kebangsaan, warna rumah, minuman, jenis rokok, dan binatang peliharaan masing-masing hanya terdapat satu buah. Misalnya, apabila orang Spanyol memelihara anjing, maka ia tidak akan memelihara kuda, siput, serigala, maupun zebra.

IV. ANALISIS PEMECAHAN MASALAH

Masalah utama dari *zebra puzzle* versi majalah *Life International* adalah menjawab dua buah pertanyaan, yaitu “Siapa yang meminum air tawar” dan “Siapa yang memelihara zebra”, berdasarkan 15 buah pernyataan yang diberikan. Untuk menyelesaikannya, seluruh tipe (kebangsaan, jenis minuman, warna rumah, jenis rokok, dan hewan peliharaan) perlu diketahui terlebih dahulu pasangannya. Salah satu cara untuk mengetahui setiap pasangan dari masing-masing tipe adalah dengan menggunakan algoritma runut-balik.

Langkah pertama yang harus dilakukan adalah menyederhanakan 15 buah pernyataan dalam teka-teki tersebut sehingga dapat diselesaikan dengan cara komputasi. Agar lebih

sederhana, digunakan matriks seperti pada tabel 1 untuk menentukan masing-masing pasangan dari seluruh tipe. Nomor 1 sampai 5 pada baris pertama matriks merupakan nomor untuk posisi rumah. Selanjutnya, dibuat notasi untuk atribut dari masing-masing tipe seperti pada tabel 2. Atribut diperoleh dari 15 pernyataan ditambah dengan air tawar dan zebra yang terdapat pada pertanyaan.

TABEL I. ILUSTRASI MATRIKS

	1	2	3	4	5
Kebangsaan					
Minuman					
Warna					
Rokok					
Peliharaan					

TABEL II. ATRIBUT DARI SETIAP TIPE DAN NOTASINYA

Atribut	Notasi	Atribut	Notasi
Kebangsaan		Rokok	
Inggris	K1	Old Gold	R1
Spanyol	K2	Chesterfields	R2
Ukraina	K3	Kools	R3
Norwegia	K4	Lucky strike	R4
Jepang	K5	Parliaments	R5
Minuman		Peliharaan	
Kopi	M1	Anjing	P1
Teh	M2	Siput	P2
Susu	M3	Serigala	P3
Jus jeruk	M4	Kuda	P4
Air tawar	M5	Zebra	P5
Warna			
Merah	W1		
Hijau	W2		
Putih	W3		
Kuning	W4		
Biru	W5		

Langkah selanjutnya adalah mengubah 15 buah pernyataan yang merupakan petunjuk dari teka-teki tersebut ke dalam bentuk yang dapat dipahami oleh komputer. Struktur data yang paling tepat untuk menyelesaikan persoalan ini adalah struktur data matriks dengan baris menyatakan setiap tipe dan kolom menyatakan masing-masing nomor rumah. Kebangsaan, minuman, warna, rokok, dan peliharaan secara berturut-turut direpresentasikan oleh baris 1 sampai dengan 5. Berikut ini adalah hasil perubahan setiap pernyataan dalam teka-teki tersebut menjadi bentuk yang dapat diselesaikan secara komputasi dimulai dari pernyataan nomor 2.

2. Orang Inggris (K1) tinggal di dalam rumah berwarna merah(W1).
kolom(K1) = kolom(W1)
3. Orang Spanyol (K2) memelihara anjing (P1).
kolom(K2) = kolom(P1)

4. Kopi (M1) diminum di dalam rumah berwarna hijau (W2).
kolom(M1) = kolom(W2)
5. Orang Ukraina minum teh.
kolom(K3) = kolom(M2)
6. Rumah berwarna hijau (W2) berada tepat di sebelah kanan rumah berwarna putih (W3).
kolom(W2) - kolom(W3) = 1
7. Orang yang merokok Old Gold (R1) memelihara siput (P2).
kolom(R1) = kolom(P2)
8. Orang yang tinggal di rumah berwarna kuning (W4) merokok Kools (R2).
kolom(W4) = kolom(R2)
9. Orang yang tinggal di rumah yang berada di posisi tengah meminum susu (M3).
kolom(M3) = 3
10. Orang Norwegia (K4) tinggal di rumah pertama.
kolom(K4) = 1
11. Orang yang merokok Chesterfields (R3) tinggal di sebelah orang yang memelihara serigala (P3).
| kolom(R3) - kolom(P3) | = 1
12. Orang yang merokok Kools (R2) tinggal di sebelah orang yang memelihara kuda (P4).
| kolom(R2) - kolom(P4) | = 1
13. Orang yang merokok Lucky Strike (R4) meminum jus jeruk (M4).
kolom(R4) = kolom(M4)
14. Orang Jepang (K5) merokok Parliaments (R5).
kolom(K5) = kolom(R5)
15. Orang Norwegia (K4) tinggal di sebelah rumah berwarna biru (W5).
| kolom(K4) - kolom(W5) | = 1

Pernyataan-pernyataan tersebut merupakan fungsi pembatas untuk menerapkan algoritma runut-balik. Berikut ini adalah langkah-langkah pengerjaan *zebra puzzle* dengan menggunakan algoritma runut-balik.

1. Letakkan atribut kebangsaan K1 pada baris pertama kolom pertama. Cek apakah terjadi pelanggaran terhadap fungsi pembatas.
2. Apabila tidak terjadi pelanggaran, buang atribut yang sudah diletakkan tersebut. Kemudian letakkan atribut minuman M1 pada baris kedua kolom pertama. Cek apakah terjadi pelanggaran terhadap fungsi pembatas.
3. Apabila tidak terjadi pelanggaran, buang atribut yang sudah diletakkan tersebut. Kemudian letakkan atribut

warna W1 pada baris ketiga kolom pertama. Cek apakah terjadi pelanggaran terhadap fungsi pembatas.

4. Apabila tidak terjadi pelanggaran, buang atribut yang sudah diletakkan tersebut. Kemudian letakkan atribut rokok R1 pada baris keempat kolom pertama. Cek apakah terjadi pelanggaran terhadap fungsi pembatas.
5. Apabila tidak terjadi pelanggaran, buang atribut yang sudah diletakkan tersebut. Kemudian letakkan atribut makanan P1 pada baris kelima kolom pertama. Cek apakah terjadi pelanggaran terhadap fungsi pembatas.
6. Apabila terjadi pelanggaran, kembali ke langkah saat peletakkan atribut yang melanggar fungsi pembatas dan mengganti atribut yang diletakkan menjadi Xn+1. dengan X adalah kode atribut dan n adalah nilai terakhir atribut yang melanggar fungsi pembatas. Kemudian ikuti langkah-langkah selanjutnya.
7. Apabila tidak ada atribut yang cocok (melanggar fungsi pembatas), lakukan runut-balik. Posisi yang ingin ditempati atribut saat ini dibiarkan kosong. Lalu atribut yang sebelumnya telah diletakkan di dalam matriks dihapus dan dikembalikan pada daftar atribut yang belum terpakai. Kemudian letakkan atribut selanjutnya dengan tipe yang sama seperti atribut yang telah dihapus.
8. Ulangi langkah 1 untuk atribut berikutnya hingga ditemukan solusi yang tidak melakukan pelanggaran terhadap seluruh fungsi pembatas. Dalam pemasangan atribut, dipastikan pemasangan sesuai dengan tipenya. Misalnya, atribut K1, K2, K3, K4, dan K5 hanya akan ditempatkan pada tipe kebangsaan. Begitu juga untuk atribut dan tipe yang lain.

V. IMPLEMENTASI DAN PENGUJIAN

A. Implementasi

Implementasi algoritma runut-balik untuk penyelesaian *zebra puzzle* ini dilakukan dengan menggunakan bahasa pemrograman C++. Berikut ini adalah *pseudo code* algoritma runut-balik untuk menyelesaikan persoalan *zebra puzzle* tersebut.

```
function SolveZebraPuzzle(input/output m :
Matriks, input/output counter : integer)

{Mencari solusi zebra puzzle dengan menggunakan
algoritma runut-balik versi rekursif.
Masukan: m, yaitu matriks yang merupakan
struktur data yang digunakan dalam proses
pencarian solusi.
counter, yaitu nilai yang menyatakan jumlah
pengulangan pada fungsi rekursif.
Diinisialisasi dengan nilai 0 sebelum
pemanggilan fungsi.
Keluaran: m, berisi seluruh atribut dari
masing-masing tipe sesuai dengan pasangannya
counter, berisi jumlah pengulangan yang
terjadi.}

Deklarasi
```

```

row, col, num : integer

Algoritma
count ← count + 1

{Apabila seluruh elemen matriks telah terisi,
maka pengulangan selesai}
if (not FindUnassignedLocation(grid, row, col))
then
  → true

{Melakukan pengulangan untuk atribut 1 sampai
5}
for num ← 1 to 5 do
  {Apabila tidak melanggar fungsi pembatas}
  if (IsSafe(grid, row, col, num)) then
    {Mengisi nilai pada sel matriks}
    grid[row][col] ← num

    {Melakukan pemanggilan rekursif}
    if (SolveSudoku(grid, count)) then
      → true

  {Nilai sel matriks diisi dengan nilai
awal}
  grid[row][col] ← 0

→ false {melakukan runut balik}

```

Fungsi FindUnassignedLocation menghasilkan nilai true apabila terdapat atribut yang masih belum diisi. Fungsi tersebut memiliki parameter keluaran, berupa nilai baris dan kolom pada matriks yang belum diisi. Fungsi IsSafe merupakan fungsi pembatas dalam algoritma runut-balik tersebut. Fungsi tersebut berisi 14 pernyataan dalam teka-teki beserta tambahan bahwa setiap baris tidak boleh berisi angka yang sama.

B. Pengujian dan Analisis Hasil Pengujian

Pengujian dilakukan dengan menggunakan laptop Asus Core i3, RAM 2 GB dan sistem operasi Windows 8.1. Pengujian dilakukan dua kali dengan menggunakan algoritma runut-balik standar dan algoritma runut-balik dengan modifikasi. Modifikasi yang dilakukan berupa penentuan prioritas pembangkitan simpul solusi. Pernyataan yang langsung mengarah pada solusi seperti pernyataan nomor 9 yang berisi "Orang yang tinggal di rumah yang berada di posisi tengah minum susu(M3). kolom(M3) = 3" dan nomor 10 yang berbunyi "Orang Norwegia (K4) tinggal di rumah pertama. kolom(K4) = 1" pada *zebra puzzle* versi majalah *Life International* ditetapkan sebagai prioritas awal.

Dari pernyataan nomor 9, diperoleh informasi bahwa pada baris minuman (baris ke-2) kolom ke-3 akan berisi M3. Sedangkan, dari pernyataan nomor 10, diperoleh informasi bahwa pada baris kebangsaan (baris ke-1) kolom ke-1 akan berisi K4. Dua informasi tersebut ditetapkan sebagai atribut yang mutlak sehingga dalam penggunaan algoritma runut-balik kedua atribut tersebut tidak akan digantikan dengan atribut yang lain. Kemudian solusi yang diperoleh dari pernyataan-pernyataan lainnya akan dibangkitkan kemudian sesuai dengan skema runut-balik standar dengan kondisi matriks awal seperti pada tabel III.

TABEL III. ILUSTRASI MATRIKS AWAL UNTUK SKEMA RUNUT-BALIK DENGAN MODIFIKASI

	1	2	3	4	5
Kebangsaan	K4				
Minuman			M3		
Warna					
Rokok					
Peliharaan					

```

Kebangsaan 1 2 3 4 5
Minuman     4 3 1 2 5
Warna       5 2 3 4 1
Rokok       4 5 1 3 2
Peliharaan  2 3 1 4 5

Jumlah pengulangan: 1523
Waktu eksekusi: 8.00 ms

```

Gambar 3. Hasil eksekusi dengan algoritma runut-balik standar

```

Kebangsaan 1 2 3 4 5
Minuman     4 3 1 2 5
Warna       5 2 3 4 1
Rokok       4 5 1 3 2
Peliharaan  2 3 1 4 5

Jumlah pengulangan: 1235
Waktu eksekusi: 7.00 ms

```

Gambar 4. Hasil eksekusi dengan algoritma runut-balik yang dimodifikasi

Dari dua buah hasil eksekusi tersebut terlihat bahwa penggunaan algoritma runut-balik yang telah dimodifikasi menghasilkan jumlah pengulangan yang lebih sedikit dan waktu eksekusi yang lebih singkat dibandingkan algoritma runut-balik standar. Pada algoritma runut-balik yang dimodifikasi jumlah pengulangan berkurang sebanyak 288 kali. Sementara, waktu eksekusi yang semula 8,00 ms berkurang menjadi 7,00 ms. Hal tersebut terjadi karena pemilihan prioritas yang dilakukan pada algoritma runut-balik yang telah dimodifikasi mengurangi kemungkinan solusi yang ada. Jumlah pengulangan yang terjadi berbanding lurus dengan waktu eksekusi. Semakin banyak pengulangan, waktu eksekusi pun semakin meningkat. Sebaliknya, semakin sedikit jumlah pengulangan, waktu eksekusi semakin singkat.

Solusi *zebra puzzle* yang diharapkan dari kedua buah pengujian tersebut menghasilkan hasil akhir yang sama. Solusi *zebra puzzle* tersebut dapat dilihat dengan lebih jelas pada tabel IV.

TABEL IV. SOLUSI ZEBRA PUZZLE

	1	2	3	4	5
Kebangsaan	K4	K3	K1	K2	K5
Minuman	M5	M2	M3	M4	M1
Warna	W4	W5	W1	W3	W2
Rokok	R2	R3	R1	R4	R5
Peliharaan	P3	P4	P2	P1	P5

Berdasarkan hasil pengujian, dapat diambil kesimpulan untuk menjawab teka-teki *zebra puzzle*. Pertanyaan pertama adalah “Siapakah yang meminum air tawar?” Jawaban dari pertanyaan tersebut dapat diperoleh dengan melihat solusi pada tabel IV. Air tawar dinotasikan sebagai M5 pada tabel tersebut. Oleh karena itu, orang yang meminum air tawar adalah orang yang tinggal di rumah nomor 1. Orang tersebut berkebangsaan K4 (Norwegia), tinggal di rumah berwarna W4 (kuning), merokok R2 (Kools), dan memelihara P2 (siput). Pertanyaan kedua yaitu “Siapakah yang memelihara zebra?” Zebra dinotasikan sebagai P5. P5 berada pada kolom ke-5 sehingga orang yang tinggal di rumah nomor 5 merupakan jawaban dari pertanyaan kedua. Orang tersebut berkebangsaan K5 (Jepang), suka meminum M1 (kopi), tinggal di rumah berwarna W2 (hijau), dan merokok R5 (Parliaments).

VI. KESIMPULAN DAN SARAN

A. Kesimpulan

Algoritma runut-balik (*backtracking*) terdiri dari dua buah versi, yaitu versi iteratif dan versi rekursif. Secara umum, versi rekursif lebih alamiah untuk algoritma runut-balik. Algoritma runut balik dengan skema rekursif ini dapat digunakan untuk mencari solusi dari *zebra puzzle*. Waktu yang diperlukan untuk menemukan solusi dari suatu persoalan *zebra puzzle* relatif singkat, kurang dari 10 ms. Modifikasi dari algoritma runut-balik dengan melakukan penentuan prioritas pencarian solusi dapat meningkatkan performansi algoritma ini dalam memecahkan persoalan. Prioritas pencarian solusi dimulai dari pernyataan yang tidak menimbulkan amiguitas dan langsung mengarah pada solusi. Setelah seluruh pernyataan tersebut ditemukan solusinya, algoritma runut-balik dapat digunakan seperti biasa untuk pernyataan-pernyataan lainnya. Dengan modifikasi tersebut, pencarian solusi dapat lebih optimal, waktu eksekusi lebih singkat, dan jumlah pengulangan yang digunakan untuk runut-balik lebih sedikit.

B. Saran

Saat ini, permainan *zebra puzzle* sudah semakin berkembang dan kini terdapat *zebra puzzle* dengan ukuran matriks 8x8. Untuk persoalan *zebra puzzle* ukuran 5x5 waktu eksekusinya masih relatif singkat. Namun, pada persoalan dengan ukuran matriks 8x8 tentu waktu eksekusinya akan bertambah secara eksponensial. Oleh karena itu, peningkatan performansi pada algoritma runut-balik perlu dilakukan. Salah satu caranya adalah dengan melakukan modifikasi lebih lanjut

terhadap prioritas penentuan solusi. Tidak hanya pernyataan yang langsung mengarah pada solusi saja yang dikerjakan terlebih dahulu, tetapi juga pernyataan yang dapat ditemukan solusinya ketika solusi lain sudah ditemukan. Algoritma runut-balik dengan MRV merupakan salah satu alternatif yang dapat digunakan.

UCAPAN TERIMA KASIH

Pertama-tama, Penulis ingin mengucapkan syukur kepada Tuhan Yang Maha Esa atas berkat dan rahmat yang telah diberikanNya kepada Penulis sehingga pembuatan makalah ini dapat selesai. Penulis juga berterima kasih kepada kedua orangtua yang selalu mendukung Penulis baik secara langsung, melalui pembiayaan pendidikan, maupun tidak langsung. Penulis juga ingin mengucapkan terima kasih kepada Bapak Rinaldi Munir, selaku dosen mata kuliah Strategi Algoritma, yang telah memberikan ilmu mengenai Strategi Algoritma, termasuk di dalamnya teori mengenai algoritma runut-balik.

REFERENSI

- [1] Munir, Rinaldi. 2009. *Diktat Kuliah IF2211 Strategi Algoritma*. Bandung: Penerbit Informatika.
- [2] Stangroom, Jeremy. 2009. *Einstein's Riddle: Riddles, Paradoxes, and Conundrums to Stretch Your Mind*. Bloomsbury USA.
- [3] James Little; Cormac Gebruers; Derek Bridge & Eugene Freuder. *Capturing Constraint Programming Experience: A Case-Based Approach*. Cork Constraint Computation Centre, University College, Cork, Ireland.
- [4] <https://web.stanford.edu/~laurik/fsmbook/examples/Einstein%27sPuzzle.html> diakses pada tanggal 15 Mei 2017 pukul 16.01

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, 15 Mei 2017



Catherine Almira/13515111