

Algoritma Puzzle Pencarian Kata

Sigit Aji Nugroho (13510021)
 Program Studi Teknik Informatika
 Sekolah Teknik Elektro dan Informatika
 Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
 sigit_aji@studets.itb.ac.id

Abstract—Puzzle pencarian kata dapat dipandang sebagai sebuah graf. Huruf-huruf yang ada pada *puzzle* tersebut sebagai simpulnya. Sedangkan hubungan antara suatu huruf dengan huruf-huruf yang ada di delapan penjurua mata anginnya dianggap sebagai sisi.

Namun untuk menyelesaikan permainan ini, akan lebih mudah jika dianggap sebagai pohon merentang maupun *list of list*. List induk merupakan list dari semua huruf yang ada dalam *puzzle*. List ini dapat dibentuk dengan membuat lintasan Hamilton dari graf. Pola sederhananya dengan membuat lintasan berdasar baris dari kiri ke kanan. List anak dari suatu huruf dapat dianggap sebagai pohon yang akarnya adalah huruf tersebut.

Index Terms—Puzzle (*puzzle* pencarian kata), pencarian atau pengecekan, lintasan hamilton atau lintasan huruf, pohon penjurua, algoritma.

I. PENDAHULUAN

Puzzle pencarian kata merupakan sebuah permainan yang sering dimuat pada surat kabar. Untuk menarik minat, biasanya penerbit akan memberikan imbalan bagi pembacanya yang bisa menemukan seluruh kata yang disembunyikan di balik huruf-huruf yang disediakan. Kata-kata yang harus dicari telah disediakan. Semua kata itu telah disusun dalam *puzzle* dengan berbagai posisi: vertikal, horizontal, dan diagonal. Kata-kata itu juga bisa tertulis dari kiri ke kanan maupun kanan ke kiri, juga dari atas ke bawah maupun sebaliknya.

A	P	Q	M	D	G	K	L	H	G
Z	O	Z	Q	Y	P	L	H	J	T
X	I	D	A	S	C	X	Z	L	R
C	U	H	K	M	N	Y	U	O	F
V	Y	G	A	S	I	C	S	X	B
B	P	O	H	O	N	E	T	N	N
N	Y	E	V	N	D	R	A	A	H
M	T	H	N	L	K	S		J	G
L	R	B		J	A	E	R	M	F
K	E	X		T	U	T	F	G	A
J	W	D	N	C	B	R	G	S	S
H	Q	I	L	K	F	H	U	U	S
G	L	D	G	J	L	A	C	B	N
G	A	Y	F	R	R	C	R	S	K
F	S	G	T	T	T	D	G	G	L

(a)

L	H	G		PUZZLE KATA:		
H	J	T				
Z	L	R		POHON		
U	O	F		PENJURU		
S	X	B		LINTASAN		
T	N	N		GRAF		
A	A	H				

(b)

Gambar 1.1 Puzzle Pencarian Kata

II. DASAR TEORI

Untuk menyelesaikan permainan ini sebenarnya ada banyak cara. Setiap orang bisa menggunakan caranya masing-masing. Ada yang memilih cara sederhana asalkan selesai. Ada pula yang menggunakan cara yang rumit agar cepat selesai. Namun ada yang perlu dihindari dalam permainan ini, yaitu pencarian yang berulang-ulang untuk kondisi yang sama. Hal ini banyak terjadi jika pemain tidak melakukan pencarian secara terurut. Pemain-pemain yang melakukan pencarian dengan cara melompat-lompat bisa saja tidak ingat apakah suatu huruf pernah ia coba. Bisa saja ia mengecek di huruf yang pernah dicek sebelumnya sehingga hal ini menyebabkan ketidakmangkusan.

III. METODE

A. Gambaran Model

		1	2	3	4	
1	A	S	C	G		BOS
2	B	L	K	J		BIG
3	Z	B	G	M		
4	O	I	O	K		
5	B	E	D	S		

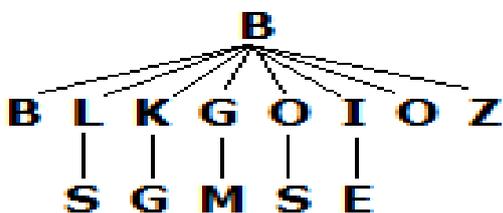
Gambar 2.1 Puzzle Pencarian Kata Sederhana

Lintasan hurufnya (list induk):

A-S-C-G-B-L-K-J-Z-B-G-M-O-I-O-K-B-E-D-S

Pola di atas bisa di dapat dengan melakukan iterasi dari baris 1 sampai y dan di dalamnya dilakukan iterasi dari kolom 1 sampai x.

Contoh pohon dari salah satu huruf:



Gambar 2.1 Pohon dengan akar huruf B (3,2)

List anak dari huruf B(3.2):

B

B-L-S

B-K-G

B-G-M

B-O-S

B-I-E

B-O

B-Z

List diambil dari sudut kiri atas dan sesuai arah jarum jam. Jika ada sudut yang terbentur tepi maka dikosongkan.

B. Pencarian Berdasarkan Huruf

Salah satu cara untuk menyelesaikan permainan ini ialah dengan menggunakan pencarian berdasarkan huruf. Setiap huruf dicek satu persatu berdasarkan lintasan Hamilton. Huruf yang dicari merupakan huruf awal dari salah satu kata yang harus ditemukan. Jika huruf itu telah ditemukan maka dicek ke delapan penjuru, yang dianggap sebagai pohon penjuru, apakah kata yang terbentuk merupakan kata yang dicari. Jika benar maka kata tersebut bisa dicoret dari daftar kata yang harus dicari dan lokasinya di *puzzle* bisa dilingkari. Jika bukan maka dicek ke penjuru selanjutnya. Jika semua penjuru telah dicek dan ternyata tidak ditemukan maka proses dilanjutkan kembali dengan mengecek kata selanjutnya apakah masih ada yang dimulai dengan huruf tersebut. Jika ada maka mengecek ke pohon penjuru lagi. Jika sampai kata terakhir tidak ditemukan maka proses dilanjutkan dengan mengecek huruf selanjutnya dari lintasan Hamilton.

Proses ini sebenarnya cukup cepat namun memiliki kelemahan jika kata yang dicari cukup banyak. Manusia memiliki ingatan yang terbatas. Terkadang pemain bisa lupa kata apa saja yang harus dicarinya. Jika satu kata saja terlupakan, kemungkinan dia sadar ketika pencarian telah sampai pada huruf terakhir lintasan Hamilton tapi ternyata masih ada kata yang belum di coret, maka ia harus

mengulanginya dari awal. Kelemahan lainnya jika pemain lupa bahwa suatu kata telah dicoret. Jika dia tetap mencari kata itu ketika huruf depannya ditemukan, maka ia telah membuang beberapa waktu.

Algoritma yang digunakan:

```
procedure PencarianHuruf (input m:
matrikshuruf, w: listkata)
{
    mencari kata-kata yang ada di list
    kata pada matriks huruf dengan metode
    pencarian berdasarkan huruf
    kondisi awal: list kata dan matriks
    huruf terdefinisi
    kondisi akhir: kata-kata ditemukan
    dalam matriks huruf}
```

Deklarasi

```
x: integer{jumlah kolom matriks huruf}
y: integer{jumlah baris matriks huruf}
n: integer{jumlah kata belum ditemukan}
c: list of character {kata yang dicari}
i,j,k: integer {untuk iterasi}
```

Algoritma

```
for i←1 to y do {iterasi baris matriks}
for j←1 to x do
{iterasi kolom matriks}
    k←1
    while (n ≠ 0 and k ≤ n) do
    {iterasi pada list kata}
        c ← w[k] {c diinisialisasi
        sebagai kata ke-j yang dicari}
        if m[j][k] = c[1] then
        {jika huruf pada matriks sama
        dengan huruf pertama suatu kata}
            (melakukan search pada
            pohon penjuru hingga penjuru habis atau
            kata ditemukan)
                if true then
                {ditemukan}
                    (kata diletakkan paling
                    belakang pada list)
                    n←n-1
                    (kata tersebut dicoret
                    dari list)
                    k←k+1
                endwhile
            endfor
        endfor
```

Berdasarkan algoritma di atas, masih ada satu lagi kelemahan yang dimiliki metode ini, yaitu iterasi pada lintasan huruf tetap dilanjutkan meski semua kata telah ditemukan ($n=0$). Dalam kondisi ini, sesungguhnya sudah tidak ada proses yang dilakukan tapi karena proses iterasi masih berjalan, maka tetapa ada waktu yang dihabiskan juga memori yang digunakan.

Untuk dunia nyata, pemain pasti akan langsung menghentikan semua pencariannya. Hal ini sebenarnya juga bisa dilakukan dalam algoritma, yaitu dengan menambahkan perintah “break” jika semua kata telah ditemukan ($n=0$). Namun perintah “break” dalam suatu iterasi tidaklah elegan. Kondisi berhenti belum tercapai tiba-tiba dipaksa keluar, maka harus dihindari.

C. Pencarian Berdasarkan Kata

Metode ini dilakukan dengan mencari kata satu persatu. Huruf awal kata dicari dalam lintasan huruf. Jika huruf yang dimaksud ditemukan maka dilakukan pengecekan pada pohon penjur. Jika ditemukan maka kata bisa langsung dicoret. Jika tidak maka harus melanjutkan pencarian ke lintasan huruf selanjutnya. Setiap kata ditemukan, pencarian dilanjutkan untuk kata selanjutnya dimulai dari awal lintasan huruf lagi.

Pencarian model ini memiliki tingkat ketelitian yang lebih tinggi bagi manusia karena pada setiap pengecekan lintasan huruf hanya perlu menghafalkan huruf depan kata yang sedang dicari. Kelemahan dari metode ini ialah jumlah pencarian secara rekursif di list huruf sebanyak jumlah kata. Namun jika kata yang dicari sering ditemukan di bagian awal list huruf, maka proses ini bisa lebih cepat.

Algoritma yang digunakan:

```

procedure PencarianHuruf (input m:
matrikshuruf, w: listkata)
{
    mencari kata-kata yang ada di list
kata pada matriks huruf dengan metode
pencarian berdasarkan kata
    kondisi awal: list kata dan matriks
huruf terdefinisi
    kondisi akhir: kata-kata ditemukan
dalam matriks huruf}
    
```

Deklarasi

```

x: integer {jumlah kolom matriks huruf}
y: integer {jumlah baris matriks huruf}
n: integer {jumlah kata belum ditemukan}
c: list of character {kata yang dicari}
i, j, k: integer {untuk iterasi}
    
```

Algoritma

```

k ← 1
while (n ≠ 0 and k ≤ n) do
    {iterasi pada list kata}
    c ← w[k] {c diinisialisasi sebagai
kata ke-j yang dicari}
    for i ← 1 to y do
        {iterasi baris matriks}
        for j ← 1 to x do
            {iterasi kolom matriks}
            if m[j][k] = c[1] then
                {jika huruf pada matriks sama
dengan huruf pertama suatu kata}
                (melakukan search pada
pohon penjur hingga penjur habis atau
kata ditemukan)
                if true then
                    {ditemukan}
                    (kata diletakkan
paling belakang pada list)
                    n ← n - 1
                    {kata tersebut dicoret
dari list}
                endifor
            endifor
        k ← k + 1
    endwhile
    
```

Perbedaan algoritma ini dibandingkan algoritma yang pertama ialah ketika kata yang dicari telah habis, maka program akan berhenti. Hal ini karena kondisi berhenti terpenuhi. Sehingga algoritma ini lebih elegan.

IV. APLIKASI PENCARIAN BERDASARKAN HURUF

Berikut ini adalah aplikasi pencarian berdasarkan huruf pada *puzzle* pencarian kata sederhana di atas.

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(a)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(b)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(c)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(d)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(e)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(f)

Keterangan:

- Iterasi pada lintasan huruf pertama
- Di dalamnya melakukan iterasi list kata, masuk kata pertama, karena huruf pertama kata tidak sama dengan huruf di lintasan maka tidak melakukan apa-apa
- Masuk kata kedua
- Karena tidak ada yang sesuai, masuk ke lintasan huruf kedua
- Kembali melakukan iterasi pada list kata
- Masuk ke kata kedua

Iterasi pada lintasan huruf terus dilanjutkan hingga ditemukan huruf yang sama dengan huruf awal salah satu kata.

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(g)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(h)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(q)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(r)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(i)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(j)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(k)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(l)

Keterangan:

- Menemukan huruf B
- Melakukan pengecekan terhadap kata pertama
- Ternyata huruf pertama kata sama dengan huruf di lintasan sehingga elakukan pengecekan pada penjurur
- Karena tidak sesuai dengan huruf kedua kata, melakukan pengecekan pada penjurur selanjutnya.
- Mengecek penjurur selanjutnya
- Mengecek penjurur selanjutnya

Keterangan:

- Sampai penjurur terakhir ternyata juga tidak ditemukan
- Maka masuk ke list kata kedua
- Kembali melakukan pengecekan terhadap penjurur
- Hingga penjurur terakhir kembali tidak ditemukan
- Proses dilanjutkan kembali ke iterasi pada lintasan huruf dan iterasi pada list kata hingga tiba ke huruf yang sama dengan huruf awal satah satu kata lagi.
- Masuk ke iterasi list kata

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(s)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(t)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(u)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(v)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(m)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(n)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(o)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(p)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(u)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(v)

Keterangan:

- Karena huruf awal kata sama dengan huruf di lintasan, melakukan pengecekan terhadap penjurur
- Pengecekan dilakukan hingga ada penjurur yang hurufnya sama dengan huruf kedua kata
- Melakukan pengecekan huruf ketiga
- Karena berbeda, pengecekan penjurur dilanjutkan

hingga penjuru terakhir

- w. Karena tetap tidak sama, melakukan pengecekan kata kedua apakah huruf pertamanya sama dengan huruf di lintasan
- x. Karena sama, melakukan pengecekan penjuru

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(y)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(z)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(aa)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(bb)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(cc)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(dd)

Keterangan:

- y. Pengecekan penjuru dilakukan hingga ditemukan yang hurufnya sama dengan huruf kedua kata
- z. Karena sama, melakukan pengecekan huruf-huruf selanjutnya
- aa. Karena kata ditemukan, matriks ditandai dan kata dicoret dari list
- bb. Karena tidak ada kata yang selanjutnya dalam list, proses dilanjutkan pada iterasi dari lintasan huruf dilanjutkan pengecekan pada list kata
- cc. Karena kata pada list tinggal satu dan huruf pada lintasan berbeda dengan huruf awal kata, iterasi lintasan dilanjutkan
- dd. Iterasi lintasan dilakukan hingga ditemukan huruf yang sama dengan huruf awal kata pada list

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(ee)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(ff)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(gg)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(hh)

Keterangan:

- ee. Melakukan pengecekan list kata
- ff. Karena huruf awal kata sama dengan huruf di lintasan, melakukan pengecekan penjuru
- gg. Karena huruf kedua kata berbeda dengan huruf di penjuru, melakukan pengecekan penjurur selanjutnya
- hh. Karena huruf kedua sama, melakukan pengecekan huruf selanjutnya

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(ii)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(jj)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(kk)

	1	2	3	4	
1	A	S	C	G	BIG
2	B	L	K	J	BOS
3	Z	B	G	M	
4	O	I	O	K	
5	B	E	D	S	

(ll)

Keterangan:

- ii. Karena kata ditemukan, matriks ditandai dan kata dicoret dari list
- jj. Karena kata dalam list sudah habis, menyelesaikan iterasi lintasan tanpa melakukan apapun.
- kk. Tiba pada huruf terakhir lintasan, selanjutnya algoritma selesai

V. KESIMPULAN

Permainan *puzzle* pencarian kata memiliki dua metode penyelesaian yang cukup sederhana. Yang pertama pencarian berdasarkan huruf. Yang kedua pencarian berdasarkan kata. Metode pencarian berdasarkan kata memiliki beberapa kelebihan dibandingkan metode yang pertama. Bagi manusia, ketelitian pada metode ini jauh lebih tinggi karena saat melakukan pencarian, yang dihafal hanyalah huruf depan kata yang dicari. Namun, metode ini mungkin lebih membosankan.

Bagi computer, metode ini bisa menyingkat lebih banyak waktu. Ketika sebuah kata telah ditemukan, proses iterasi pada lintasan huruf tidak perlu dilanjutkan. Proses bisa langsung dilanjutkan untuk kata selanjutnya. Ketika kata sudah ditemukan semua, program selesai. Hal ini berbeda dengan metode pencarian berdasarkan huruf. Meskipun semua kata telah ditemukan, proses iterasi pada lintasan huruf tetap dilanjutkan hingga huruf terakhir. Walaupun di dalam iterasi ini tidak ada proses yang dikerjakan lagi, dampaknya bisa cukup terasa jika sisa lintasan huruf masih sangat banyak.

REFERENSI

- [1] Munir, Rinaldi. *Diktat kuliah 2091 Struktur Diskrit*. Bandung: Prodi Teknik Informatika STEI ITB, 2008, hal. VIII-1 – IX-32.
- [2] www.mediatren.com/puzzle-pencarian-kata
waktu: Minggu, 11 Desember 2011 jam 13.35 WIB
- [3] beta.mediatren.com/software/cell.../912-puzzle-pencarian-kata.html
waktu: Minggu, 11 Desember 2011 jam 13.43 WIB

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, 12 Desember 2011

ttd

Nama dan NIM