PENYELESAIAN PERSOALAN MATCHING LENGKAP

DENGAN MENGGUNAKAN ALGORITMA HUNGARIAN

PITUA SIALLAGAN

0396010377

UNIVERSITAS INDONESIA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

JURUSAN MATEMATIKA

DEPOK

2003

PENYELESAIAN PERSOALAN MATCHING LENGKAP

DENGAN MENGGUNAKAN ALGORITMA HUNGARIAN

Skripsi diajukan sebagai salah satu syarat

untuk memperoleh gelar Sarjana Sains

Oleh:

PITUA SIALLAGAN

0396010377

DEPOK

2003

SKRIPSI : PENYELESAIAN PERSOALAN MATCHING LENGKAP DENGAN MENGGUNAKAN ALGORITMA HUNGARIAN

NAMA : PITUA SIALLAGAN

NPM : 0396010377

SKRIPSI INI TELAH DIPERIKSA DAN DISETUJUI

DEPOK, 26 MEI 2012

Drs. SURYADI SLAMET M.Sc	HENDRI MURFI, S.Si, M.Kom
PEMBIMBING I	PEMBIMBING II

Tanggal lulus Ujian Sidang Sarjana: 10 Februari 2003

Penguji I : Drs. Suryadi SIS, M.Sc

Penguji II : Hendri Murfi, S.Si, M.Kom

Penguji III : Dra. Denny Riama Silaban, M.Kom

Kupersembahkan untuk

Mama dan Papa tercinta

Juga untuk adik-adikku

Rotua, John, Nova, dan Mei tersayang

KATA PENGANTAR

Puji dan syukur penulis persembahkan kepada Tuhan Yang Maha Esa karena kasih dan anugerahNya penulis berhasil menyelesaikan skripsi ini yang merupakan salah satu syarat untuk menempuh ujian kesarjanaan pada jurusan Matematika FMIPA Universitas Indonesia.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna, baik dari materi maupun sistematis pembahasannya. Oleh karenanya, segala kritik dan saran yang membangun yang berkenaan dengan skripsi ini akan penulis terima dengan senang hati.

Selama proses penelitian dan penyusunan skripsi ini, penulis banyak menerima bantuan dari berbagai pihak, baik berupa bimbingan, saran maupun dorongan moril dan materiil.

Pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya secara khusus kepada Bapak Drs. Suryadi Slamet selaku pembimbing pertama dan Bapak Hendri Murfi, M.Kom selaku pembimbing kedua yang telah memberikan petunjuk, saran, dan dorongan serta pemikiran yang sangat berarti selama penulisan.

Pada kesempatan ini juga penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Drs, Suryadi Slamet, selaku Ketua Jurusan Matematika pada Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA) Universitas Indonesia.

2. Bapak Drs. Ponidi Msi, selaku dosen wali, yang telah banyak membantu memberikan dorongan dan semangat kepada penulis.
3. Seluruh dosen jurusan Matematika FMIPA Universitas Indonesia dan staf pengajar yang telah membekali ilmu pengetahuan yang sangat berharga selama penulis menempuh pendidikan di Universitas Indonesia.
4. Seluruh staf administrasi dan perpustakaan jurusan Matematika FMIPA Universitas Indonesia atas semua bantuan yang diberikan dalam kelancaran penulisan skripsi ini.
5. Bapa dan Mama, serta adik-adikku Rotua, John, Rinova, dan Meida atas curahan cinta dan kasih sayang dan juga yang telah banyak memberikan doa, dukungan dan semangat sehingga penulis mampu menyelesaikan skripsi ini.
6. Teman-teman angkatan 1996 terutama Kikie, Susanti, Helen, Endah, Sunar, Joy, Ifan, Bobby, Opik, Umar, Hery, Miftah, Ali, Peter John, dll, yang telah memberikan dukungan doa dan dorongan selama penulis menyelesaikan skripsi ini.
7. Sahabatku Vannie, Debbie, Sonny, Serti, Lani, Lina alias Once, yang telah banyak membantu dengan arahan, semangat, dorongan dan dukungan doa selama penyusunan skripsi ini.
8. Teman-temanku di kost Pondok Ripi : Heru "Makasih banyak ru..", Iik, Batoe, Rudi, Yurry, Joko, Joger, Daud atas warnetnya, Tio, Iboy, Ai, Oeboer alias Wisnu, Ferry, Soichi, Kiki, MJ Cs dll yang tidak bisa

disebutkan satu-persatu, yang telah memberikan semangat dan dorongan selama penyusunan skripsi ini.

9. Teman-temanku di depan halte FMIPA UI: Cece dan Akang yang mau memberikan hutangan, Hafid, Dodo, Ratih, Agung, Bambie, Rio, Jul, Widhi, Andhika dll.
10. Ruth Veronica yang turut memberiku semangat dari jauh "Semoga kau berhasil mengejar cita-citamu ya..".
11. Watashi ni jikan ga ogeru mayuu san ni, kono san ka getsu kan renraku ni naru, watashi no rombun sakusei
12. Semua pihak yang tidak dapat penulis sebutkan satu persatu yang banyak memberikan bantuan untuk penyelesaian skripsi ini.

Depok, Januari 2003

Penulis

ABSTRAK

Tugas akhir ini membahas penyelesaian persoalan *matching* lengkap dengan menggunakan graf *bipartite* sebagai modelnya. Teori yang digunakan untuk menentukan penyelesaian persoalan tersebut adalah teorema Hall. Dan algoritma yang dibahas untuk menjawab ada atau tidaknya *matching* lengkap pada persoalan tersebut adalah algoritma Hungarian.

Kata kunci : *matching* lengkap; algoritma Hungarian; teorema Hall.

v + 33 hlm.

Bibliografi : 4 (1996-2001)

DAFTAR ISI

Halaman

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam skripsi ini akan dibahas penyelesaian persoalan *matching lengkap.* Untuk membahas persoalan ini akan dibahas pula teori yang digunakan untuk menentukan syarat perlu dan cukup agar persoalan dapat diselesaikan dan sebuah algoritma yang akan menghasilkan penyelesaian. Teori dimaksud adalah Teorema Hall dan algoritma dimaksud adalah algoritma Hungarian.

Persoalan *matching lengkap* akan dijelaskan melalui contoh berikut. Pada suatu perusahaan dibutuhkan beberapa tenaga untuk mengisi beberapa lowongan. Sejumlah pelamar yang mempunyai keahlian tertentu melamar untuk menempati sejumlah terbatas lowongan yang tersedia dengan kebutuhan keahlian tertentu. Ada satu persyaratan mutlak yaitu kesesuaian antara keahlian yang dikuasai oleh pelamar dengan kebutuhan di tempat kerja. Pihak personalia harus melakukan penempatan yang tepat, yaitu pelamar harus dapat memperoleh penempatan yang sesuai dengan keahliannya. Tidak tertutup kemungkinan ada pelamar yang memiliki kualifikasi yang cocok untuk lebih dari satu posisi yang ditawarkan perusahaan tersebut.

Tugas ini mungkin terasa berat bagi pihak personalia. Karena apabila ini tidak dijalankan dengan tepat akan berpengaruh terhadap produktivitas perusahaan tersebut. Kemudian pertanyaan yang muncul dalam benak pihak personalia adalah apa syaratnya agar perusahaan tersebut dapat berhasil menempatkan pegawai baru yang sesuai dengan kualifikasi yang diharapkan di tempat yang tepat.

Persoalan ini, yang dikenal dengan nama persoalan *matching lengkap*, dapat dipecahkan dengan menggunakan graf. **Philip Hall** pada tahun 1935 memecahkan persoalan seperti ini yang disebutnya sebagai persoalan perkawinan. Persoalan perkawinan ini adalah sebagai berikut: jika diberikan sebuah himpunan berhingga dari anak gadis dan sebuah himpunan berhingga dari anak pria, di mana setiap anak gadis mengenal beberapa anak pria, maka syarat apa yang diperlukan agar semua anak gadis dapat dikawinkan dengan anak pria yang dikenalnya ?

Hal ini yang mendasari penulis untuk mencoba mengaplikasikannya pada permasalahan lain yaitu persoalan penempatan himpunan pelamar pada himpunan lowongan pekerjaan yang tepat sesuai dengan kualifikasi yang dimiliki pelamar. Kemudian mencoba menyelesaikannya menggunakan algoritma Hungarian untuk mengetahui ada atau tidaknya *matching lengkap* pada persoalan tersebut.

1.2 Tujuan Penulisan

Tujuan penulisan ini adalah untuk menyelesaikan persoalan *matching lengkap* dengan menggunakan graf bipartite sebagai model. Untuk dapat menyelesaikan persoalan ini akan dibahas teorema Hall dan algoritma Hungarian untuk menjawabnya.

1.3 Metodologi Penulisan

Metode yang digunakan dalam penulisan skripsi ini adalah studi literatur dan analisa masalah.

1.4 Sistematika Penulisan

Berikut ini adalah pembahasan dari tiap-tiap bab yang akan dibahas dalam penulisan tugas akhir ini.

Bab I Berisi pendahuluan mengenai latar belakang masalah, tujuan penelitian serta metode yang digunakan dalam penulisan.

Bab II Berisi dasar-dasar teori dari penyelesaian masalah.

Bab III Berisi penggunaan teorema Hall untuk menyelesaikan permasalahan tersebut.

Bab IV Berisi pembahasan metode Hungarian untuk menjawab permasalahan tersebut.

Bab V Merupakan penutup dari pembahasan yang dirangkum dalam suatu kesimpulan mengenai kegunaan teorema Hall dan algoritma Hungarian untuk menyelesaikan masalah *matching lengkap*.

BAB II

LANDASAN TEORI

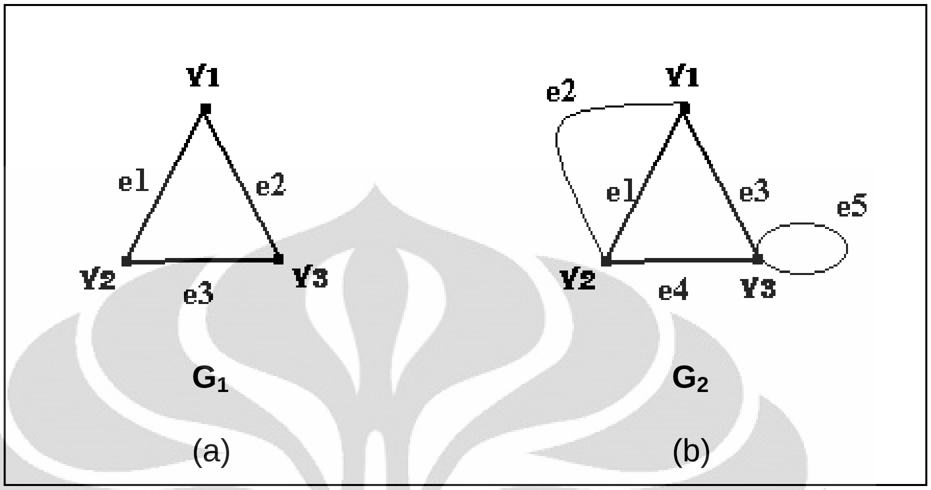
Sebelum masuk kedalam masalah tersebut, berikut ini akan dibahas teori-teori yang mendukung dan merupakan bagian dari masalah *matching lengkap* ini. Persoalan ini sangat erat hubungannya dengan graf. Sebelum pembahasan dilanjutkan, berikut ini akan diberikan beberapa definisi dari teori graf untuk digunakan dalam penyelesaian masalah di atas.

2.1 Definisi Graf

Definisi 2.1.1

Suatu graf G(V,E) didefinisikan sebagai pasangan himpunan (V,E) di mana V adalah himpunan berhingga dan tidak kosong dari simpul-simpul (*vertices* atau *nodes*) dan E adalah himpunan berhingga dari busur-busur (*edges*).

Biasanya graf ini sering hanya ditulis sebagai graf G, tanpa menyatakan himpunan V dan E-nya. Busur pada graf digambarkan sebagai garis yang kedua ujungnya merupakan simpul. Simpul dinyatakan dengan huruf kecil u, v,… atau v_1, v_2,… atau dengan bilangan asli 1, 2, 3,… dan digambarkan sebagai bulatan kecil. Sedangkan busur dinyatakan dengan e_1, e_2,… atau dengan kedua ujungnya (v_1,v_2), (v_1,v_3),… (lihat gambar 2.1a dan 2.1b).



Gambar 2.1

Pada G_1, himpunan simpulnya adalah $V_1 = \{v_1,v_2,v_3\}$ dan himpunan busurnya adalah $E_1 = \{e_1,e_2,e_3\} = \{(v_1,v_2), (v_1,v_3), (v_2,v_3)\}$ dan pada G_2, himpunan simpulnya adalah $V_2 = \{v_1,v_2,v_3\}$ dan himpunan busurnya adalah $E_2 = \{e_1,e_2,e_3,e_4,e_5\} = \{(v_1,v_2), (v_1,v_2), (v_1,v_3), (v_2,v_3), (v_3,v_3)\}$. Pada G_2, busur e_5 disebut sebagai gelang (*loop*) dan busur e_1 dan e_2 disebut sebagai busur ganda (*multiple edges*) karena kedua busur ini menghubungi dua simpul yang sama yaitu simpul v_1 dan v_2.

Definisi 2.1.2

Graf sederhana (*simple graph*) adalah graf yang tidak mengandung gelang dan busur ganda.

Banyaknya simpul pada suatu graf G(V,E), yaitu $|V|$, dinyatakan dengan huruf *n*, sedangkan banyaknya busur, yaitu $|E|$, dinyatakan dengan huruf *m*. Dua buah simpul v_1 dan v_2 dikatakan bertetangga (*adjacent*) bila dalam graf tersebut terdapat busur e dengan v_1 dan v_2 sebagai ujungnya; atau dengan kata lain, dalam himpunan busur E terdapat busur e = (v_1,v_2). Busur e = (v_1,v_2) dikatakan hadir (*incident*) pada simpul v_1 dan v_2 (demikian pula sebaliknya, v_1 dan v_2 hadir pada busur e). Demikian pula halnya dengan busur, dua buah busur dikatakan bertetangga bila kedua busur tersebut mempunyai suatu simpul persekutuan v (bertemu pada simpul v). Banyaknya busur yang hadir pada suatu simpul v disebut sebagai derajat (*degree*) dari simpul v dan ditulis sebagai d(v).

2.2 Jenis-jenis Graf

Graf sendiri terdiri dari berbagai macam antara lain:

a. Graf Berarah (*Directed Graphs*), yaitu suatu graf di mana pada setiap busurnya diberikan suatu orientasi arah.

b. Graf Kosong (*Null Graphs*), yaitu suatu graf di mana himpunan busurnya merupakan himpunan kosong dengan kata lain graf ini berbentuk sebagai himpunan dari n buah simpul terisolasi.

c. Graf Lengkap (*Complete Graphs*), yaitu suatu graf sederhana di mana setiap pasang simpulnya bertetangga.

d. Graf Teratur (*Regular Graphs*), yaitu suatu graf di mana setiap simpulnya mempunyai derajat yang sama.

e. Graf Bipartite (*Bipartite Graphs*), yaitu suatu graf G di mana himpunan simpulnya dapat dipisah menjadi dua himpunan bagian V_1 dan V_2, sedemikian sehingga setiap busur di G menghubungkan sebuah simpul di V_1 ke sebuah simpul di V_2.

f. Graf Lingkaran dan graf roda (*Cycles* dan *Wheels*), yaitu suatu graf yang bentuknya menyerupai lingkaran.

g. Graf Lintasan (*Paths*), yaitu suatu graf yang bentuknya menyerupai sebuah garis lurus.

h. Graf Planar (*Planar Graphs*), yaitu suatu graf yang dapat digambarkan pada suatu bidang datar di mana busur-busurnya tidak saling berpotongan.

i. Graf Platonic (*Platonic Graphs*), yaitu graf yang dibentuk dari simpul dan busur lima benda Platonic teratur: tetrahedron, octahedron, cube (kubus), icosahedron dan dodecahedron.

Pada pembahasan selanjutnya graf yang akan digunakan untuk menyelesaikan permasalahan di atas adalah graf bipartite.

2.3 Graf Bipartite

Definisi 2.3.1

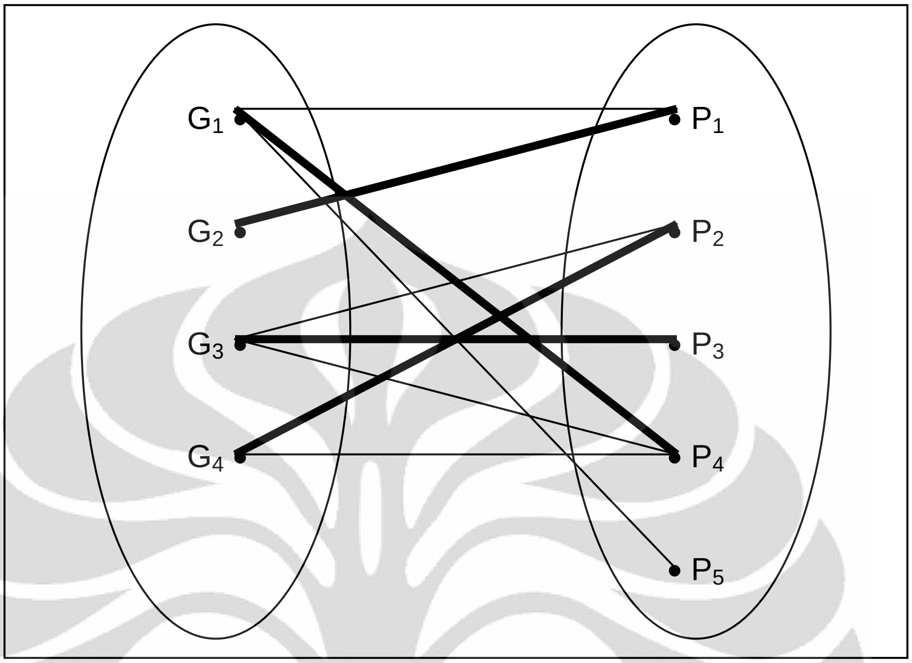
Graf bipartite adalah suatu graf G dimana himpunan simpulnya dapat dipisah menjadi dua himpunan bagian X dan Y sedemikian sehingga setiap busur di G menghubungkan sebuah simpul di X ke sebuah simpul di Y ($E \subseteq X \times Y$) dan dinyatakan sebagai $G(X \cup Y, E)$ dimana $X \cap Y = \emptyset$.

Apabila setiap simpul di X bertetangga dengan semua simpul di Y, maka $G(X \cup Y, E)$ disebut sebagai graf bipartite lengkap (*complete bipartite graph*) dan ditulis sebagai $K_{r,s}$, dimana $r = |X|$ dan $s = |Y|$.

Graf bipartite ini dapat disesuaikan dengan pokok persoalan *matching lengkap*, misalnya himpunan X adalah himpunan gadis dan himpunan Y adalah himpunan pria. Busur-busur dari X ke Y menunjukkan relasi kenal, yaitu bila terdapat busur (x_i, y_j) maka ini berarti gadis x_i kenal pria y_j. Dengan model ini persoalan *matching lengkap* atau persoalan perkawinan akan di-selesaikan.

gadis	pria yang dikenal gadis
G_1	P_1 P_4 P_5
G_2	P_1
G_3	P_2 P_3 P_4
G_4	P_2 P_4

Gambar 2.2

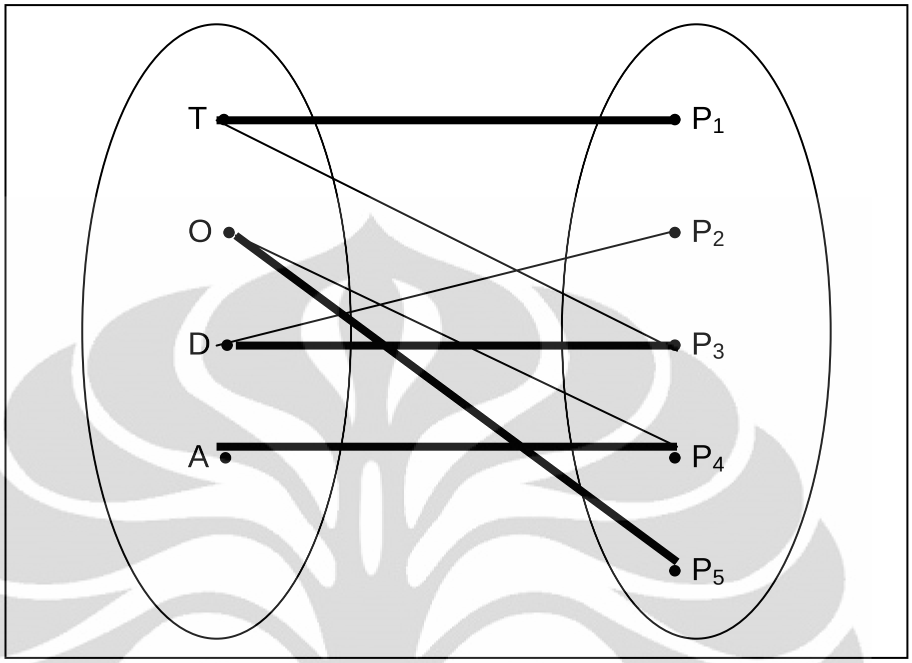


Gambar 2.3

Contoh model di atas dapat juga dimodifikasi untuk model penempatan tenaga kerja, seperti ditunjukkan pada gambar berikut:

lowongan pekerjaan	pelamar
Teknisi (T)	P_1 P_3
Operator (O)	P_4 P_5
Desainer (D)	P_2 P_3
Akuntan (A)	P_4

Gambar 2.4



Gambar 2.5

Dari model di atas, diharapkan perusahaan tersebut mendapatkan calon pegawai yang memenuhi kualifikasi tertentu. Untuk kasus di mana sudut pandang yang digunakan dari sisi penyalur tenaga kerja, himpunan anggota X yang semula beranggotakan himpunan lowongan pekerjaan diganti dengan himpunan pelamar. Hal ini berlaku juga sebaliknya untuk anggota himpunan Y.

Semua model permasalahan di atas dapat ditunjukkan dalam graf bipartite atau graf bagian (*subgraph*) dari graf bipartite lengkap.

Definisi 2.3.2

Suatu graf $G_1(V_1,E_1)$ disebut sebagai graf bagian (*subgraph*) dari graf G(V,E) apabila dipenuhi $V_1 \subseteq V$ dan $E_1 \subseteq E$ yaitu bila himpunan simpul dan busur dari G_1 merupakan himpunan bagian dari himpunan simpul dan busur dari G.

Pada pemodelan di atas dimana setiap pelamar melamar pada lowongan pekerjaan yang berbeda seperti yang ditunjukkan pada gambar 2.3 dan gambar 2.5 merupakan salah satu contoh persoalan *matching* yaitu *matching lengkap*.

2.4 Matching

Definisi 2.4.1

Matching pada suatu graf G adalah sebuah himpunan M dari busur-busur di G dimana tidak ada pasangan busur-busur pada M yang saling bertetangga (hadir di satu simpul)

Definisi 2.4.2

Matching lengkap dari X ke Y pada graf bipartite G(X,Y) sebagai hubungan satu-satu antara simpul-simpul di X dengan simpul-simpul di Y, sedemikian sehingga semua simpul di X mempunyai pasangan di Y.

Definisi terakhir ini merupakan dasar dari ditemukannya teorema Hall dan akan dibahas dalam bab selanjutnya.

BAB III

TEOREMA HALL

Dari bab sebelumnya telah diberikan gambaran singkat penggunaan graf bipartite untuk memodelkan persoalan *matching lengkap*. Pada bab ini akan dibahas pemodelan persoalan *matching lengkap* secara lebih rinci dan syarat perlu dan cukup untuk memperoleh penyelesaian.

3.1 Permasalahan

Perrmasalahan *matching lengkap* dapat dimodelkan sebagai berikut: Sebuah perusahaan kontraktor mengiklankan lowongan pekerjaan untuk tukang batu (B), tukang kayu (K), tukang listrik (L), tukang pipa (P) dan tukang cat (C). Ada 5 pelamar {P_1, P_2, P_3, P_4, P_5} yang melamar pada perusahaan kontraktor tersebut dengan keahlian sebagai berikut:

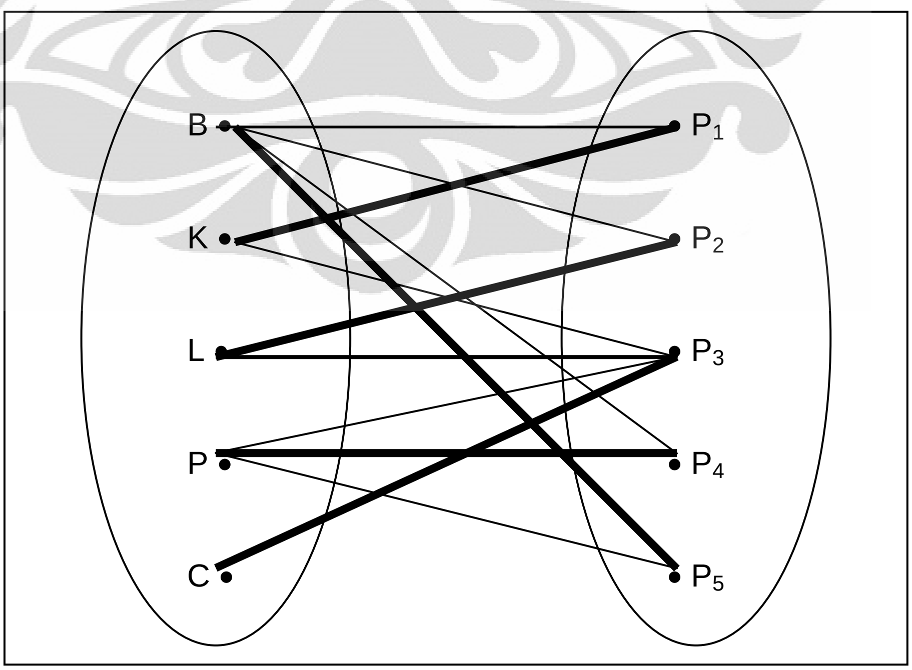
- Seorang dengan keahlian tukang batu dan tukang kayu (P_1)
- Seorang dengan keahlian tukang batu dan tukang listrik (P_2)
- Seorang dengan keahlian sebagai tukang kayu, tukang listrik, tukang pipa dan tukang cat (P_3)
- Dua orang dengan keahlian sebagai tukang batu dan tukang pipa (P_4 dan P_5)

Pertanyaannya adalah, dapatkah persoalan ini diselesaikan dengan syarat bahwa semua lowongan pekerjaan diisi oleh tukang yang tepat ? Persoalan ini dapat digambarkan dalam bentuk graf bipartite G(X,Y) dimana himpunan lowongan pekerjaan merupakan himpunan simpul X dan himpunan pelamar merupakan himpunan simpul Y.

Hal ini dapat ditunjukkan dalam gambar tabel sebagai berikut.

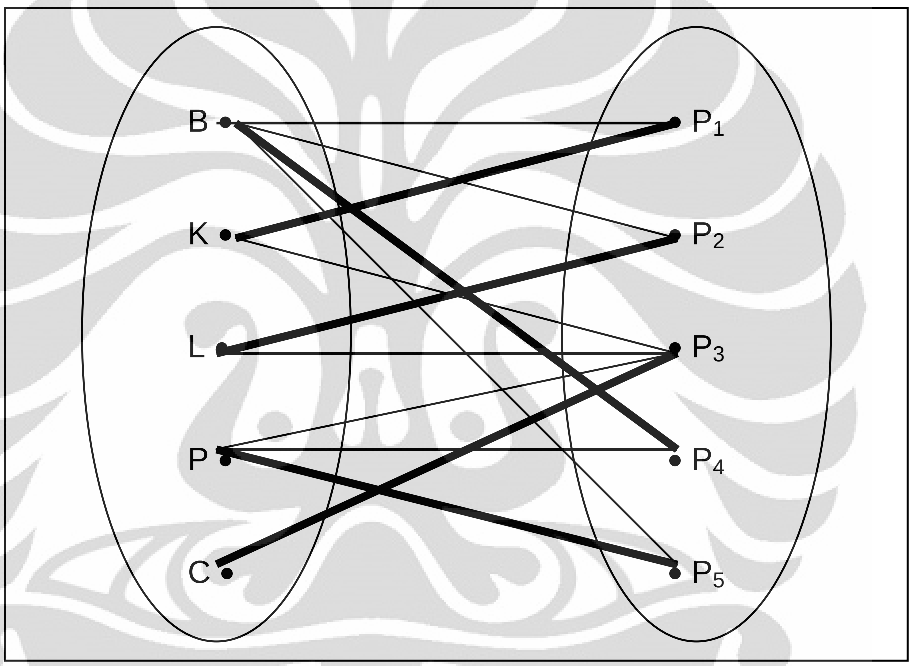
lowongan pekerjaan	pelamar
B	P_1 P_2 P_4 P_5
K	P_1 P_3
L	P_2 P_3
P	P_3 P_4 P_5
C	P_3

Gambar 3.1



Gambar 3.2

Dari gambar 3.2 ditunjukkan kemungkinan penyelesaian dari persoalan di atas yaitu tukang batu diisi oleh pelamar P_5, tukang kayu diisi oleh pelamar P_1, tukang listrik diisi oleh pelamar P_2, tukang pipa diisi oleh pelamar P_4, dan tukang cat diisi oleh pelamar P_3.



Gambar 3.3

Dari gambar 3.3 ditunjukkan kemungkinan penyelesaian yang lain dari persoalan di atas yaitu tukang batu diisi oleh pelamar P_4, tukang kayu diisi oleh pelamar P_1, tukang listrik diisi oleh pelamar P_2, tukang pipa diisi oleh pelamar P_5, dan tukang cat diisi oleh pelamar P_3.

Dari contoh sederhana di atas terlihat adanya beberapa kemungkinan jawaban dari persoalan ini. Oleh karena itu untuk memperoleh jawaban dari persoalan perlu dilakukan dua hal, yaitu yang pertama menentukan apakah persoalan ini akan mempunyai jawaban dan yang kedua mencari sebuah jawaban dari persoalan. Di bawah ini akan dibahas syarat perlu dan cukup agar persoalan dapat diselesaikan. Sedangkan untuk memperoleh sebuah jawaban akan dibahas pada Bab IV, yaitu dengan menggunakan Algoritma Hungarian.

3.2 Syarat Perlu dan Cukup

Untuk menyelesaikan persoalan di atas perlu dicari syarat perlu dan cukup agar semua persoalan seperti ini dapat ditentukan mempunyai penyelesaian atau tidak mempunyai penyelesaian. Dari ilustrasi graf bipartite seperti pada gambar 3.2 dan gambar 3.3, terlihat secara nyata bahwa syarat perlu agar persoalan ini dapat dipecahkan adalah jumlah pelamar harus lebih besar atau sama dengan banyaknya lowongan pekerjaan ($|Y| \geq |X|$). Syarat cukupnya adalah banyaknya anggota sub himpunan pelamar dengan suatu keahlian harus lebih besar atau sama dengan jumlah anggota sub himpunan lowongan pekerjaan yang sesuai dengan keahlian sub himpunan pelamar bersangkutan.

Syarat perlu dan cukup ini dinyatakan dalam teorema Hall.

3.3 Teorema Hall

Teorema Hall

Misalkan graf G = G(X,Y) adalah graf bipartite, maka akan terdapat suatu *matching lengkap* dari himpunan simpul X ke himpunan simpul Y jika dan hanya jika $|N(S)| \geq |S|$ untuk setiap himpunan bagian $S \subseteq X$.

Catatan: Notasi N(S) adalah himpunan simpul-simpul yang merupakan tetangga dari simpul-simpul di S. Untuk selanjutnya, bila kita membahas graf bagian G_1 atau G_2, maka notasi N(S) di G_1 atau G_2 akan disesuaikan menjadi, masing-masing, $N_1(S)$ untuk graf G_1 dan $N_2(S)$ untuk graf G_2. Pembuktian terdiri dari dua bagian, yang pertama adalah syarat perlu dan yang kedua adalah syarat cukup.

Bukti :

(⇒) Diketahui terdapat *matching lengkap* dari X ke Y, akan dibuktikan $\forall S \subseteq X$ maka $|N(S)| \geq |S|$. Hal ini jelas, karena terdapat *matching lengkap* dari X ke Y, maka bila kita ambil sembarang $S \subseteq X$, maka $\forall x \in S$ akan mempunyai $y \in N(S) \subseteq Y$ yang berbeda. Maka $|N(S)| \geq |S|$.

(⇐) Diketahui $|N(S)| \geq |S|$ $\forall S \subseteq X$, akan dibuktikan terdapat *matching lengkap* dari X ke Y. Secara umum akan dibuktikan menggunakan induksi matematik terhadap jumlah simpul.

Misalkan $|X| = r$ maka:

- Untuk r = 1 hal ini jelas.
- Misalkan benar untuk $|X| < r$.
- Akan dibuktikan benar untuk $|X| = r$.

  Dalam pembuktian ini terdapat dua kasus yang perlu diperhatikan:

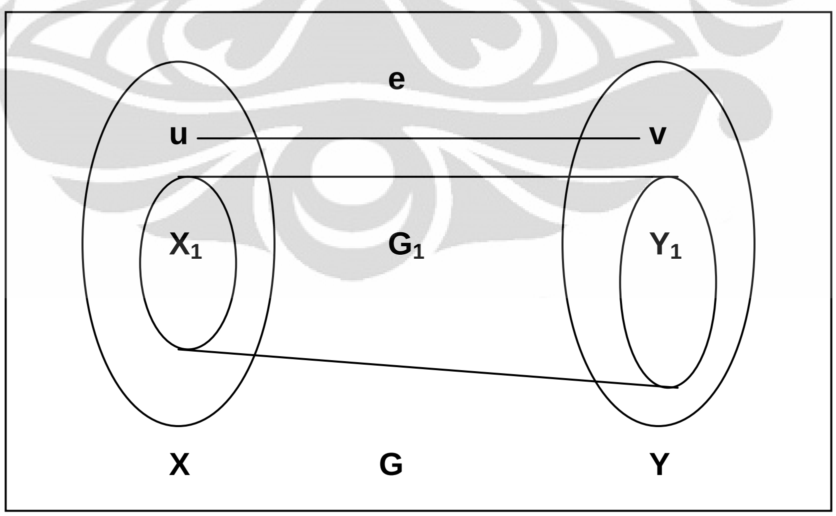
1. $|N(S)| > |S|$ untuk setiap $S \subset X$ dan $S \neq \emptyset$.
2. $|N(X_1)| = |X_1|$ untuk suatu $X_1 \subset X$ dan $X_1 \neq \emptyset$.

Kasus I

Perhatikan gambar 3.4.

Misalkan $|N(S)| > |S|$ untuk setiap $S \subset X$ dan $S \neq \emptyset$. Dan busur

e = uv.



Gambar 3.4

Pada graf G_1, himpunan $V(G_1)=V(G)-\{u,v\}$, $X_1=X-\{u\}$, $Y_1=Y-\{v\}$, dan $G_1(X_1,Y_1)$ adalah graf bipartite.

Pada graf G_1, $|X_1| < r$ dan $N_1(S)=N(S)-\{v\}$, maka $|N_1(S)|=|N(S)|-1 \geq |S|$ atau $|N_1(S)| \geq |S|$ untuk setiap $S \subset X_1$. Oleh karena itu sesuai dengan asumsi induksi bahwa $|X| < r$ dan $|N(S)| \geq |S|$ untuk setiap $S \subseteq X$ pada G_1, maka terdapat *matching lengkap* dari X_1 ke Y_1.

Dengan demikian , apabila pada graf G_1 ditambahkan busur e=uv, kita akan memperoleh *matching lengkap* dari X ke Y untuk graf G.

Kasus II

Perhatikan gambar 3.5.

$|N(X_1)|=|X_1|$ untuk suatu $X_1 \subset X$ dan $X_1 \neq \emptyset$.

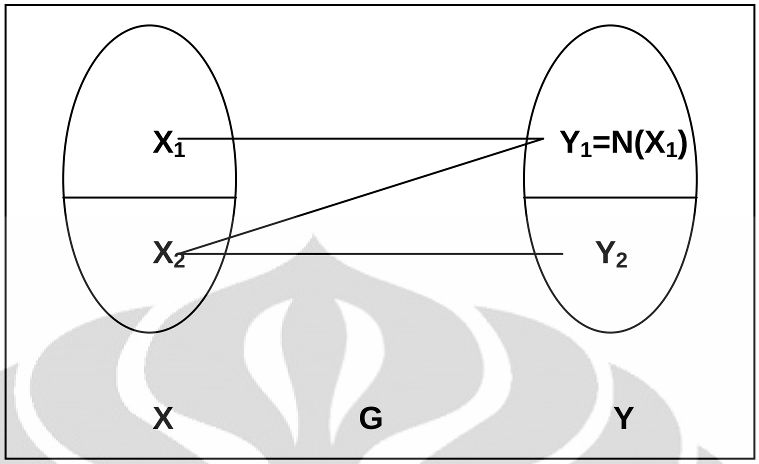
Misalkan $X = X_1 \cup X_2$ dengan $X_1 \cap X_2 = \emptyset$;

$Y_1 = N(X_1)$ dan $Y = Y_1 \cup Y_2$ dengan $Y_1 \cap Y_2 = \emptyset$

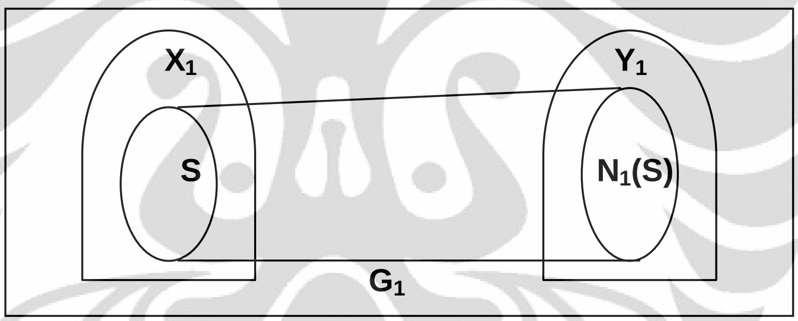
maka $Y_2 \supseteq N(X_2)$ karena ada kemungkinan $x \in X_2$ dan $N(X) \in Y_1$.

Bila G_1 adalah graf yang dibentuk oleh X_1 dan Y_1 yaitu $G_1(X_1,Y_1)$ dan G_2 adalah graf yang dibentuk oleh X_2 dan Y_2 yaitu $G_2(X_2,Y_2)$.

Untuk membuktikan teorema, yang harus dilakukan adalah menunjukkan bahwa pada graf G_1 dan G_2 terdapat *matching lengkap* dari X_1 ke Y_1 untuk graf G_1 dan dari X_2 ke Y_2 untuk graf G_2.



Gambar 3.5



Gambar 3.6

Perhatikan gambar 3.6.

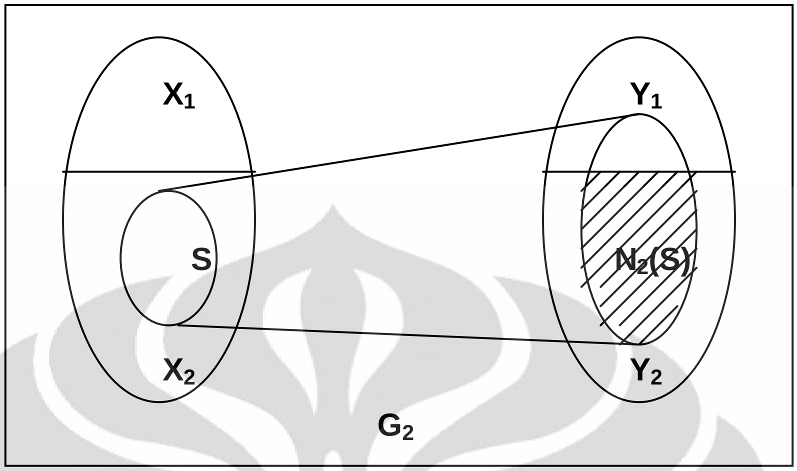
Dalam graf G_1, $|X_1| < r$ dan $X_1 \neq \emptyset$, maka

$$N(S)=N_1(S)\ ,\ \forall S \subseteq X_1 \text{ karena } Y_1=N(X_1)$$

Oleh karena itu , di G_1, $N_1(S)=N(S) \geq |S|$ atau $|N_1(S)| \geq |S|$, $\forall S \subseteq X_1$.

Sesuai dengan asumsi induksi bahwa $|X_1| < r$ dan $|N_1(S)| \geq |S|$,

$\forall S \subseteq X_1$, maka di G_1 terdapat *matching lengkap* dari X_1 ke Y_1.



Gambar 3.7

Perhatikan gambar 3.7.

Dalam graf G_2, $|X_2| < r$ dan $X_2 \neq \emptyset$, maka akan dibuktikan bahwa $|N_2(S)| \geq |S|$ untuk setiap himpunan bagian $S \subseteq X_2$.

Pembuktian ini dengan kontradiksi, misalkan bahwa $|N_2(S)| < |S|$ untuk suatu $S \subseteq X_2$ di G_2. Perhatikan dua himpunan simpul X_1 dan S, $X_1 \cap S = \emptyset$, maka untuk $N(X_1 \cup S)$ di G berlaku hubungan :

- $N(X_1 \cup S) = N(X_1) \cup N(S) = Y_1 \cup N_2(S)$, karena $Y_1 = N(X_1)$
- $|N(X_1 \cup S)| = |Y_1| + |N_2(S)| = |X_1| + |N_2(S)|$, karena $|Y_1| = |N(X_1)| = |X_1|$
- $|N(X_1 \cup S)| < |X_1| + |S|$, karena pemisalan $|N_2(S)| < |S|$
- $|N(X_1 \cup S)| < |X_1 \cup S|$, dimana $X_1 \cap S = \emptyset$.

Hasil akhir ini kontradiksi dengan asumsi teorema yaitu pada graf G berlaku $|N(S)| \geq |S|$ untuk setiap himpunan bagian $S \subseteq X$. Oleh karena itu pemisalan di G_2 bahwa $|N_2(S)| < |S|$ tidak benar, yang berarti $|N_2(S)| \geq |S|$ untuk setiap himpunan bagian $S \subseteq X_2$. Sesuai dengan asumsi induksi bahwa $|X_2| < r$ dan $|N_2(S)| \geq |S|$, $\forall S \subseteq X_2$, maka di G_2 terdapat *matching lengkap* dari X_2 ke Y_2.

Karena untuk G_1 dan G_2 terdapat *matching lengkap*, maka di $G = G_1 \cup G_2$ terdapat *matching lengkap* dari $X = X_1 \cup X_2$ ke $Y = Y_1 \cup Y_2$.

Contoh lain dari *matching lengkap* di atas adalah sebagai berikut: Jika terdapat graf bipartite G(X, Y), dengan busur menghubungkan elemen-elemen di X ke elemen-elemen di Y. Misalkan X = {1, 2, 3} dan Y = {4, 5, 6, 7}. *Matching* pada graf bipartite di halaman berikut adalah sekumpulan busur dimana setiap busur menghubungkan setiap elemen di X ke elemen Y yang berbeda antara yang satu dengan yang lainnya. Dalam contoh di atas, {14, 25, 36) adalah *matching*. *Matching* lain yang dapat terjadi adalah {14, 25, 37}.

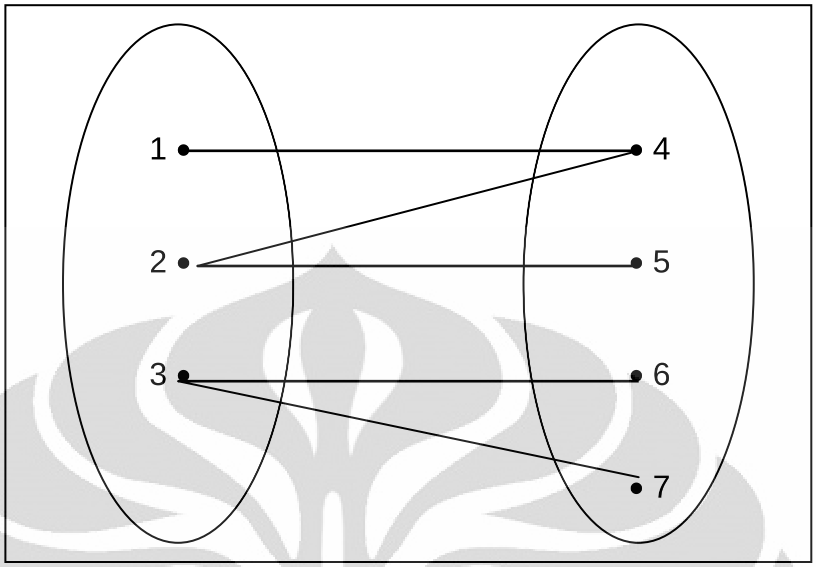
Teorema Hall membuat kondisi dimana *matching* itu ada. Sebagai contoh:

({1}) = {4}, ({2}) = {4, 5}, ({3}) = {6, 7}

({1, 2}) = {4, 5}, ({1, 3}) = {4, 6, 7}, ({2, 3}) = {4, 5, 6, 7}

({1, 2, 3}) = {4, 5, 6, 7}.

Jumlah anggota himpunan tetangga dari dari sembarang subset A di X lebih besar atau sama dengan jumlah anggota himpunan di A, maka kondisi Hall terpenuhi, dan dapat ditarik kesimpulan bahwa *matching* ada dalam contoh tersebut.



Gambar 3.8

Sebaliknya, misalkan busur yang menghubungkan 2 ke 5 dihilangkan. Maka tidak terdapat *matching*, walaupun 1 dan 2 terhubung dengan 4. Kondisi Hall tidak terpenuhi, karena

$(\{1, 2\}) = \{4\}$,

$|(\{1, 2\})| < |\{1, 2\}|$.

BAB IV

METODE HUNGARIAN

Teorema Hall yang telah dibahas pada bab III dapat dipergunakan pula pada berbagai macam aplikasi. Jenis aplikasi yang dapat menggunakan teorema Hall ini biasanya berbentuk memasangkan atau mengawinkan dua himpunan yang berbeda. Anggota himpunan pertama dipasangkan dengan anggota dari himpunan kedua sedemikian sehingga setiap anggota dari himpunan pertama akan habis dipasangkan dengan anggota yang berbeda dari himpunan kedua. Ini dikenal dengan nama *matching lengkap*. Untuk dapat menyelesaikan suatu persoalan *matching lengkap* diperlukan suatu algoritma.

Ada beberapa algoritma yang dapat digunakan untuk memecahkan masalah di atas. Salah satunya adalah metode Hungarian. Metode ini akan memberikan algoritma yang menyelesaikan persoalan ada atau tidaknya *matching lengkap*.

Secara umum, algoritma ini terdiri dari dua bagian, yaitu:

1. membangun dahulu suatu pohon yang disebut dengan pohon M-*alternating*, dan
2. mencari penyelesaiannya.

Algoritma Hungarian akan mencari lintasan-lintasan *augmenting*. Lintasan M-*augmenting* adalah lintasan M-*alternating* di mana simpul awal dan akhirnya merupakan anggota M-*unmatched*-simpul v dari G yang tidak ada dalam busur M-sedangkan lintasan M-*alternating*, busurnya bisa ada dalam E(G)-M dan M. Untuk mencari lintasan M-*augmenting* pada graf G, algoritma ini membangun pohon M-*alternating*. Setelah pohon M-*alternating* dibentuk, barulah algoritma Hungarian mencari jawab persoalan *matching lengkap*. Kedua proses ini dituliskan dalam satu algoritma yaitu algoritma Hungarian.

Pohon M-*alternating* (T) adalah pohon yang memiliki sifat-sifat sebagai berikut:

1. Akar r dari T adalah himpunan simpul yang belum mempunyai pasangan (M-*unmatched*) dari himpunan simpul X
2. Untuk setiap bilangan ganjil i yang lebih kecil dari kedalaman (*depth*) pohon T, setiap busur di T yang menghubungkan suatu simpul pada tingkat (*level*) ke i ke suatu simpul pada tingkat ke i+1 merupakan lintasan yang menghubungkan simpul yang matching (*matched*)
3. Simpul daun dari T adalah anggota X

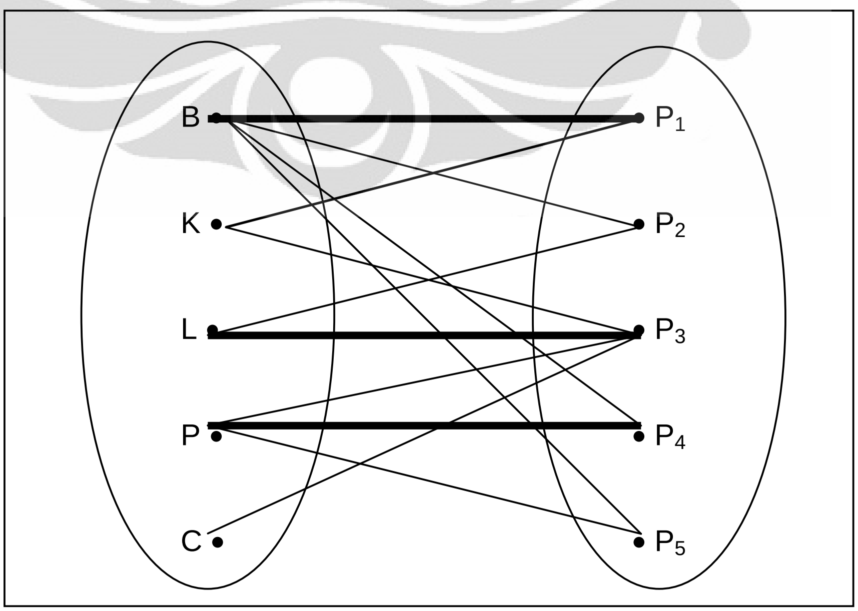
ALGORITMA HUNGARIAN

```
Procedure Hungarian(G,M,S)
Input:    G (graf bipartite yang simpulnya telah dipartisi (X,Y)
          M (matching awal, mungkin  juga kosong)
Output: M (matching lengkap  jika ada satu)
          S (sekumpulan simpul yang memenuhi kondisi |N(S)|<|S|
          jika tidak ada matching lengkap)

AugmentingM:=.true.
while AugmentingM do
  if semua simpul di X ada dalam M then {M adalah matching lengkap}
     AugmentingM:=.false.
  else                                    {buat pohon T M-alternating}
     r:= sembarang M yang belum mempunyai pasangan di X
     T:= pohon yang terdiri dari satu simpul r
     GrowingTree:=.true.
     while  GrowingTree  do
        if N(X_T) = Y_T then
```

```
        S:= X_T                                          {|N(S)|<|S|)
        GrowingTree:=.false.
        AugmentingM:=.false.
    else
        y:= sembarang simpul di N(X_T)-Y_T
        x:= sembarang simpul di X_T yang bertetangga  dengan  y
        if y ada dalam M then                              {T augment}
           z:= M(y)                      {z adalah pasangan y dalam M}
           T:= T ∪ xy ∪ yz
        else                            {didapat lintasan M-augmenting}
           P:= lintasan dalam T dari r ke x bersama dengan
                busur xy
           M= M ⊕ E(P)                                     {M augment}
           GrowingTree:=.false.
        endif
    endif
  endwhile
 endif
endwhile
end Hungarian


Catatan: M ⊕ E(P) = (M ∪ E(P)) - (M ∩ E(P)).
```

Algoritma Hungarian

Contoh:



Gambar 4.1

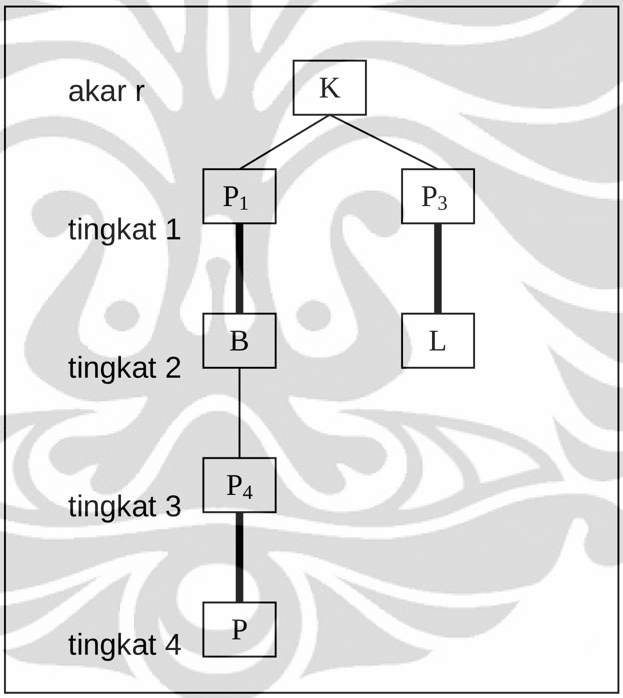
Bila algoritma Hungarian diterapkan pada graf bipartite gambar 4.1 maka:

Iterasi 1: *Matching* awal = (BP_1, LP_3, PP_4)

Himpunan M-*matched* = (B, L, P)

Himpunan M-*unmatched* = (K, C)

Karena tidak semua simpul di X *matching* maka perlu dibuat pohon M-*alternating.* Tentukan r dari himpunan M-*unmatched*. Dalam contoh di atas, r = K dapat dilihat pada gambar 4.2.



Gambar 4.2

Himpunan X_T = (B, K, L, P)

Himpunan Y_T = (P_1, P_3, P_4)

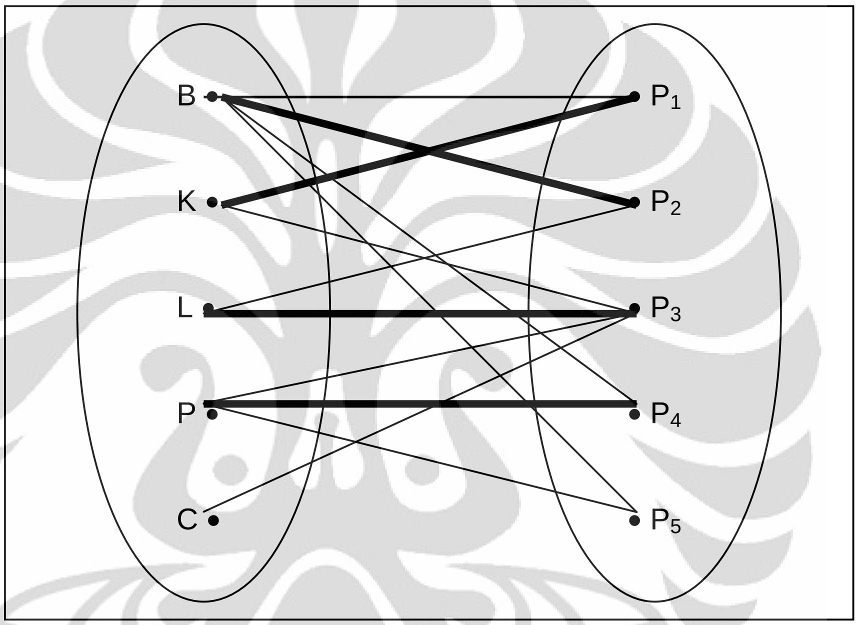
$N(X_T) \neq Y_T$. Kemudian tentukan x dan y.

Dari pohon M-*alternating* gambar 4.2 didapat y = P_2, x = B.

Karena y tidak termasuk dalam M-*matched*, maka lintasan

pertama M-*augmenting*-nya adalah P = K P_1 B P_2.

M = M $\oplus$ E(P) = {BP_2, KP_1, LP_3, PP_4} sepeti yang ditunjukkan pada gambar 4.3.



Gambar 4.3

Graf di atas belum *matching lengkap*.

Iterasi 2: *Matching* awal {BP_2, KP_1, LP_3, PP_4}

Himpunan M-*matched* = (B, K, L, P)

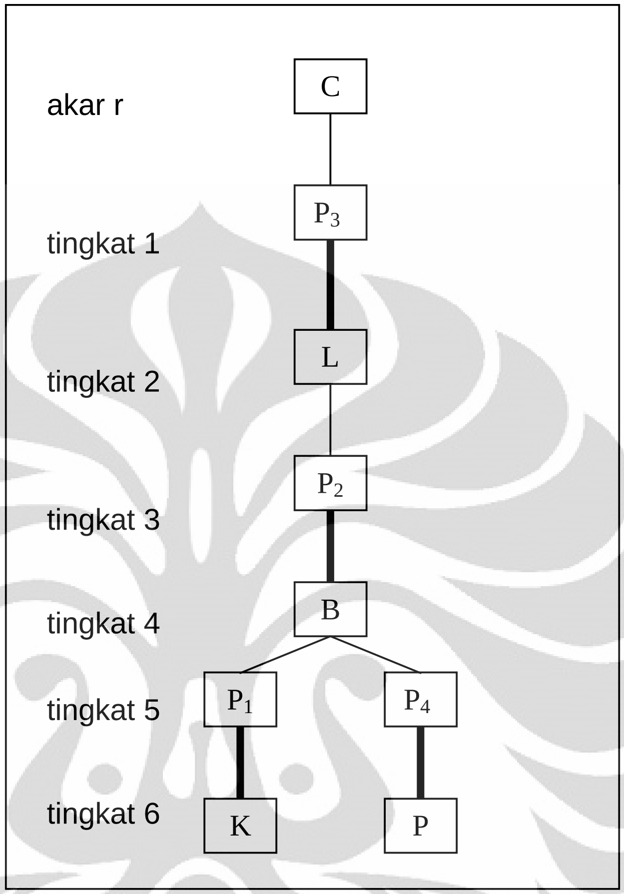
Himpunan M-*unmatched* = (C)

Kemudian kembali lagi mencari pohon M-*alternating* yang baru

seperti yang ditunjukkan pada gambar 4.4.

Tentukan r dari himpunan M-*unmatched*

Didapat r = C.



Gambar 4.4

Himpunan X_T = (B, K, L, P, C)

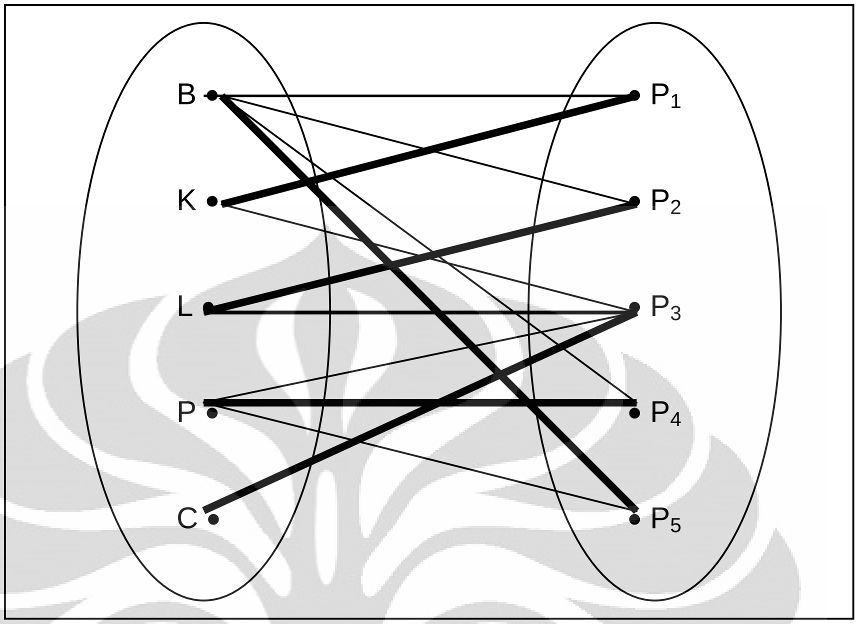
Himpunan Y_T = (P_1, P_2, P_3, P_4)

$N(X_T) \neq Y_T$. Kemudian tentukan x dan y.

Dari pohon M-*alternating* gambar 4.4 didapat y = P_5, x = B.

Karena y tidak termasuk dalam M-*matched*, maka lintasan kedua M-*augmenting*-nya adalah P = C P_3 L P_2 B P_5.

M = M $\oplus$ E(P) = {BP_5, KP_1, LP_2, PP_4, CP_3} seperti yang ditunjukkan pada gambar 4.5 pada halaman berikut.



Gambar 4.5

Graf di atas telah *matching lengkap*.

Maka algoritma Hungarian telah selesai mencari *matching lengkap*.

BAB V

PENUTUP

Dari hasil pembahasan teorema Hall serta algoritma Hungarian pada bab sebelumnya, maka semua persoalan yang dapat dimodelkan menjadi *matching lengkap* dapat diselesaikan. Yang perlu diingat dalam menyelesaikan persoalan ini adalah dipenuhinya syarat perlu dan cukup. Setelah itu algoritma Hungarian dapat digunakan untuk mencari satu dari beberapa kemungkinan penyelesaian.

Di samping algoritma Hungarian, ada lagi beberapa algoritma seperti algoritma Murty, algoritma Chegireddy dan Hamacher. Algoritma-algoritma ini bukan hanya menyelesaikan persoalan *matching lengkap* sederhana, akan tetapi telah berhasil menyelesaikan persoalan *matching lengkap* yang sudah diubah. Perubahan dimaksud adalah pada persoalan *matching lengkap* di mana setiap busurnya telah diberikan suatu bobot dan penyelesaian yang dicari bukan hanya *matching lengkap* saja, tetapi *matching lengkap* yang optimal sesuai dengan kriteria optimal dari persoalan. Juga algoritma-algoritma tersebut, seperti disebutkan dalam referensi, bilamana digunakan hanya untuk mencari *matching lengkap* (semua bobot di busur diabaikan) kalah efisien dibandingkan dengan algoritma Hungarian.

Melalui pembahasan teorema Hall dan algoritma Hungarian, diharapkan skripsi ini dapat berguna dan diterapkan dalam menjawab persoalan-persoalan sejenis. Karena pembahasan pada penulisan ini contoh metode yang digunakan hanya Metode Hungarian, untuk penulisan berikut agar kiranya pembahasan dapat diperluas pada metode-metode lain yang dapat mengungkap adanya *matching lengkap* pada model graf bipartite dan juga dibuat program komputernya. Sehingga dapat dibandingkan secara numerik setiap metode yang ada dan diperoleh kesimpulan yang lebih baik.

DAFTAR PUSTAKA

1. Berman, Kenneth A. & Paul, Jerome L. 1997. *Fundamentals of Sequential and Parallel Algorithms*. International Thomson Publishing (ITP). Boston: 672-679.
2. Bradenburg. 2001. *Matching on Bipartite Graphs*, Proseminar Algorithmen,4 hlm.http://www.infosun.fmi.unipassau.de/br/lehrstuhl/kurse/proseminar_ss01.html, 10 April 2001.
3. Slamet, Suryadi. 1998. *Pengantar Teori Graf*, Jurusan Matematika FMIPA-UI, Depok: iii + 108 hlm.
4. Wilson, Robin J.1996. *Introduction to Graph Theory*, Longman, England: vii + 171 hlm.