

Penerapan dan Perbandingan Teknik Pencarian Melebar dan Mendalam Pada Permainan “Wordament”

Muhammad Nizami (13512501)
Program Sarjana Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13512501@std.stei.itb.ac.id

Abstrak—Wordament adalah sebuah permainan teka-teki kata. Cara memainkannya adalah dengan membuat kata dari suatu elemen matriks karakter, dilanjutkan dengan tetangga Moore-nya, terus hingga suatu karakter terakhir, dan diperiksa ke dalam kamus. Permainan ini dapat dimodelkan menjadi persoalan penelusuran graf. Salah satu solusi untuk persoalan ini adalah dengan pencarian melebar dan mendalam. Pencarian melebar dan pencarian mendalam memberikan seluruh kata yang dapat dibentuk dari matriks kata. Apabila batasan waktu diperhitungkan, terlihat perbedaan antara pencarian melebar dan pencarian mendalam.

Kata Kunci—BFS, DFS, Wordament, graf.

I. PENDAHULUAN

Wordament adalah sebuah permainan teka-teki kata yang dipublikasi oleh Microsoft Studios, dirilis pertama kali untuk Windows Phone pada 24 April 2012.[1]

Permainan ini mengandung sebuah matriks 4x4 atas alfabet Bahasa Inggris. Pemain memainkan permainan ini dengan membuat kata yang dimulai dari salah satu elemen matriks, dilanjutkan dengan tetangga Moore-nya. Tetangga Moore adalah tetangga 8-titik, yaitu tetangga kiri, kanan, atas, bawah, diagonal kiri-atas, diagonal kanan-atas, diagonal kiri-bawah, diagonal kanan-bawah. Pembentukan kata dapat dilanjutkan kepada tetangga Moore selanjutnya, hingga suatu elemen yang dinyatakan sebagai akhir dari kata tersebut. Panjang kata bervariasi.

Kata tersebut diperiksa ke dalam kamus. Apabila kata tersebut ada di dalam kamus, kata tersebut diterima dan dihitung skornya. Apabila kata

tersebut tidak ada di dalam kamus, kata tersebut ditolak.

Setiap kata yang dibentuk dan diterima, akan dimasukkan ke dalam himpunan kata yang telah dibentuk. Satu kata hanya diperhitungkan satu kali. Nilai setiap kata diberikan sebagai berikut:

skor =

total poin seluruh huruf pada kata tersebut
× panjang kata + bonus kata panjang

(1)

Setiap huruf memiliki poin yang berbeda-beda seperti pada tabel berikut :

Huruf	Poin
E	1
A,I,N,O,R,S,T	2
C,D,L	3
G,H,M,P,U	4
B,F,Y	5
K,V,W	6
Z	8
X	9
J,Q	10

Bonus kata panjang diberikan khusus untuk kata yang lebih atau sama dengan 10 huruf.[2]

Permainan ini, biasanya dilakukan dalam suatu batasan waktu. Pemain dapat membuat kata sebanyak-banyaknya dalam batasan waktu tersebut.

II. TEKNIK PENCARIAN MENDALAM DAN MELEBAR

Algoritma traversal graf adalah suatu algoritma mengunjungi simpul dengan cara yang sistematis. Dalam hal ini, graf dapat dikatakan sebagai representasi persoalan.

Algoritma pencarian graf menyeluruh, bila diasumsikan graf terhubung, di antaranya pencarian melebar dan pencarian mendalam. Pencarian melebar dan pencarian mendalam ini juga merupakan contoh pencarian tanpa informasi.

Dalam proses pencarian solusi, dapat digunakan pendekatan graf statis dan graf dinamis. Graf statis telah terbentuk sebelum pencarian, dan graf dinamis dibentuk selama pencarian. Properti graf dinamis di antaranya simpul awal, simpul, cabang, ruang status, dan ruang solusi. Pembangkitan status dilakukan dengan cara melakukan langkah legal pada status induknya.

Pencarian melebar adalah pencarian yang mendahulukan simpul-simpul pada aras yang sama terlebih dahulu sebelum melanjutkan kepada aras berikutnya. Pencarian mendalam adalah pencarian yang menelusuri simpul dengan aras lebih dalam terlebih dahulu, dan kembali ke aras sebelumnya saat telah mencapai daun.

Pencarian melebar dapat diimplementasi dengan memasukkan dan menelusuri simpul dalam sebuah antrian, sementara pencarian mendalam dapat diimplementasi dengan tumpukan.

III. PENENTUAN PROPERTI POHON DINAMIS DAN ALGORITMA PENCARIAN MENDALAM DAN PENCARIAN MELEBAR UNTUK PERMAINAN WORDAMENT

Pada persoalan permainan *Wordament* pada dasarnya tidak diberikan sebuah graf. Karenanya, perlu dibentuk graf selama pencarian, maka graf yang dibentuk haruslah graf dinamis.

Pada persoalan ini, pada permainan *Wordament*, pohon dinamis yang dibentuk memiliki properti sebagai berikut:

1. Simpul awal: kata dengan panjang nol
2. Simpul: sebuah string atas pasangan terurut $\langle i, j \rangle$ yang menyatakan posisi elemen matriks huruf pembentuk kata
3. Cabang: pada aras 1, dibentuk simpul berupa salah satu pasangan terurut $\langle i, j \rangle$ dengan $i, j \in \{1, 2, 3, 4\}$. Pada aras berikutnya, cabang berupa posisi tetangga Moore dari simpul induknya.
4. Ruang status: himpunan seluruh string atas pasangan terurut $\langle i, j \rangle$
5. Ruang solusi: himpunan seluruh string atas pasangan terurut $\langle i, j \rangle$ sedemikian hingga string atas kata yang ditunjuk pasangan terurut tersebut ada dalam kamus kata

Dari properti pohon ruang status yang didefinisikan di atas, dapat dibentuk algoritma pencarian mendalam dan melebar untuk permainan ini.

Berikut ini adalah bentuk umum algoritma pencarian melebar yang digunakan pada percobaan

kali ini:

1. Bentuk simpul akar, masukkan pada antrian
2. Selama antrian tidak kosong, lakukan:
 - a. Ambil satu simpul dari antrian
 - b. Periksa apakah simpul merupakan solusi. Bila merupakan solusi, masukkan kepada himpunan kata yang terbentuk
 - c. Buat simpul anak, masukkan ke antrian.

Berikut ini adalah bentuk umum algoritma pencarian mendalam yang digunakan pada percobaan kali ini:

1. Bentuk simpul akar, masukkan pada tumpukan
2. Selama tumpukan tidak kosong, lakukan:
 - a. Ambil satu simpul dari tumpukan
 - b. Periksa apakah simpul merupakan solusi. Bila merupakan solusi, masukkan kepada himpunan kata yang terbentuk
 - c. Buat simpul anak, masukkan ke tumpukan.

Secara khusus, algoritma pencarian kata pada permainan *wordament*, secara melebar adalah sebagai berikut:

1. Untuk setiap pasangan terurut $p = \langle i, j \rangle$ yang menandakan posisi di dalam matriks, buat suatu senarai $\langle p \rangle$ berukuran 1 dan masukkan ke dalam antrian
2. Selama antrian tidak kosong, lakukan:
 - a. Ambil satu senarai l dari antrian
 - b. Ambil elemen pertama dari senarai, mulai pembuatan kata dari posisi tersebut.
 - c. Lanjutkan pembuatan kata ke tetangga yang posisi relatifnya ditunjukkan oleh elemen list berikutnya, hingga elemen terakhir.
 - d. Untuk setiap tetangga t , jika titik yang ditunjukkan urutan posisi dan gerakan $\langle p_1, p_2, \dots, p_i \rangle.t$ menghasilkan posisi yang belum dikunjungi dalam iterasi ini, masukkan $\langle p_1, p_2, \dots, p_i \rangle.t$ ke dalam antrian.
 - e. Tentukan apakah kata yang dibentuk terdapat dalam kamus.

Algoritma pencarian kata pada permainan *wordament*, secara mendalam adalah sebagai berikut:

1. Untuk setiap pasangan terurut $p = \langle i, j \rangle$ yang

menandakan posisi di dalam matriks, buat suatu senarai $\langle p \rangle$ berukuran 1 dan masukkan ke dalam tumpukan

2. Selama antrian tidak kosong, lakukan:
 - a. Ambil satu senarai l dari tumpukan
 - b. Ambil elemen pertama dari senarai, mulai pembuatan kata dari posisi tersebut.
 - c. Lanjutkan pembuatan kata ke tetangga yang posisi relatifnya ditunjukkan oleh elemen list berikutnya, hingga elemen terakhir.
 - d. Untuk setiap tetangga t , jika titik yang ditunjukkan urutan-urutan posisi dan gerakan $\langle p_1, p_2, \dots, p_l \rangle.t$ menghasilkan posisi yang belum dikunjungi dalam iterasi ini, masukkan $\langle p_1, p_2, \dots, p_l \rangle.t$ ke dalam antrian.
 - e. Tentukan apakah kata yang dibentuk terdapat dalam kamus.

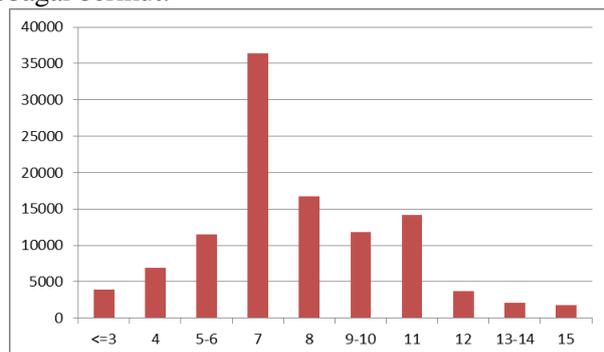
IV. ANALISIS PENGGUNAAN TEKNIK PENCARIAN MENDALAM DAN MELEBAR UNTUK MEMBENTUK SOLUSI

Analisis dilakukan dengan menguji coba teknik pencarian mendalam dan melebar untuk membentuk solusi pada instansiasi persoalan dengan matriks karakter sebagai berikut:

H H E L
A C A B
N R O T
F I T I

Pengujian dilakukan dengan program yang dibangun di atas bahasa Java, dengan kakas Netbeans. Pengujian dilakukan pada komputer personal dengan spesifikasi prosesor AMD A10-5745M APU 2.10 GHz dengan RAM 4.00 GB (3.20 GB dapat digunakan)

Kamus yang digunakan adalah subset kamus Bahasa Inggris dengan distribusi panjang kata sebagai berikut:



Hasil yang didapatkan dengan algoritma pencarian melebar, terurut sesuai urutan penemuan, adalah sebagai berikut:

selesai. skor: 4113 Banyak Percobaan: 6031636

Kata	Skor	Percobaan ke-
HAH	30	68
HAT	24	93
EAR	15	105
EAT	15	106
ARF	27	133
ART	18	134
ARC	21	137
ACE	18	145
AHA	24	149
COT	21	163
COB	30	169
CAN	21	170
CAR	21	171
CAT	21	176
CAB	30	177
CAL	24	180
ABT	27	216
ABO	27	217
ALB	30	228
ALE	18	229
NIT	18	233
ROT	18	268
ROC	21	273
ROB	27	274
RAN	18	282
RAH	24	284
RAT	18	287
ORT	18	305
ORC	21	307
OAR	18	314
OAT	18	315
HART	40	338
HALE	40	488
HEAR	36	499
HEAT	36	500
HEAL	40	503
EARN	28	526
EACH	40	538
ARCO	36	664
ARCH	44	666
ARAB	44	672
ACHE	40	701
CRAB	48	758
COIF	48	781
COIR	36	783
COIN	36	784
CORN	36	790
COAT	36	798
COAL	40	802
CART	36	812

CHAR	44	849	ORCH	44	248571
CHAT	44	853	ORCA	36	250302
ABLE	44	992	ORT	18	259144
NARC	36	1092	RAH	24	523440
RIOT	32	1146	RACE	32	544089
ROBE	40	1213	RAT	18	549436
RANI	32	1243	RATIO	50	554623
RACE	32	1250	RAN	18	614018
ORCA	36	1334	RANCH	65	615330
ORCH	44	1336	RANI	32	619968
ORAL	36	1348	ROB	27	682446
HANCE	60	1451	ROBE	40	682447
HEART	55	2122	ROBLE	65	684179
ACORN	55	2866	ROC	21	689393
ACORN	55	2872	ROACH	65	697972
COTTA	55	3212	ROT	18	699397
CORAL	60	3250	RIOT	32	964863
COATI	55	3268	NARC	36	1279631
CAROB	70	3323	NARCO	55	1284781
CARAT	55	3325	NIT	18	1480974
CABOT	70	3400	NITRO	50	1504702
CABLE	70	3405	ALE	18	1769450
CHART	65	3430	ALB	30	1814764
CHEAT	60	3462	AHA	24	1902184
CHELA	65	3470	ABLE	44	2017558
ABORT	65	3888	ABO	27	2034900
NITRO	50	4093	ABORT	65	2039568
NARCO	55	4287	ABT	27	2046041
ROACH	65	4671	ACE	18	2068225
ROBLE	65	4717	ACHE	40	2085219
RANCH	65	4812	ACORN	55	2101290
RATIO	50	4881	ACORN	55	2102859
OBEAH	70	5428	ARC	21	2195886
ARABLE	90	10289	ARCH	44	2196773
COITAL	84	11658	ARCO	36	2199682
CORTIN	78	11669	ART	18	2213196
CARINA	78	12023	ARF	27	2220616
ACROBAT	126	33718	CHAR	44	2418899
CAROTIN	105	38073	CHART	65	2423319
CHARIOT	119	38891	CHARIOT	119	2427147
OCARINA	105	55982	CHELA	65	2446873
CHELATOR	152	112763	CHELATOR	152	2448681
BUILD SUCCESSFUL (total time: 26			CHEAT	60	2466146
minutes 17 seconds)			CHAT	44	2494003
Hasil yang didapatkan dengan algoritma			CAL	24	2514882
pencarian mendalam, terurut sesuai urutan			CAB	30	2539643
penemuan, adalah sebagai berikut:			CABLE	70	2541104
selesai. skor: 4113			CABOT	70	2542563
Banyak Percobaan: 6031636			CAT	21	2546047
Kata	Skor	Percobaan ke-	CAR	21	2552994
OBEAH	70	13818	CAROB	70	2553882
OCARINA	105	100154	CAROTIN	105	2554497
OAT	18	155007	CART	36	2555433
OAR	18	162775	CARINA	78	2557455
ORAL	36	235790	CARAT	55	2598580
ORC	21	247738	CAN	21	2608826

COB	30	2631010
COAL	40	2638431
COAT	36	2642226
COATI	55	2642827
COT	21	2643778
CORAL	60	2650394
CORN	36	2651945
CORTIN	78	2653403
COITAL	84	2655743
COIN	36	2664510
COIR	36	2666493
COIF	48	2669695
COTTA	55	2671931
CRAB	48	2753470
ACROBAT	126	3248738
ARAB	44	3263950
ARABLE	90	3264324
EACH	40	4237506
EAT	15	4248646
EAR	15	4263971
EARN	28	4267337
HEAL	40	4417606
HEAT	36	4437915
HEAR	36	4444349
HEART	55	4446399
HALE	40	4646067
HAT	24	4716914
HART	40	4738793
HAH	30	4754264
HANCE	60	4866156

BUILD SUCCESSFUL (total time: 27 minutes 16 seconds)

Secara teoretis, dengan lebar maksimal b dan kedalaman maksimal m , pencarian mendalam dan melebar kepada seluruh pohon memiliki kompleksitas waktu yang sama, $O(b^m)$. Kompleksitas ruang pencarian mendalam adalah $O(bm)$ dan kompleksitas ruang pencarian melebar adalah $O(b^m)$.

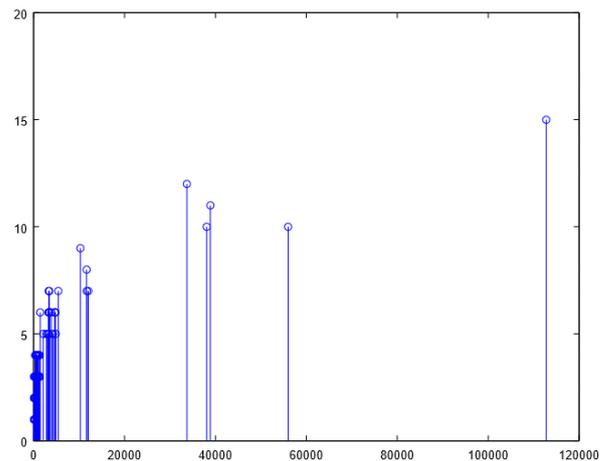
Hal ini juga terlihat dalam waktu eksekusi dari program eksperimen.

Dikarenakan pencarian dilakukan secara menyeluruh, kedua teknik pencarian mengeluarkan solusi yang sama, namun dengan urutan penemuan yang berbeda.

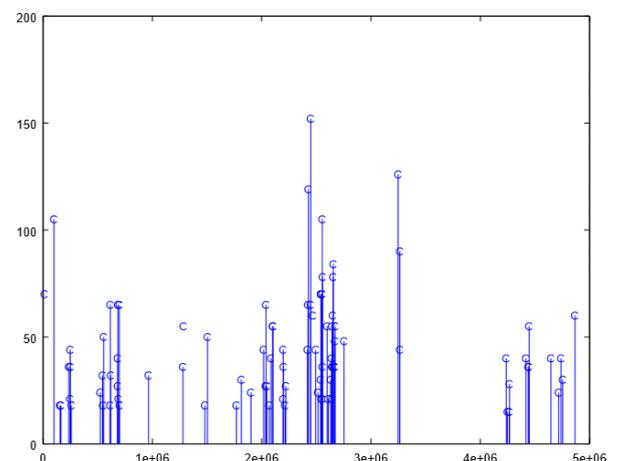
Permainan *Wordament* dibatasi waktu, oleh karena itu perlu dilihat seberapa cepat pencarian mendapatkan hasil.

Pencarian melebar mendapatkan solusi pertama pada percobaan ke-68 dan solusi terakhir pada percobaan ke- 112763. Pencarian mendalam mendapatkan solusi pertama pada percobaan ke-13818 dan solusi terakhir pada percobaan ke-4866156.

Berikut ini adalah *stem plot* skor kata yang ditemukan terhadap jumlah percobaan pada pencarian melebar.



Berikut ini adalah *stem plot* skor kata yang ditemukan terhadap jumlah percobaan pada pencarian mendalam.



Terlihat bahwa sebagian besar penemuan pencarian melebar terjadi pada awal pencarian, dan penemuan pada pencarian mendalam lebih merata dari awal hingga akhir pencarian.

Terlihat pula bahwa skor kata yang ditemukan oleh pencarian melebar cenderung naik, sementara skor kata yang ditemukan oleh pencarian mendalam lebih merata. Hal ini karena skor kata terkait dengan panjang kata, dan pencarian melebar mendahulukan aras rendah yang berkorespondensi dengan kata yang pendek, sementara pencarian mendalam melakukan penelusuran hingga aras terdalam terlebih dahulu, yang berkorespondensi dengan kata yang panjang.

Hal tersebut dapat dijelaskan dengan melihat perilaku permutasi karakter alfabet Bahasa Inggris dan distribusi kata dalam kamus. Semakin banyak panjang karakter, banyaknya permutasi semakin

besar. Untuk panjang karakter n , banyaknya permutasi adalah 26^n . Sementara, banyaknya kata dalam kamus terhadap panjang karakter memiliki puncak pada $n=7$, dan turun untuk $n>7$.

V. SIMPULAN

Pada permainan *Wordament*, algoritma pencarian melebar dan mendalam memberikan hasil yang sama apabila dilakukan kepada keseluruhan pohon. Pencarian melebar lebih dahulu memberikan hasil daripada pencarian mendalam.

DAFTAR PUSTAKA

- [1] http://old.seattletimes.com/html/technologybrierdudleysblog/2017789418_bogging_success_of_microsofts.html ,4 April 2015
- [2] Tim Asisten dan Pengajar IF2110. *Tugas Besar – IF2110/Algoritma dan Struktur Data*. Bandung: STEI-ITB, 2014.

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, 5 Mei 2015



Muhammad Nizami
13512501