

# ALGORITMA RUNUT BALIK DALAM PENYELESAIAN PERMAINAN WORD DIAGRAM

Ivan Saputra

Mahasiswa Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
e-mail: if15091@students.if.itb.ac.id

## ABSTRAK

Permainan *Word Diagram* adalah permainan mencari kata-kata yang terdapat pada sebuah matriks dua dimensi dengan arah vertikal, horisontal, maupun diagonal dari sebuah seranai kata yang akan dicari. Pemain dinyatakan menang apabila pemain dapat menemukan seluruh kata yang terdapat di matriks dari seranai kata soal. Kita dapat menggunakan berbagai algoritma yang telah ada untuk mencari penyelesaian dari permainan ini. Salah satunya algoritma yang akan digunakan dalam makalah ini adalah algoritma runut balik. Algoritma runut balik ini berbasis pada algoritma *DFS (Depth First Search)* dimana persoalan akan dibagi menjadi beberapa ruang solusi. Hanya pencarian yang mengarah ke solusi yang akan dipertimbangkan. Apabila simpul ruang solusi tidak lagi mengarah pada penyelesaian, simpul itu akan dimatikan, lalu dilakukan runut balik ke simpul ruang solusi yang masih hidup. Algoritma runut balik dilakukan pada setiap kata dari seranai kata yang akan dicari sampai semua kata ditemukan pada matriks.

**Kata kunci:** Word Diagram, *DFS*, runut balik.

## 1. PENDAHULUAN

Permainan *Word Diagram* atau sering juga disebut sebagai *Word Searching Puzzle* telah ada sejak dahulu dan semakin berkembang sekarang ini. Karena permainan ini ringan dan mudah dimainkan tetapi juga cukup menantang, maka permainan ini menjadi cukup populer di kalangan para pengguna komputer. Tujuan permainan ini adalah mencari kata-kata pada sebuah matriks dua dimensi. Biasa kata-kata yang dicari adalah kata-kata dalam bahasa Inggris. Oleh karena itu, permainan ini juga bermanfaat menambah perbendaharaan kata dalam bahasa Inggris.

Pencarian kata dapat dilakukan secara vertikal, horisontal, maupun diagonal. Kata-kata yang akan dicari telah ditentukan sebelumnya pada seranai kata yang

tertulis di luar matriks. Permainan dibatasi oleh *timer* atau penghitung waktu. Permainan berakhir bila waktu telah habis atau pemain dapat menemukan seluruh kata pada matriks dua dimensi tersebut.

Untuk mencari penyelesaian dari permainan ini dapat digunakan berbagai macam algoritma. Salah satunya adalah algoritma runut balik. Pencarian dilakukan terhadap kata-kata dalam seranai kata di dalam sebuah matriks dua dimensi. Dimulai dari huruf per huruf yang bersesuaian. Apabila terdapat huruf yang tidak cocok, maka akan ditinggalkan dan beralih ke huruf lain.

## 2. ISI

Dalam bab ini akan dijelaskan lebih detail mengenai deskripsi permainan, cara memainkannya, algoritma runut balik yang akan digunakan, serta penerapan algoritma runut balik untuk mencari penyelesaian pada permainan *Word Diagram*.

### 2.1 Deskripsi Masalah

Permainan *Word Diagram* atau juga disebut *Word Searching Puzzle* bertujuan untuk mencari kata – kata di dalam sebuah matriks dua dimensi yang tersusun dari huruf – huruf dari sebuah seranai kata. Huruf - huruf di dalam matriks dapat tersusun secara horisontal, vertikal, maupun diagonal dengan arah kiri ke kanan, kanan ke kiri, atas ke bawah, bawah ke atas, kiri bawah ke kanan atas, kanan bawah ke kiri atas, kiri atas ke kanan bawah, serta kanan bawah ke kiri atas.

Untuk menggambarkan permainan ini dalam dunia nyata, penulis menggunakan software permainan bernama *Babel Deluxe* dimana terdapat permainan *Word Diagram* sebagai salah satu bagian dari permainan – permainan dalam *Babel Deluxe*.

Pada awal permainan pemain akan diberi matriks dua dimensi yang tersusun oleh huruf – huruf di tengah layar, seranai kata di sebelah kanan matriks, serta sebuah penghitung waktu yang dinyatakan dalam jam pasir di bagian kiri bawah. Ukuran dari matriks tergantung pada

tingkat kesulitan yang sedang dimainkan. Semakin tinggi tingkat kesulitan, semakin besar matriks yang diberikan, dan semakin banyak juga seranai kata yang akan dicari. Pada tampilan berikut adalah contoh tampilan permainan *Word Diagram* pada awal permainan.



Gambar 1. Status Permainan *Word Diagram* pada awal permainan

Pemain dapat memainkan permainan ini dengan membuat suatu garis melewati huruf – huruf pada matriks. Garis dapat diletakkan di huruf awal sampai ke huruf akhir dengan arah vertikal, horisontal, maupun diagonal. Huruf – huruf yang dilewati garis tersebut harus membentuk kata yang sesuai dengan salah satu kata pada seranai kata di sebelah kanan matriks. Apabila benar, kata yang bersesuaian pada seranai kata di sebelah kanan akan hilang dan pemain dapat mencari kata yang masih terdapat pada seranai kata tersebut. Berikut adalah tampilan ketika semua kata dalam seranai kata telah ditemukan.



Gambar 2. Status Permainan *Word Diagram* pada akhir permainan

## 2.2 Algoritma Runut - balik

Istilah runut – balik pertama kali diperkenalkan oleh D.H. Lehmer pada tahun 1950. Selanjutnya, R.J. Walker, Golomb, dan Baumert menyajikan uraian umum tentang runut – balik dan penerapannya pada berbagai persoalan [MUN07].

Algoritma runut – balik adalah perbaikan dari algoritma *brute – force* yang berbasis pada algoritma DFS ( *Depth First Search* ). Perbedaan dengan algoritma *brute – force* adalah pada algoritma ini tidak diperiksa semua kemungkinan solusi yang ada, hanya kemungkinan yang mengarah pada solusi saja yang dicari. Sehingga waktu pencarian dapat dihemat. Pada akhir pencarian, hanya satu solusi yang akan muncul sebagai hasil pencarian dengan algoritma runut – balik. Algoritma runut – balik ini biasa dinyatakan dalam algoritma rekursif. Bisa juga dinyatakan secara iteratif, namun algoritma runut – balik merupakan tipikal dari algoritma rekursif.

Properti – properti dalam algoritma runut – balik adalah sebagai berikut :

- Solusi persoalan  
Solusi dinyatakan sebagai vektor dengan  $n$ -tuple :

$$X = (x_1, x_2, x_3, \dots, x_n), x_i \in \text{himpunan berhingga } S_i. \\ \text{Mungkin saja } S_1 = S_2 = \dots = S_n \quad (1)$$

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

- Fungsi pembangkit nilai  $x_k$   
Dinyatakan sebagai :  
 $T(k)$   
 $T(k)$  membangkitkan nilai untuk  $x_k$ , yang merupakan komponen vektor solusi.
- Fungsi pembatas (pada beberapa persoalan fungsi ini dinamakan 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 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. [MUN07]

Algoritma runut – balik membuat simpul – simpul ruang solusi yang akan ditelusuri secara pencarian mendalam dengan membentuk lintasan dari akar ke daun. Simpul – simpul yang sudah dilahirkan dinamakan simpul

hidup ( *live node* ). Simpul hidup yang sedang diperluas disebut simpul-E ( *Expand node* ). Apabila sampai pada simpul yang tidak mengarah ke solusi, simpul akan dimatikan menjadi simpul mati ( *dead node* ) dengan menggunakan fungsi pembatas. Lalu pencarian dilakukan dengan membangkitkan simpul anak yang lain. Bila tidak ada lagi simpul anak yang dapat dibangkitkan, maka akan dilakukan runut – balik ke simpul hidup terdekat. Pencarian dihentikan apabila telah ditemukan solusi atau tidak ada lagi simpul hidup.

### 2.3 Penerapan Algoritma Runut - balik

Untuk menemukan penyelesaian dari permainan *Word Diagram* dengan algoritma runut – balik ini, penulis membagi ke dalam n persoalan dimana n adalah jumlah kata dalam seranai kata yang akan dicari. Masing – masing persoalan tersebut kemudian dibagi lagi menjadi langkah – langkah sebagai berikut :

#### Langkah 1

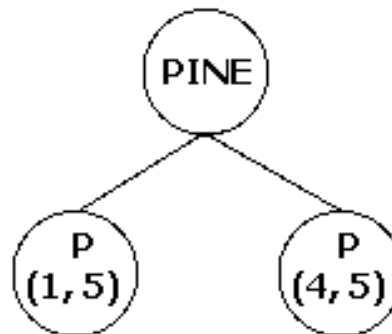
Langkah 1 dimulai dengan melihat huruf pertama pada kata yang akan dicari. Lalu, penulis mencari huruf tersebut di dalam matriks dan menjadikan tiap huruf tersebut menjadi simpul hidup tingkat pertama dalam pohon ruang solusi pencarian kata.

Sebagai contoh, kata yang akan dicari adalah kata “PINE” yang diberi *highlight* warna kuning pada daftar kata di kanan dengan huruf awal ‘P’. Lalu dicari semua huruf pada matriks.



Gambar 3. Langkah 1

Sehingga kita peroleh gambar pohon ruang solusi sebagai berikut :



Gambar 4. Ruang Solusi pada Langkah Pertama

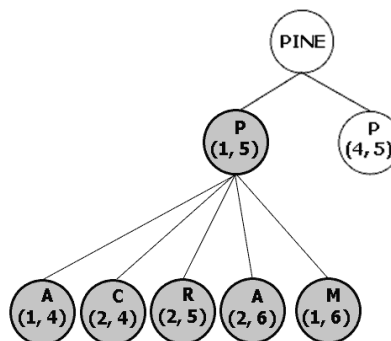
Simpul P(1,5) berarti huruf P pada posisi (1,5) pada matriks. Demikian juga halnya pada P(4,5).

#### Langkah 2

Kita mencari mulai dari simpul pertama secara mendalam dengan membangkitkan simpul anak. Karena kata – kata dalam matriks dapat mempunyai arah vertikal, horisontal, maupun diagonal, maka kita mempunyai simpul-E sebanyak delapan buah. Simpul – simpul tersebut berisi huruf yang terletak tepat berdampingan dengan huruf ‘P’ di dalam matriks, baik di atas, kanan atas, kanan bawah, bawah, kiri bawah, kiri, dan kiri atas. Karena simpul P(1,5) terletak di pingir kiri, huruf di sebelah kiri bawah, kiri, dan kiri atas tidak ada, maka simpul anak yang dapat dibangkitkan hanya 5 buah.

Lalu penulis mencocokkan masing – masing huruf pada simpul tersebut dengan huruf kedua pada kata yang akan dicari. Apabila tidak sama, simpul anak tersebut akan dimatikan dan beralih pada simpul anak lain yang masih hidup.

Karena semua simpul anak tidak ada yang memenuhi dan tidak mungkin lagi dibangkitkan simpul anak yang lain, kita melakukan runut – balik, dengan mematikan simpul P(1,5) dan beralih pada simpul P(4,5).

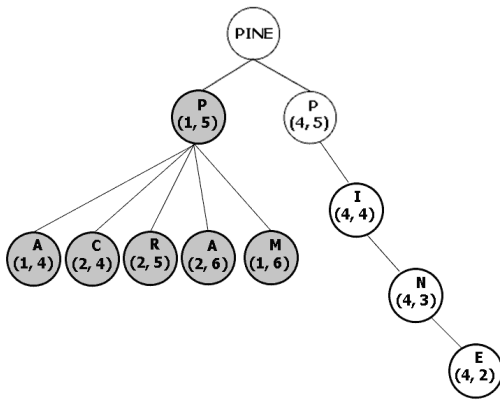


Gambar 5. Ruang Solusi pada Langkah Kedua

### Langkah 3

Lalu kita mengekspansi simpul P(4,5). Apabila ditemukan simpul anak dengan huruf yang sama dengan huruf kedua pada kata yang dicari, kita membangkitkan hanya satu simpul anak berisi huruf yang tepat berdampingan dengan arah yang konsisten. Sebagai contoh simpul I(4,4) hanya mempunyai anak simpul N(4,3) karena simpul I(4,4) berada di atas simpul induk P(4,5) maka simpul I(4,4) hanya mempunyai simpul anak yang tepat berada di atasnya. Lalu penulis mencocokkan huruf pada simpul anak dengan huruf pada letak yang bersesuaian dengan kata yang dicari.

Apabila semua huruf pada kata yang dicari terpenuhi oleh suatu lintasan simpul yang kita bangkitkan, maka kita telah menemukan solusi untuk kata tersebut.



Gambar 6. Ruang Solusi pada Langkah Ketiga



Gambar 7. Langkah Ketiga

Langkah – langkah tersebut dilakukan pada masing – masing kata pada seranai kata yang dicari sampai semua kata pada seranai ditemukan.

Permainan berakhir bila semua kata dapat ditemukan seperti tergambar pada tampilan berikut :



Gambar 8. Semua Kata dalam Daftar telah Ditemukan

## IV. KESIMPULAN

Permainan *Word Diagram* sangat tepat diselesaikan dengan algoritma runut – balik. Apabila fungsi pembatas dan fungsi pembangkit yang digunakan sesuai dengan aturan bermain yang digunakan, pencarian solusi pada permainan ini menjadi sangat mangkus.

Jadi kunci penting dalam menggunakan algoritma ini dalam pencarian solusi permainan *Word Diagram* adalah pembuatan fungsi pembangkit yang baik. Seperti pada contoh : pembangkit simpul tingkat pertama membangkitkan delapan simpul anak, namun untuk tingkat berikutnya hanya membangkitkan masing – masing satu simpul anak dengan arah yang bersesuaian pada matriks. Dengan cara itu, pencarian kata dalam matriks menjadi lebih mangkus.

## REFERENSI

[MUN07] Munir, Rinaldi, “Strategi Algoritmik”, Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, 2007.

This document was created with Win2PDF available at <http://www.daneprairie.com>.  
The unregistered version of Win2PDF is for evaluation or non-commercial use only.