Jurnal As-Salam, 1(1) Januari - April 2017
(Print ISSN 2528-1402, Online ISSN 2549-5593)

Optimisasi Kombinatorial dengan 2 *Join*

Mimi Sari Syahputri[1], Saib Suwilo[2], Mardiningsih[3]

[1]Mahasiswa Pasca Sarjana Matematika FMIPA USU
Email: Mimmystrawberry2906@gmail.com

[2,3]Departemen Matematika FMIPA USU

Abstrak. Sebuah 2 *join* merupakan generalisasi dari 1 *join* dan merupakan *edge cutset* yang muncul secara alami dari dekomposisi beberapa kelas *graf* tertutup yang diambil dari *induced subgraf*. Sebuah 2 *join* digunakan untuk penyelesaian masalah optimisasi kombinatorial waktu polinomial dan berperan sampai akhir pada susunan karakteristik. Tidak semua *graf* memiliki 2 *join* maka akan dberikan algoritma untuk mendeteksi adanya 2 *join* pada sebuah *graf* yang difokuskan untuk 4-*tuple*. *Graf* yang dapat dideteksi memiliki 2 *join* merupakan terhubung yang dapat dipartisi.

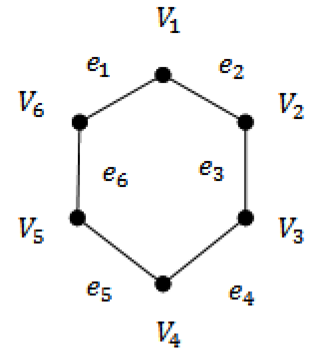
Kata kunci: 2 *join*, 1 *join*, *Edge Cutset*, generalisasi, dekomposisi, Optimisasi Kombinatorial, 4-*tuple*.

Pendahuluan

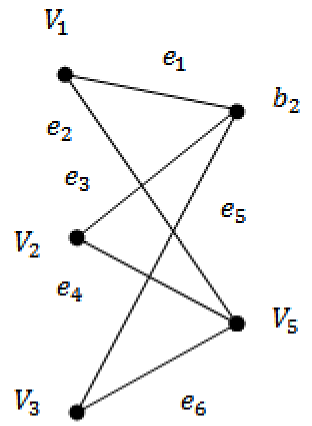
1. Latar belakang

Ada banyak permasalahan dari optimisasi kombinatorial, salah satunya adalah 2 *join* pada *graf*. *Graf* yang digunakan disini semuanya merupakan *graf* sederhana. Suatu *graf* G terdiri dari dua himpunan yang berhingga, yaitu himpunan *vertex-vertex* tak kosong (V(G)) dan himpunan *edge-edge* (E(G)). Dua *vertex* dikatakan terhubung/*adjacent* jika ada *edge* yang menghubungkan keduanya.

Pada sebuah *graf*, yang dikatakan 2 *join* adalah *graf* yang dapat di-*bipartite*. Suatu *graf* G disebut *graf* bipartisi jika dan hanya jika panjang setiap sirkuit dalam *graf* tersebut adalah genap. Sirkuit merupakan suatu *walk* tertutup. Jika V(G) merupakan gabungan dari dua himpunan tak kosong $V_1\ dan\ V_2$ dan setiap garis pada G menghubungkan suatu titik di V_1 dengan *vertex* di V_2. Apabila pada *graf bipartite* untuk setiap *vertex* di V_1 terhubung dengan setiap *vertex* di V_2, maka *graf* tersebut merupakan *graf bipartite* lengkap. Jika V_1 terdiri dari m *vertex* dan V_2 terdiri dari n *vertex*, maka *graf bipartite* lengkap tersebut sering disimbolkan dengan $K_{m,n}$.



Gambar 1.a *Graf Bipartite*



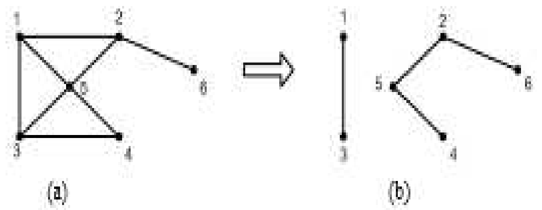
Gambar 1.b *Graf Bipartite*

Zambelli (2004) menerangkan bahwa sebuah *graf* G mempunyai 2 *join*, jika V(G) dapat dipartisi kedalam *non-empty* himpunan bagian tak kosong V_1 dan V_2 masing-masing dengan panjang paling sedikit 3, dengan *disjoint* himpunan pasangan tak kosong himpunan bagian $A_1, B_1 \subseteq V_1$ dan $A_2, B_2 \subseteq V_2$ bahwa setiap *vertex* di A_1 adalah *adjacent* ke setiap *vertex* di A_2, setiap *vertex* di B_1 adalah *adjacent* ke setiap *vertex* di B_2 dan tidak ada *edge* lain diantara V_1 dan V_2

Sebuah 2 *join* merupakan *edge cutsets* yang muncul secara alami dari dekomposisi beberapa kelas dari *graf* tertutup yang diambil dari *induced* sub*graf*, seperti *graf skew bipartite*, *graf even-hole-free*, *graf perfect* dan *graf claw-free*. 2 *join* selalu digunakan untuk penyelesaian masalah optimisasi kombinatorial waktu polynomial: menemukan kelompok bobot maksimum dan *no balance skew partition*, *no homogenues pair* dan menemukan pewarnaan optimal untuk *graf Berge* dengan pengelompokan bobot maksimum untuk *graf even-hole-free* dengan *no star cutset* (Trotigno dan Vuskovic, 2011).

Cut-set dari suatu *graf* terhubung G adalah himpunan sisi yang apabila dibuang dari G menyebabkan G tidak terhubung. Maka *cut-set* menghasilakan dua buah komponen terhubung. Didalam sebuah *cut-set* tidak boleh mengandung himpunan bagian yang juga merupakan *cut-set*. Pada *graf* yang terdapat pada gambar 2, {(1,2), (1,5), (3,5), (3,4)} adalah *cut-set*. Terdapat banyak *cut-set* pada sebuah *graf* terhubung.

Himpunan {(1,2), (2,5)} juga merupakan *cut-set*, {(1,3), (1,5), (4,5)} adalah *cut-set*, {(2,6)} juga *cut-set*, tetapi {(1,2), (2,5), (4,5)} bukan *cut-set* sebab himpunan bagiannya {(1,2), (2,5)} adalah *cut-set*.



Gambar 2. *Cutset*

Sebuah *hole* pada *graf* G merupakan sebuah *induced* sub*graf* dari G yang merupakan sebuah *cycle* dengan panjang paling sedikit empat dan merupakan ganjil atau genap jika *graf* tersebut memiliki panjang ganjil atau memiliki panjang genap. Jadi jika dikatakan sebuah *graf* G merupakan *even-hole-free* itu berarti pada *graf* G tersebut tidak terdapat *cycle* dengan panjang genap, begitu pula untuk *odd-hole-free* yang berarti pada *graf* G tersebut tidak terdapat *cycle* dengan panjang ganjil (Chudnousky, et al,2004).

Conforti, et al (2001) mengatakan bahwa sebuah *cycle* merupakan *even* jika berisi *vertex-vertex* yang bernilai genap dan merupakan *odd* jika berisi *vertex-vertex* yang bernilai ganjil. Sebuah *hole* adalah penghubung *cycle* dengan *vertex-vertex*nya paling sedikit empat. Maka dikatakan bahwa, sebuah *graf* G berisi sebuah *graf* H, jika H adalah sebuah *induced* sub*graf* dari G dan sebuah *graf* merupakan *H-free* jika tidak terdapat H.

Sebuah 2 *join* berperan sampai akhir pada susunan karakteristik dari beberapa kelas kompleks dari *graf* tertutup dari *induced* sub*graf* dan pengenalan waktu polynomial dan hubungan algoritma dengan beberapa kelas. 2 *join* menggunakan teorema dekomposisi untuk *graf bipartite* dengan koresponden penyeimbang matrik 0, 1. Tidak setiap *graf* memiliki 2 *join*. *Graf* yang dapat dideteksi jika memiliki sebuah 2 *join* adalah *graf* terhubung.

Cornuejois dan Cunningham (1985) pertama kali mengenalkan mengenai 2 *join* dalam tulisannya "*Studying Composition Operations*" dengan hasil sempurna. Merupakan generalisasi dari *edge cutset* yang dikenal dengan 1 *join* (*join/split* dekomposisi) yang dikenalkan oleh Cunningham dan Edmonds; sebuah partisi (X_1, X_2) dari *vertex* dari sebuah *graf* G adalah sebuah 1 *join* jika untuk i = 1,2 terdapat *non-empty* $A_i \subseteq X_i$, memenuhi:

- Setiap *vertex* dari A_1 adalah adjacent ke *vertex* A_2 dan disana tidak terdapat *edge*s lainnya diantara X_1 dan X_2
- Untuk i = 1,2, $|X_i| \geq 2$.

Kemudian oleh Cornuejois dan Cunningham (1985) yang pertama menjelaskan mengenai sebuah special dari 2 *join*, 2 *join* yang sebenarnya pada *graf* G adalah sebuah partisi (X_1, X_2) dari V(G) terdapat *disjoint non-empty* $A_i, B_i \subseteq X_i$ $(i = 1,2)$, memenuhi:

- Setiap *vertex* dari A_1 adalah adjacent ke semua *vertex* dari A_2, dan setiap *vertex* dari B_1 adalah *adjacent* ke semua *vertex* dari B_2
- Tidak terdapat *edge-edge* lain diantara X_1 dan X_2
- Untuk i = 1,2, setiap komponen dari $G|X_i$ keduanya memenuhi A_i dan B_i
- Untuk i = 1,2, jika $|A_i| = |B_i| = 1$ dan $G|X_i$ adalah sebuah path nilai gabungan dari A_i dan B_i, maka memiliki panjang ganjil ≥ 3.

Pada observasi yang dilakukan Cornuejois dan Cunningham (1985) untuk *Spanning Tree* T dari G, ada 2 *join* (X_1, X_2) harus memuat *edge* dari T diantara X_1 dan X_2, maka untuk menemukan 2 *join* pada sebuah *graf*, perlu dilakukan pengecekan O(nm) pasangan dari *edge-edge* $a_1, a_2\ dan\ b_1, b_2$, diberikan jumlah waktu bergerak dari $O(n^3m)$ untuk menemukan 2 *join*. Mereka juga mengklaim bahwa ada lebih dari n aplikasi dari algoritma pendeteksi 2 *join* dibutuhkan untuk mendekomposisi sebuah *graf* ke dalam faktor-faktor yang tidak dapat diperkecil / dikurangi *graf* tersebut tidak memiliki 2 *join*.

2. Perumusan Masalah

Perumusan masalah dalam penelitian ini adalah mengidentifikasi sebuah permasalahn optimisasi kombinatorial dengan 2 *join* yang difokuskan menggunakan 4-tuple.

3. Tujuan Penelitian

Adapun yang menjadi tujuan dalam penelitian ini adalah untuk membuktikan adanya 2 *join* pada sebuah *graf* dengan menggunakan 4-tuple.

Metode Penelitian

Metode penelitian ini bersifat literatur dan kepustakaan dengan mengumpulkan informasi dari beberapa jurnal. Langkah-langkah yang dilakukan adalah sebagai berikut:

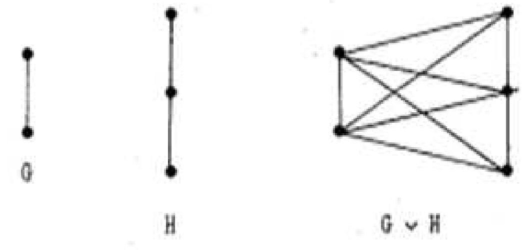
1. Membuat sebuah *graf bipartite*
2. Deteksi apakah *graf* tersebut memiliki sebuah 2 *join* atau tidak dengan algoritma.
3. Mengidentifikasi algoritma tersebut yang difokuskan untuk 4-*tuple*.
4. Membuktikan algoritma dengan 4-*tuple* untuk 2 *join*.

Hasil dan Pembahasan

1. Dua *Join*

Sebuah *graf* G yang berisi sesuatu yang terhubung *(connected)* maka *graf* G tersebut mempunyai sebuah (bisa berupa 1 *join*, 2 *join*,…, *join*).

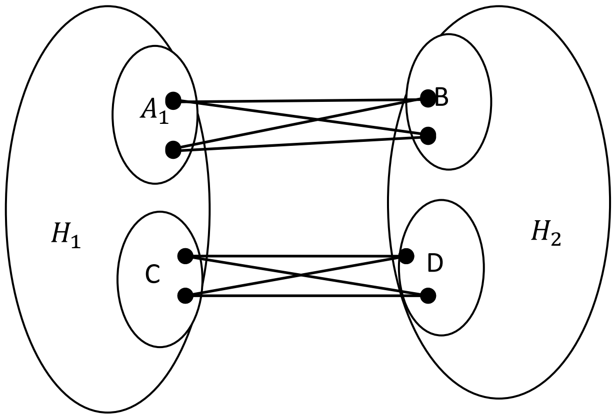
Join dari dua buah *graf* G_1 dan G_2 yang dinotasikan dengan $G_1 + G_2$ merupakan *graf* yang terdiri dari $G_1 \cup G_2$ dan semua sisi uv, dengan $u \in V(G_1)$ dan $v \in V(G_2)$. Pada gambar 5 diperlihatkan contoh bentuk *join* untuk *graf* K_3 dan K_2, *join* yang dimaksud yang ditunjukan pada gambar dari 2 buah *graf* tersebut adalah apabila setiap *vertex* dari masing-masing *graf* saling dihubungkan oleh *edge* baru sehingga kedua *graf* menjadi terhubung sehingga keduanya menjadi *graf* terhubung.



Gambar 3. *Join* dari 2 *graf* G dan H

Perlu diketahui bahwa tidak semua *graf* memiliki 2 *join*. Maka akan muncul pertanyaan *graf* seperti apa yang memiliki 2 *join*?, mengapa 2 *join* itu perlu dideteksi? *Graf* yang dapat dideteksi memiliki 2 *join* adalah *graf* tertutup yang diambil dari *subgraphs* yang diinduksi, seperti *graph bipartite* seimbang, *graph even-hole-free*, graph sempurna dan *graph claw-free*, dan *graf* terhubung. Mengapa perlu untuk mendeteksi sebuah 2 *join* pada *graf* terhubung dengan cepat? karena 2 *join* sangat berperan pada susunan karakteristik dari beberapa kelas kompleks dari *graf* tertutup dari induksi *subgrafs* pada permasalahan optimisasi kombinatorial.

Teorema dekomposisi untuk *graf even-hole-free* dilakukan dengan menggunakan 2 *join*. *Cutsets* yang digunakan untuk dekomposisi *graf even-hole-free* adalah *edge cutset* yang disebut 2 *join* dan *vertex cutset* yang disebut *star*, *double star* dan triple *star cutsets*. Sebuah *graf* G terhubung memiliki sebuah 2 *join*, dinotasikan dengan $H_1|H_2$ dengan sets A,B,C,D merupakan *nonempty* dan *disjoint*, jika *vertex-vertex* dari G dapat dipartisi ke sets H_1 dan H_2 jadi $A, C \subseteq H_1$, $B, D \subseteq H_2$, semua *vertex* dari A adalah adjacent ke semua *vertex-vertex* di B, semua *vertex* dari C adalah *adjacent* ke semua *vertex-vertex* di D dan hanya ada *adjacies* diantara H_1 dan H_2. Untuk $i = 1,2$ $|H_i| > 2$ dan jika A dan C (resp. B dan D) adalah keduanya dari cardinal 1 maka *graf induced* dengan $H_1(resp. H_2)$ adalah bukan sebuah penghubung *path*.



Gambar 4. 2 *Join*

Mendeteksi sebuah 2 *join* pada sebuah *graf* sangat jelas berkaitan untuk mendeteksi sebuah 2 *join* pada sebuah *graf* terhubung, jadi pada input *graf* pada algoritma dapat diasumsikan menjadi terhubung. Dapat ditunjukan dengan n merupakan nomor dari *vertex* dari sebuah input *graf* G dan dengan m merupakan nomor dari *edge* dari *graf* G. Pada penelitian sebelumnya dalam sebuah algoritma $O(n^3m)$ untuk menemukan sebuah 2 *join* pada sebuah *graf* G (atau dalam mendeteksi tidak ditemukan satu pun) akan ditunjukan. Algoritma merupakan dasar dari sebuah himpunan dari ukuran pembagi bahwa untuk pasangan yang diberikan dari *edge* a_1, a_2 dan b_1, b_2 jelas, pada waktu $O(n^2)$, apakah terdapat sebuah 2 *join* dengan bagian $(X_1, X_2 A_1, B_1, A_2, B_2)$ dimana untuk $i = 1,2$, $a_i \in A_i$ dan $b_i \in B_i$, dan akan ditemukan jika ada. Berikutnya dijelaskan sebuah metode baru untuk mencapai tujuan dengan mudah pada waktu $O(n + m)$.

Pada *graf* terhubung berikut algoritma untuk mencari 2 *join*:

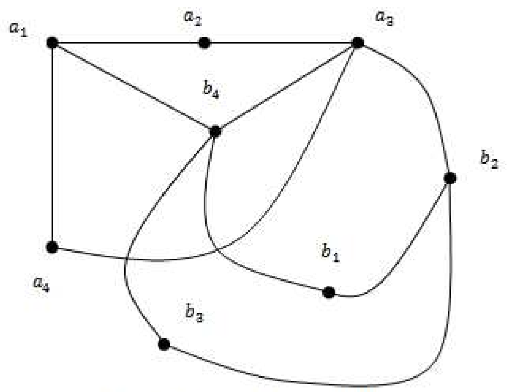
Input : Sebuah *graf* G terhubung.
Output: Sebuah *graf* G dengan 2 *join* $X_1|X_2$ dengan himpinan kusus (A_1, A_2, B_1, B_2), jika vetex-*vertex* dari himpunan G dapat dipartisi ke himpunan $X_1 \, dan \, X_2$, jika terpenuhi:

(i) Untuk i = 1,2, $A_i \cup B_i \subseteq X_i$ dimana $A_i \, dan \, B_i$ merupakan *non-empty* dan *disjoint*.
(ii) Setiap *vertex* dari A_1 *adjacent* ke semua *vertex* di A_2, dan setiap *vertex* dari B_1 *adjacent* ke semua *vertex* di B_2.
(iii) Untuk i = 1,2; setiap komponen dari $G|X_i$ keduanya memenuhi $A_i \, dan \, B_i$.
(iv) Untuk i = 1,2; jika $|A_i| = |B_i| = 1$ dan $G[X_i]$ sebuah *path* gabungan dari $A_i \, dan \, B_i$, maka memiliki panjang masing-masing ≥ 3.
(v) $G[V_i]$ memuat sebuah *path*, tetapi bukan sebuah *path* penghubung.

Running time: $O(n^2)$

2. Kasus mencari sebuah 2 *Join* compatible dengan 4-*Tuple*

Perlu dilakuka pembuktian untuk 2 *join* yang compatible dengan 4-tuple, maka dibuatlah sebuah *graf* terhubung untuk membuktikannya sebagai berikut:



Gambar 5. *Graf* Terhubung

Dari *graf* terhubung tersebut dapat dibuat sebuah *graf* bipartisi yang akan compatible dengan 2 *join* (X_1, X_2) dari G jika $a_1, b_1 \in X_1$ dan $a_2, b_2 \in X_2$.

Dengan
$E = \{(a_1, a_2), (a_1, a_4), (a_1, b_4), (a_3, a_2),$

$(a_3, a_4), (a_3, b_4), (a_3, b_2)\};$

$\{(b_1, b_4), (b_1, b_2), (b_3, b_4), (b_3, b_2)\}.$

Kemudian *graf* terhubung tersebut akan dipartisi . *Graf* bipartisi tersebut yang akan dibuktikan untuk 4-*tuple* berikut adalah algoritmanya:

Input: S_0 sebuah himpunan dari *vertex-vertex* dari sebuah *graf* G yang mana: $|S_0| \geq 3$ dan empat *vertex-vertex* a_1, b_1, a_2, b_2 pasangan yang berbeda dengan: $a_1, b_1 \in S_0$, $a_2, b_2 \notin S_0$, $a_1a_2, b_1b_2 \in E$ dan $a_1b_2, b_1a_2 \notin E$.

Permulaan:

$S \leftarrow S_0; T \leftarrow V(G)\backslash S_0;\ A \leftarrow N(a_1) \cap T;\ B \leftarrow N(b_1) \cap T;$

If $A \cap B \neq \emptyset$ maka lanjutkan $(A \cap B)$;

Vertex-vertex a_1, b_1, a_2, b_2 adalah pisahkan yang tidak ditandai. Untuk *vertex-vertex* lainnya dari G:

Tandai (x) ← $\alpha.\beta$ untuk setiap *vertex* $x \in N(a_2) \cap N(b_2)$;

Tandai (x) ← α untuk setiap *vertex* $x \in N(a_2)\backslash N(b_2)$;

Tandai (x) ← β untuk setiap *vertex* $x \in N(b_2)\backslash N(a_2)$;

Setiap *vertex* lain dari G ditandai dengan ϵ;

Catat bahwa sebuah *vertex* dapat tidak ditandai, atau ditandai dengan $\epsilon, \alpha, \beta \; atau \; \alpha.\beta$.

Loop utama:

While terdapat sebuah *vertex* $x \in S$ ditandai

Do pencarian (x); tidak ditandai (x);

Fungsi pencarian (x):

Kasus nilai dari harga (x):

If tanda (x) = $\alpha.\beta$ *then* Berhenti

OUTPUT: tidak ada 2 *join* (S,T) dengan $S_0 \subset S$ yang *compatible* dengan 4-tuple.

If tanda (x) = α then bergerak $\big(A\Delta(N(x) \cap T)\big)$;

If tanda (x) = β then bergerak $\big(B\Delta(N(x) \cap T)\big)$;

If tanda (x) = ϵ then bergerak $(N(x) \cap T)$;

Fungsi bergerak (Y):
Fungsi ini hanya memindahkan sebuah himpunan bagian $Y \subset T$ dari T ke S.
$S \leftarrow S' \cup Y;\ A \leftarrow A\backslash Y;\ B \leftarrow B\backslash Y;\ T \leftarrow T\backslash Y$;

Kesimpulan

Dari penyelesai kasus hanya difokuskan untuk 4-*tuple* yang *compatible* berdasarkan algoritma yang ada dengan langkah-langkah yang telah dijelaskan maka .ketika semua bergerak dari T ke S merupakan hal penting (berasal dari item terakhir), jika ditemukan sebuah *vertex* dengan nama α, β di S maka tidak diharapkan ada 2 *join* pada *graf* tersebut. Jika proses tidak berhenti karena dari sebuah *vertex* ditandai α, β maka semua *vertex* dari S akan diperiksa dan karena itu tidak ditandai. Jadi, jika $|T| \geq 3$, akan berakhir di (S,T), maka akan ada sebuah 2 *join* yang sesuai dengan Z: $X_1 = S, X_2 = T$, $A_1 = S \cap N(a_2)$, $B_1 = S \cap N(b_2)$, $A_2 = T \cap N(a_1)$, $B_2 = T \cap N(b_1)$. Ketika semua bergerak dari T ke S merupakan hal yang penting, 2 *join* dapat diklaim minimal (secara tidak langsung jika $|T| \leq 2$, maka tidak diharapkan ada 2 *join*). Dalam penjelasannya penyelesain untuk masalah optimisasi tidak dijelaskan lebih rinci karena mengingat kendala yang banyak yang sulit diselesaikan.

Referensi

Trotignon, N. and Vuskovic, K. (2011). Combinatirial Optimization with 2-*Join*s. Journal of Combinarotial Theory, Series B 102, 153–185.

Chvtal, V. (1985). Star-cutsets and Perfect Graphs. J. Combin. Theory, Ser.B 39, 189-199.

Charbit,P. Habib, M. Trotignon, N. Vukovic, K. (2010). Detecting 2-*join*s faster. Article in Journal of Discrete Algorithms, Issue, 1–16.

Chudnovsky, M. Cornujols, G. Liu, X. Seymour, P. Vukovic,K. (2004). Recognizing Berge Graphs. Combinatorica, 25 (2), 143-186.

Conferti, M. Cornuejois, G. Kapoor, A. and Vuskovic, K. (2001). Even-Hole-Free Graphs Part 1 : Decomposition Theorem. Journal of Graph Theory, 39,1, 6–49.