

Pembandingan Kompleksitas Algoritma Pada Penyelesaian Permasalahan Graph Isomorphism

Ryan Nathan Soetandyo, Arya Yudhi Wijaya dan Rully Soelaiman
 Jurusan Teknik Informatika, Fakultas Masing-masing, Institut Teknologi Sepuluh Nopember (ITS)
 Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia
e-mail: rully@if.its.ac.id

Abstrak—Permasalahan graf isomorphism adalah permasalahan untuk menentukan apakah 2 buah graf bersifat isomorphic. Kedua graf bisa dikatakan isomorphic jika simpul pada graf 1 bijeksi terhadap simpul pada graf 2. Permasalahan “ISOMORPH” yang terdapat pada situs penilaian daring SPOJ akan digunakan sebagai masalah yang merepresentasikan permasalahan graf isomorphism. 2 macam algoritma akan digunakan untuk menyelesaikan permasalahan tersebut. 2 algoritma tersebut adalah algoritma VF2 dan algoritma Schmidt & Druffel. Hasil uji coba menunjukkan bahwa permasalahan “ISOMORPH” bisa diselesaikan dengan waktu penyelesaian tercepat sebesar 0.0 detik dengan menggunakan memori sebesar 2.7 MB.

Kata Kunci— Graf, Isomorphism, Schmidt & Drufel , VF2

I. PENDAHULUAN

Graf adalah sebuah model yang direpresentasikan sebagai kumpulan titik atau simpul dan beberapa grais yang menghubungkan antar titik atau yang disebut sebagai edge Graf bisa digunakan sebagai model berbagai macam relasi dalam berbagai macam bidang seperti fisika, biologi, dan teknologi informasi. Salah satu masalah yang muncul di graf adalah masalah isomorphism.

Graf A dan graf B bisa dikatakan isomorphic jika semua simpul di graf A bisa dipetakan ke simpul di graf B secara bijeksi. Untuk bisa mengetahui apakah kedua graf bersifat isomorphic ada beberapa pilihan algoritma yang bisa digunakan seperti VF2, Schmidt Druffel *fast backtracking*, dan lain lain.

Pada tugas akhir ini, akan diselesaikan permasalahan dengan judul “ISOMORPH” pada situ penilaian daring SPOJ. Pada permasalahan tersebut akan terdapat beberap graf yang harus dicari pasangan isomorphic-nya. Permasalahn tersebut akan diselesaikan dengan menggunakan 2 macam algoritma yaitu algoritma VF2 dan algoritma Schmidt & Druffel *fast backtracking*.

II. TINJUAN PUSTAKA

A. Algoritma VF2

Algoritma VF2 adalah salah satu algoritma yang bisa digunakan untuk menyelesaikan permasalahan isomorphism. Algoritma ini bisa digunakan untuk graf *directed* dan graf *undirected*.

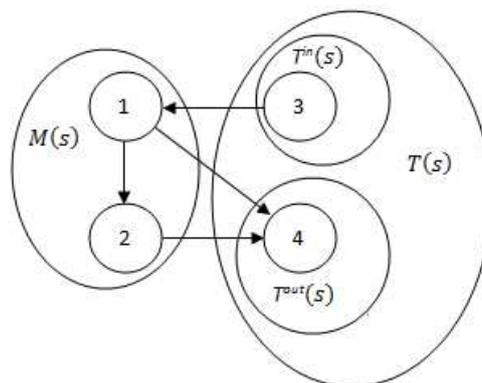
Proses pencocokan antara 2 graf $G1 = (V1, E1)$ dengan $G2 = (V2, E2)$ tersediir dari penentuan M pemetaan simpul $G1$ yang berasosiasi dengan simpul $G2$.

M di ekspresikan sebagai set dari susunan pasangan (n,m) (dengan $n \in V1$ dan $m \in V2$), dimana setiap pasangan tersebut merepresentasikan pemetaan antara simpul n dari $V1$ dengan m dari $V2$.

State Space Representation (selanjutnya disebut SSR) bisa digunakan untuk mendeskripsikan proses pembandingan graf, jika setiap keadaan s dari proses pembandingan merepresentasikan sebagian dari solusi akhir. Sebuah bagian dari solusi pemetaan (s) adalah *subset* dari M .

Pada prinsipnya, solusi dari masalah pembandingan adalah dengan mencoba semua kemungkinan pemetaan. Untuk kemungkinan pemetaan di setiap s dari s_0 menuju solusi akhir terdapat 3 kriteria yang digunakan untuk mendeteksi adanya kemungkinan kondisi pada saat ini tidak menghasilkan pemetaan solusi. 3 kriteria tersebut adalah *0-look-ahead rules*, *1-look-ahead rules* dan *2-look-ahead rules*, yang akan dijelaskan pada Tabel 1.

1-look-ahead rules dan *2-look-ahead rules* digunakan untuk mengurangi jumlah kemungkinan pemetaan. Kedua *rule* tersebut menggunakan kumpulan simpul yang bisa mengunjungi simpul-simpul di $M_1(s)$ dan $M_2(s)$ yang dilambangkan dengan $Tin1(s)$ dan $Tin2(s)$, serta menggunakan kumpulan simpulan yang bisa dikunjungi oleh simpul-simpul di $M_1(s)$ dan $M_2(s)$ yang dilambangkan dengan $Tout1(s)$ dan $Tout2(s)$, dan $T_1 s = Tin1 s \cup Tout1(s)$, $T_2 s = Tin2 s \cup Tout2(s)$. Gambar 1 mengilustrasikan $T(s)$ dan $M(s)$.



Gambar 1 : Ilustrasi pencarian $T(s)$ dan $M(s)$

TABEL 1. LOOK AHEAD RULES

Look Ahead	Rule	Kondisi
0	R_pred	Jika dan hanya jika setiap predecessor simpul n' dari n di pemetaan sementara berkorespondensi dengan predecessor simpul m' dari m dan sebaliknya.
	R_succ	Jika dan hanya jika setiap successor simpul n' dari n di pemetaan sementara berkorespondensi dengan successor simpul m' dari m dan sebaliknya.
1	R_termin	Jika dan hanya jika jumlah predecessors dan successors dari n yang merupakan bagian dari $Tin1(s)$ sama dengan jumlah predecessors dan successors dari m yang merupakan bagian dari $Tin2(s)$
	R_termout	Jika dan hanya jika jumlah predecessors dan successors dari n yang merupakan bagian dari $Tout1(s)$ sama dengan jumlah predecessors dan successors dari m yang merupakan bagian dari $Tout2(s)$
2	R_new	Jika dan hanya jika jumlah predecessors dan successors dari n yang bukan merupakan bagian dari $M_1(s)$ maupun $T_1(s)$ sama dengan jumlah predecessors dan successors dari m yang bukan merupakan bagian dari $M_2(s)$ maupun $T_2(s)$

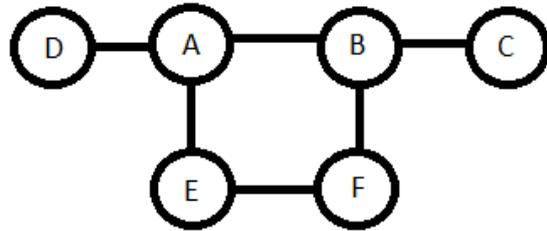
B. Schmidt & Druffel Fast Backtracking Algorithm

Douglas & Schmidt fast backtracking algorithm menggunakan distance matrix untuk menyelesaikan masalah graf isomorphism. Distance matrix adalah karakteristik dari sebuah graf yang memberikan informasi tentang hubungan antar simpul di dalam graf tersebut. Distance matrix D adalah $N \times N$ matrix dengan setiap element d_{ij} merepresentasikan panjang dari jarak terdekat antara simpul V_i dan V_j . Jika $i=j$ maka $d_{ij} = 0$, jika tidak ada path antara 2 simpul V_i dan V_j , panjangnya didefinisikan tak hingga.

Definisikan matriks row characteristic sebagai XR dengan ukuran $(N-1)$, setiap elemen x_{im} adalah jumlah simpul yang memiliki jarak m dari simpul V_i . Definisikan XC sebagai matriks column characteristic dengan ukuran $(N-1)$, setiap elemen x_{im} adalah jumlah simpul yang bisa memiliki jarak terpendek m dari simpul i . Matriks characteristic X dibentuk dengan menggabungkan XR dan XC . Tabel 2(a), Tabel 2(b), Tabel 2(c), Tabel 2(d) secara berturut-turut menunjukkan matriks distance, XR , XC dan X dari graf pada Gambar 2. Tabel 2(e), Tabel 2(f), Tabel 2(g), Tabel 2(h) secara berturut-turut menunjukkan

matriks distance, XR , XC dan X dari graf pada Gambar 3.

Gambar 2 dan Gambar 3 adalah graf undirected yang akan digunakan sebagai ilustrasi langkah-langkah algoritma Schmidt dan Druffel Fast Backtracking. Karena graf bersifat undirected maka row characteristic matrix akan sama nilainya dengan column characteristic matrix, hal ini bisa dilihat pada Tabel 2(b) dengan Tabel 2(c), serta Tabel 2(d) dengan Tabel 2(e).



Gambar 2. Ilustrasi Graf 1

TABEL 2. (A) DISTANCE MATRIX GRAF 1, (B) ROW CHARACTERISTIC MATRIX GRAF 1, (C) COLUMN CHARACTERISTIC MATRIX GRAF 1, (D) CHARACTERISTIC MATRIX GRAF 1, (E) DISTANCE MATRIX GRAF 2, (F) ROW CHARACTERISTIC MATRIX GRAF 2, (G) COLUMN CHARACTERISTIC MATRIX GRAF 2, (H) CHARACTERISTIC MATRIX GRAF 2.

(a)							(b)					
	A	B	C	D	E	F		1	2	3	4	5
A	0	1	2	1	1	2	A	3	2	0	0	0
B	1	0	1	2	2	1	B	3	2	0	0	0
C	2	1	0	3	3	2	C	1	2	2	0	0
D	1	2	3	0	2	3	D	1	2	2	0	0
E	1	2	3	2	0	1	E	2	2	1	0	0
F	2	1	2	3	1	0	F	2	2	1	0	0

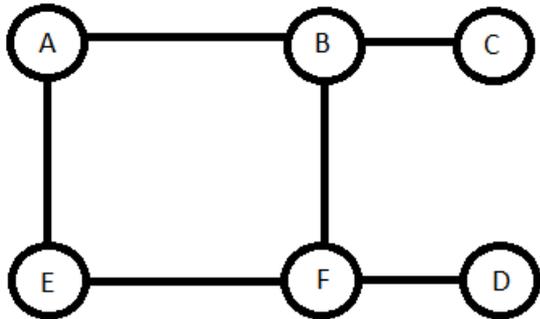
(c)						(d)					
	1	2	3	4	5		1	2	3	4	5
A	3	2	0	0	0	A	3	2	0	0	0
B	3	2	0	0	0	B	3	2	0	0	0
C	1	2	2	0	0	C	1	2	2	0	0
D	1	2	2	0	0	D	1	2	2	0	0
E	2	2	1	0	0	E	2	2	1	0	0
F	2	2	1	0	0	F	2	2	1	0	0

(e)							(f)					
	A	B	C	D	E	F		1	2	3	4	5
A	0	1	2	3	1	2	A	2	2	1	0	0
B	1	0	1	2	2	1	B	3	2	0	0	0
C	2	1	0	3	3	2	C	1	2	2	0	0
D	3	2	3	0	2	1	D	1	2	2	0	0
E	1	2	3	2	0	1	E	2	2	1	0	0
F	2	1	2	1	1	0	F	3	2	0	0	0

(g)						(h)					
	1	2	3	4	5		1	2	3	4	5
A	2	2	1	0	0	A	2	2	1	0	0
B	3	2	0	0	0	B	3	2	0	0	0
C	1	2	2	0	0						
D	1	2	2	0	0						

E	2	2	1	0	0
F	3	2	0	0	0

C	1	2	2	0	0
D	1	2	2	0	0
E	2	2	1	0	0
F	3	2	0	0	0



Gambar 3. Ilustrasi Graf 2.

III. UJI COBA DAN ANALISIS

A. Uji Coba Kebenaran Algoritma VF2

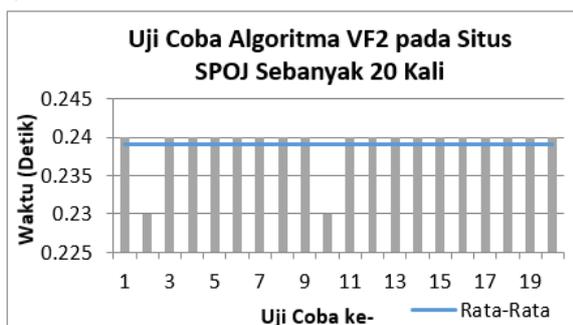
Uji kebenaran dilakukan dengan membandingkan hasil keluaran dari program dengan permasalahan dan mengirimkan kode sumber program dari implementasi algoritma VF2 yang telah dibuat, pada persoalan yang bersangkutan yaitu SPOJ “ISOMORPH”.



Gambar 4. Hasil Uji Coba Algoritma VF2 pada situs SPOJ

Kode sumber program mendapatkan umpan balik *Accepted* dari situs SPOJ. Hal ini berarti keluaran kode sumber telah sesuai dengan persoalan yang terdapat pada situs SPOJ. Waktu tercepat dari pengumpulan kode sumber program adalah 0,23 detik dengan memori maksimal yang dibutuhkan program sebesar 2,8MB.

Dilakukan pengujian sebanyak 20 kali dalam situs SPOJ untuk menemukan rata-rata waktu program, serta memori yang dibutuhkan. Dari pengujian sebanyak 20 kali, waktu yang dibutuhkan program minimum 0,23 detik, maksimum 0,24 detik dan rata-rata 0,239 detik. Memori yang dibutuhkan program minimum sebesar 2,7MB dan maksimum sebesar 2,8MB.



Gambar 5. Grafik Hasil Uji Coba Algoritma VF2 pada situs SPOJ

B. Uji Coba Kebenaran Algoritma Schmidt & Druffel

Uji kebenaran dilakukan dengan membandingkan hasil keluaran dari program dengan permasalahan dan mengirimkan kode sumber program dari implementasi algoritma Schmidt & Druffel yang telah dibuat, pada persoalan yang bersangkutan yaitu SPOJ “ISOMORPH”.



Gambar 6. Hasil Uji Coba Algoritma Schmidt & Druffel pada situs SPOJ

Kode sumber program mendapatkan umpan balik *Accepted* dari situs SPOJ. Hal ini berarti keluaran kode sumber telah sesuai dengan persoalan yang terdapat pada situs SPOJ. Waktu tercepat dari pengumpulan kode sumber program adalah 0,00 detik dengan memori maksimal yang dibutuhkan program sebesar 2,8MB.

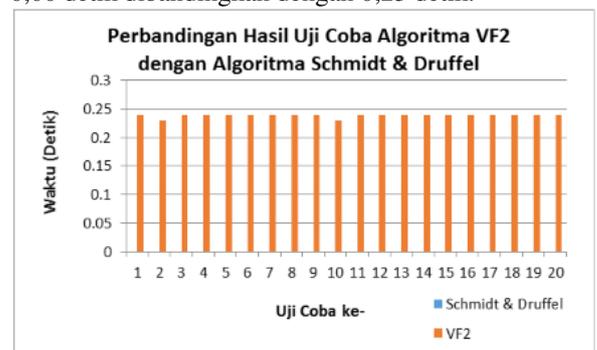
Dilakukan pengujian sebanyak 20 kali dalam situs SPOJ untuk menemukan rata-rata waktu program, serta memori yang dibutuhkan. Dari pengujian sebanyak 20 kali, waktu yang dibutuhkan program konstan 0,00 detik. Memori yang dibutuhkan program minimum sebesar 2,7MB dan maksimum sebesar 2,8MB.



Gambar 7. Grafik Hasil Uji Coba Algoritma Schmidt & Druffel pada situs SPOJ

C. Analisa Hasil Uji Coba

Hasil uji coba menunjukkan algoritma Schmidt & Druffel bisa menyelesaikan permasalahan “ISOMORPH” lebih cepat dibandingkan dengan algoritma VF2 dengan waktu penyelesaian sebesar 0,00 detik dibandingkan dengan 0,23 detik.



Gambar 8. Grafik Perbandingan Hasil Uji Coba Algoritma VF2 dengan Algoritma Schmidt & Druffel

Perbedaan waktu penyelesaian permasalahan “ISOMORPH” disebabkan oleh perbedaan kompleksitas waktu dari kedua algoritma. Algoritma VF2 memiliki kompleksitas waktu sebesar $O(N!N^2)$

sedangkan algoritma Schmidt & Druffel memiliki kompleksitas waktu yang lebih baik yaitu sebesar $O(KN^3)$

IV. KESIMPULAN

Dari hasil uji coba yang telah dilakukan terhadap implementasi penyelesaian permasalahan "ISOMORPH" dapat diambil kesimpulan sebagai berikut:

1. Implementasi algoritma VF2 dan Schmidt & Druffel dapat menyelesaikan permasalahan "ISOMORPH"
2. Algoritma Schmidt & Druffel memiliki kompleksitas $O(KN^3)$ yang lebih cepat dibandingkan dengan algoritma VF2 $O(N!N^2)$.
3. Algoritma VF2 bisa menyelesaikan permasalahan "ISOMORPH" dengan waktu rata-rata 0.239 detik dengan memori maksimum sebesar 2.8MB, dan algoritma Schmidt & Druffel bisa menyelesaikan permasalahan dengan waktu konstan 0.0 detik dengan memori maksimum sebesar 2.8MB.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih Tuhan Yang Maha Esa atas segala karunia dan rahmat-Nya yang telah diberikan selama ini.

Penulis juga mengucapkan terima kasih kepada orang tua, saudara serta keluarga penulis yang tiada henti-hentinya memberikan semangat, Bapak Arya Yudhi Wijaya dan Bapak Rully Soelaiman selaku dosen pembimbing penulis serta seluruh pihak yang tidak bisa penulis sebutkan satu persatu yang telah memberikan dukungan selama penulis mengerjakan penelitian ini.

DAFTAR PUSTAKA

- [1] Douglas C. Schmidt, Larry E. Druffel. "A Fast Backtracking Algorithm to Test Directed Graphs for Isomorphism Using Distance Matrices." *Journal of the Association for Computing Machinery* 23, no. 3 (July 1976): 433 - 445.
- [2] Jonatan L. Gross, Jay Yellen, Ping Zhang. *Handbook Of Graph Theory Second Edition*. Taylor & Francis Group, 2014.
- [3] L.P.Cordella, P.Foggia, C.Sansone, M.Vento. "Performance Evaluation of the VF Graph Matching Algorithm." *IEEE*, 1999: 1172-1177.
- [4] Luigi P. Cordella, Pasquale Foggia, Carlo Sansone, Mario Vento. "A (Sub)Graph Isomorphism Algorithm for Matching Large Graphs." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, no. 10 (Oktober 2004): 1367-1372.