

Algoritma Penentuan Graf Bipartit

Zain Fathoni - 13508079

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Kampus ITB Jln. Ganesha No. 10 Bandung
e-mail: if18079@students.if.itb.ac.id

ABSTRAK

Teori graf merupakan pokok bahasan yang telah berusia tua namun masih diterapkan di banyak bidang sampai saat ini. Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Berbagai jenis graf beserta istilah-istilah dasarnya sering digunakan dalam pembahasan mengenai graf. Salah satu terminologi dasar dalam graf yang banyak dikenal adalah graf bipartit (*bipartite graph*). Graf ini cukup unik dengan beberapa sifat khususnya. Dalam makalah ini dibahas salah satu algoritma yang dapat digunakan untuk menentukan apakah suatu graf merupakan graf bipartit atau bukan. Algoritma tersebut berupa penelusuran simpul-simpul dalam suatu graf, dan menandainya dengan penanda yang memiliki dua elemen yang bersifat berkebalikan, misalnya bilangan biner. Dengan kompleksitas waktunya yang sebesar $O(m)$, algoritma ini cukup mangkus. Namun demikian, algoritma ini kurang bisa diterapkan di semua graf, karena algoritma ini hanya dapat menangani graf terhubung. Apabila graf yang ditangani tidak terhubung, maka komponen yang terpisah tidak akan dapat ditelusuri dengan menggunakan algoritma ini.

Kata kunci: Graf, simpul, sisi, graf bipartit.

1. PENDAHULUAN

Teori graf merupakan pokok bahasan yang telah berusia tua namun masih diterapkan di banyak bidang sampai saat ini. Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Representasi visual dari graf adalah dengan menyatakan objek dengan simpul, noktah, bulatan, titik, atau *vertex*, sedangkan hubungan antara objek dinyatakan dengan garis atau *edge* (sisi).

Berbagai jenis graf beserta istilah-istilah dasarnya sering digunakan dalam pembahasan mengenai graf. Salah satu terminologi dasar dalam graf yang banyak dikenal

adalah graf bipartit (*bipartite graph*). Graf ini cukup unik dengan beberapa sifat khususnya.

Untuk menentukan suatu graf bipartit atau bukan, tentunya diperlukan suatu cara. Dalam makalah ini akan dibahas salah satu cara yang dapat digunakan untuk menentukan apakah suatu graf bipartit atau tidak.

2. METODE

Metode penulisan makalah yang digunakan adalah metode studi pustaka, hipotesis dan analisa kasus. Setelah penulis membaca berbagai tulisan mengenai graf bipartit, penulis mencoba untuk membuat suatu algoritma yang dapat digunakan untuk menentukan apakah suatu graf bipartit atau tidak. Kemudian algoritma hasil buatan tersebut dicoba digunakan pada suatu contoh kasus untuk menentukan kemangkusan dan akurasi algoritma tersebut.

3. DASAR TEORI

3.1. Pengertian Graf

Pengertian graf adalah himpunan simpul yang dihubungkan oleh sisi-sisi.[1] Secara matematis, sebuah graf G didefinisikan sebagai pasangan himpunan (V,E) , yang dalam hal ini :

$$V = \text{himpunan tidak kosong dari simpul-simpul (vertices atau nodes)} = \{v_1, v_2, \dots, v_n\} \quad (1)$$

dan

$$E = \text{himpunan sisi (edges) yang menghubungkan sepasang simpul} = \{e_1, e_2, \dots, e_n\} \quad (2)$$

atau dapat ditulis singkat dengan notasi $G = (V,E)$. [1]

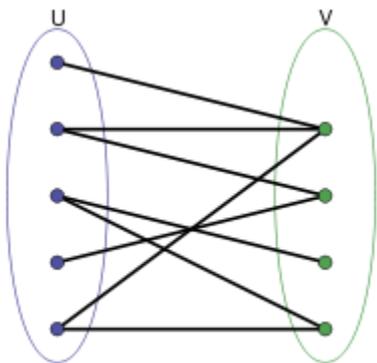
Definisi di atas menyatakan bahwa V tidak boleh kosong, sedangkan E boleh kosong. Jadi, sebuah graf dimungkinkan tidak mempunyai sisi satu buah pun, tetapi simpulnya harus ada, minimal satu. Graf yang hanya

mempunyai satu buah simpul tanpa sebuah sisi pun dinamakan graf trivial.[1]

3.2 Pengertian Graf Bipartit

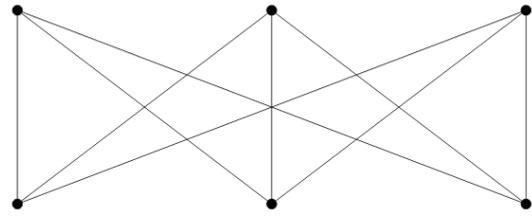
Salah satu terminologi dasar yang telah banyak dikenal dalam pembahasan mengenai graf adalah graf bipartit (*bipartite graph*). **Graf bipartit** adalah graf yang simpulnya dapat dipisah menjadi dua himpunan, misalnya bagian U dan V, sedemikian sehingga setiap sisi pada graf tersebut menghubungkan sebuah simpul di U dengan sebuah simpul di V. Graf tersebut dapat dinyatakan sebagai $G(U, V)$. Dapat dikatakan bahwa setiap pasang simpul di U tidak bertetangga, demikian pula dengan simpul-simpul di V.[1]

Kedua himpunan U dan V tersebut juga dapat dianggap sebagai suatu pewarnaan graf dengan dua warna. Apabila semua simpul di U diberi warna biru, dan semua simpul di V diberi warna hijau, masing-masing sisi akan mempunyai ujung yang berbeda warna, sama halnya dengan ketentuan pada masalah pewarnaan graf. Namun demikian, pewarnaan demikian (dua warna) tidak mungkin dilakukan pada graf nonbipartit, misalnya segitiga: setelah satu simpul diberi warna biru dan simpul lain diberi warna hijau, simpul ketiga dari segitiga tersebut bertetangga dengan simpul-simpul yang mengandung kedua warna tersebut, sehingga harus ada satu warna lagi selain kedua warna tersebut.[2] Gambar di bawah ini merupakan salah satu contoh dari graf bipartit.



Gambar 1. Contoh Graf Bipartit

Apabila setiap simpul di U bertetangga dengan semua simpul di V, maka $G(U, V)$ disebut sebagai **graf bipartit lengkap** (*complete bipartite graph*), dilambangkan dengan $K_{m,n}$, dengan m dan n masing-masing merupakan jumlah simpul di setiap himpunan (U dan V), dan jumlah sisi pada graf bipartit lengkap adalah $m \cdot n$. Gambar berikut merupakan salah satu contoh graf bipartit lengkap dengan $m = n = 3$. [1]

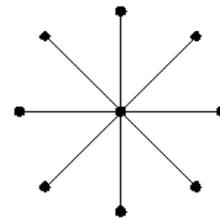


Gambar 2. Contoh Graf Bipartit Lengkap

3.3 Aplikasi Graf Bipartit

Contoh persoalan yang dapat dinyatakan sebagai graf bipartit adalah persoalan utilitas: misalkan ada tiga buah rumah: $H_1, H_2,$ dan H_3 , masing-masing dihubungkan dengan tiga buah utilitas: air (W), gas (G), dan listrik (E) dengan alat pengantar (pipa, kabel, dsb.) yang direpresentasikan dengan sisi-sisi pada graf. Graf yang merepresentasikannya sama dengan graf pada gambar 2 di atas.[1]

Salah satu contoh persoalan graf bipartit lainnya adalah topologi bintang (*star topology*) pada jaringan komputer LAN. Dalam hal ini, U berisi sebuah simpul sebagai pusat, sedangkan V berisi simpul-simpul lainnya yang merepresentasikan *client* pada topologi bintang. Graf tersebut adalah graf $K_{1,n}$. Berikut gambar grafnya.[1]



Gambar 3. Topologi Bintang

Selain kedua contoh di atas, masih banyak aplikasi graf bipartit yang berkaitan dengan kehidupan sehari-hari.

4. CARA PENENTUAN GRAF BIPARTIT

4.1 Cara Melihat secara Menyeluruh

Berbagai cara dapat dilakukan untuk menentukan graf bipartit. Salah satu cara yang dapat dilakukan adalah dengan melihat graf tersebut secara keseluruhan. Cara ini cukup cepat apabila kedua himpunan simpul (U dan V) telah terkelompokkan secara geometris seperti pada Gambar 1 dan Gambar 2.

Kelemahan dari cara ini yakni apabila bentuk graf tersebut diubah bentuknya menjadi graf lain yang isomorfik, cara ini belum tentu dapat dengan mudah dilakukan. Selain itu, kelemahan lain dari cara ini adalah,

cara ini hanya dapat dilakukan oleh manusia, karena gambar graf harus dilihat secara menyeluruh. Cara semacam ini tidak mungkin dapat diaplikasikan ke dalam suatu bentuk algoritma yang nantinya dapat diaplikasikan ke dalam komputer.

Oleh karena itu, diperlukan suatu cara yang lebih sistematis dan bersifat algoritmik, sehingga penentuan suatu graf bipartit atau bukan dapat dilakukan oleh komputer, melalui algoritma yang diaplikasikan ke dalamnya.

4.2 Algoritma untuk Penentuan Graf Bipartit

Penulis telah mencoba suatu cara yang sistematis dan bersifat algoritmik yang dapat digunakan untuk menentukan apakah suatu graf bipartit atau tidak. Cara ini lebih mungkin diaplikasikan ke dalam komputer daripada cara yang telah dibahas sebelumnya.

Berikut rincian algoritmanya:

1. Pilih sembarang simpul dalam graf tersebut. Sebut simpul tersebut sebagai simpul-X.
2. Tandai (beri label) simpul yang telah dipilih (simpul-X). Untuk mempermudah, sebagai penanda digunakan sesuatu yang hanya memiliki dua elemen yang bersifat berkebalikan. Dalam hal ini, akan digunakan bilangan biner, sehingga simpul yang telah dipilih tersebut akan diberi tanda '0'.
3. Tandai semua simpul yang bertetangga dengan simpul-X (dan belum pernah terpilih sebelumnya sebagai simpul-X) dengan tanda/label yang berlawanan dengan yang dimiliki oleh simpul-X (dalam hal ini adalah '1').
4. Pilih salah satu simpul dari simpul tetangga yang telah ditandai tadi, kemudian perlakukan seperti simpul-X pada langkah 3 (tandai semua simpul tetangganya dengan tanda yang berlawanan).
5. Apabila simpul tetangga tersebut telah memiliki tanda, pastikan bahwa tanda tersebut sama dengan tanda yang seharusnya diberikan pada langkah tersebut.
6. Ulangi langkah 3-5 hingga semua simpul telah ditelusuri (pernah dipilih menjadi simpul-X).
7. Apabila terjadi kasus konflik di mana tanda yang akan diberikan berbeda dengan tanda yang telah dimiliki oleh simpul tetangga tersebut, maka dapat disimpulkan bahwa graf tersebut bukan graf bipartit.
8. Namun apabila langkah 3-5 dapat dilakukan pada semua simpul hingga tidak ada simpul yang tersisa, maka dapat disimpulkan bahwa graf tersebut bipartit.

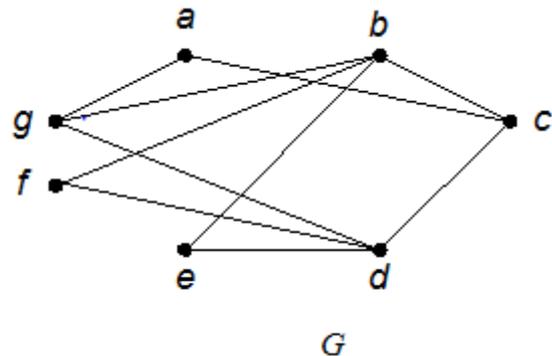
Algoritma di atas mungkin terlihat rumit dalam penjabarannya, tetapi apabila dicoba dilakukan akan terasa lebih mudah.

Algoritma di atas cukup akurat, namun hanya dapat digunakan pada graf terhubung. Apabila graf tidak terhubung, komponen yang tidak terhubung tidak dapat ditelusuri.

4.3 Analisa Kasus Algoritma Penentuan Graf Bipartit

Dalam upabab berikut akan dilakukan penerapan algoritma penentuan graf bipartit pada suatu kasus. Contoh kasus yang disediakan ada 2, yakni kasus di mana graf merupakan graf bipartit, dan graf bukan merupakan graf bipartit.

Contoh kasus 1:



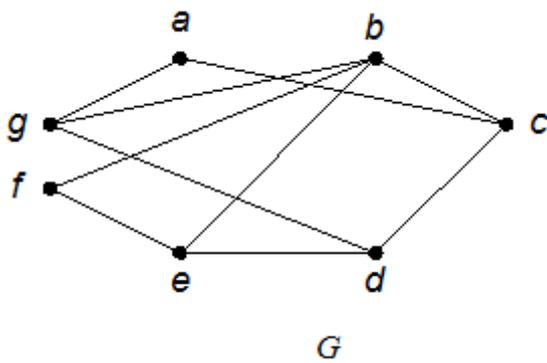
Gambar 4. Graf Bipartit

Tabel 1 Langkah-langkah Penentuan Graf Bipartit pada Contoh Kasus 1

Langkah	Simpul-X	Label simpul X	Simpul tetangga	Label simpul tetangga	Simpul bertanda '0'	Simpul bertanda '1'
1	a	0	c,g	1	a	c,g
2	c	1	b,d	0	a,b,d	c,g
3	g	1	b,d	0	a,b,d	c,g
4	b	0	e,f	1	a,b,d	c,e,f,g
5	d	0	e,f	1	a,b,d	c,e,f,g
6	e	1	-	0	a,b,d	c,e,f,g
7	f	1	-	0	a,b,d	c,e,f,g

Penelusuran simpul-X dapat dilakukan pada ketujuh simpul hingga tidak ada simpul yang tersisa, dan tidak terjadi kasus konflik, sehingga dapat disimpulkan bahwa graf tersebut bipartit.

Contoh kasus 2:



Gambar 5. Graf Tidak Bipartit

Tabel 2 Langkah-langkah Penentuan Graf Bipartit pada Contoh Kasus 2

Langkah	Simpul-X	Label simpul X	Simpul tetangga	Label simpul tetangga	Simpul bertanda '0'	Simpul bertanda '1'
1	a	0	c,g	1	a	c,g
2	c	1	b,d	0	a,b,d	c,g
3	g	1	b,d	0	a,b,d	c,g
4	b	0	e,f	1	a,b,d	c,e,f,g
5	d	0	e	1	a,b,d	c,e,f,g
6	e	1	f	0	????	????

Langkah terhenti di pengulangan yang ke-6, di mana simpul tetangga ternyata ada yang telah bertanda '1', yakni simpul f, sedangkan simpul tersebut akan diberi tanda '0'. Terjadi kasus konflik, sehingga dapat disimpulkan bahwa graf tersebut tidak bipartit.

5. ANALISA ALGORITMA PENENTUAN GRAF BIPARTIT

5.2 Notasi Algoritmik

Algoritma yang telah dijabarkan pada upabab 4.2 di atas akan dapat diurai dalam bentuk notasi algoritmik sebagai berikut:

procedure CekBipartit (input G : graf, output U, V : set of node)

Deklarasi

A : set of node {menampung simpul yang belum diberi tanda}

X : node {simpul-X}

Y : node {simpul tetangga}

Bipartit : boolean

Algoritma

U, V \leftarrow himpunan kosong

A \leftarrow simpul-simpul dari graf G

X \leftarrow simpul pertama pada himpunan A

U \leftarrow U \cup {X}

Bipartit \leftarrow true

while (masih ada simpul tetangga dari X yang belum pernah dipilih) do

 X \leftarrow simpul tetangga yang belum pernah dipilih

for i \leftarrow 1 to (banyaknya simpul tetangga dari X yang belum pernah dipilih) do

if Y \in A then {Y masih belum ditandai}

if X \in U then {Y ditandai berlawanan dgn X}

 V \leftarrow V \cup {Y}

else

 U \leftarrow U \cup {Y}

else

if ((X \in U) and (Y \in U)) or ((X \in V) and (Y \in V)) then {X dan Y bertanda sama}

 Bipartit \leftarrow false

quit {keluar dari kalang/loop}

5.2 Kompleksitas Algoritma

Dari notasi algoritmik di atas, terlihat bahwa pengulangan terus dilakukan apabila masih ada simpul tetangga yang belum pernah dipilih, maka simpul tetangga tersebut akan dipilih, dan akan dicari simpul lain yang juga belum pernah dipilih. Apabila kita cermati dengan seksama, sisi-sisi yang akan ditelusuri hanya sisi-sisi yang menghubungkan antara simpul yang belum pernah dipilih dengan simpul-simpul sekitarnya, sehingga sisi-sisi yang sudah pernah ditelusuri tidak akan ditelusuri lagi.

Dengan demikian, penelusuran sisi-sisi hanya akan dilakukan satu kali di setiap sisi, sehingga dapat dikatakan bahwa algoritma ini bergantung pada banyaknya sisi pada graf tersebut.

Jumlah operasi untuk setiap sisi yang ditelusuri hanya sebanyak tiga kali, yakni dua kali pengecekan kondisi, dan satu kali operasi *assignment*. Misalkan banyaknya sisi pada graf tersebut adalah m, maka $T(m) = 3m$.

$$3m \leq 3m, \text{ untuk } m_0 \geq 1, \text{ maka}$$

$$T(m) = O(m) \tag{1}$$

Dari hasil perhitungan di atas, dapat diketahui bahwa algoritma ini memiliki kompleksitas waktu sebesar $O(m)$. Nilai ini tergolong sangat kecil, sehingga dapat dikatakan bahwa algoritma ini cukup mangkus untuk menyelesaikan permasalahan ini.

6. KESIMPULAN

1. Cara melihat secara menyeluruh kurang sesuai untuk menentukan apakah suatu graf bipartit atau tidak. Oleh karena itu, diperlukan cara yang lebih sistematis dan bersifat algoritmik.

2. Telah ditemukan algoritma yang dapat digunakan untuk menentukan apakah suatu graf bipartit atau tidak, yakni dengan cara menelusuri simpul-simpulnya dan menandainya dengan penanda yang berkebalikan.
3. Kompleksitas algoritma ini adalah $O(m)$ yang tergolong cukup mangkus, dengan m adalah jumlah sisi pada graf tersebut.
4. Kekurangan dari algoritma penentuan graf bipartit ini adalah keterbatasan lingkup masalahnya, yakni masalah yang dapat diselesaikan hanya dalam lingkup graf terhubung.

REFERENSI

- [1] Munir, Rinaldi. "Diktat Kuliah IF2091 Struktur Diskrit". Program Studi Teknik Informatika ITB. 2004.
- [2] Wikipedia, *Bipartite Graph*
http://en.wikipedia.org/wiki/Bipartite_graph
Waktu Akses: 21 Desember 2009 pukul 17:00 WIB
- [3] Wikipedia, *Complete Bipartite Graph*
http://en.wikipedia.org/wiki/Complete_bipartite_graph
Waktu Akses: 21 Desember 2009 pukul 17:00 WIB