



**PENERAPAN ALGORITMA KUHN-MUNKRES  
DALAM PENYELESAIAN MASALAH PENUGASAN  
*MULTI-OBJECTIVE***

Skripsi

disajikan sebagai salah satu syarat  
untuk memperoleh gelar Sarjana Sains

Program Studi Matematika

oleh

Tiara Budi Utami

4111413034

**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS NEGERI SEMARANG**

**2017**



**UNNES**  
UNIVERSITAS NEGERI SEMARANG

## PERNYATAAN

Saya menyatakan bahwa skripsi ini bebas plagiat, kecuali yang secara tertulis dirujuk dalam skripsi ini dan disebutkan dalam daftar pustaka. Apabila di kemudian hari terbukti terdapat plagiat dalam skripsi ini, maka saya bersedia menerima sanksi sesuai ketentuan peraturan perundang-undangan.

Semarang, 9 Agustus 2017



Tiara Budi Utami  
4111413034

**UNNES**  
UNIVERSITAS NEGERI SEMARANG

## PENGESAHAN

Skripsi yang berjudul

Penerapan Algoritma Kuhn-Munkres dalam Penyelesaian Masalah

Penugasan *Multi-objective*

disusun oleh

Tiara Budi Utami

4111413034

telah dipertahankan di hadapan sidang Panitia Ujian Skripsi FMIPA UNNES pada tanggal 9 Agustus 2017.



Prof. Dr. Zaenuri, S.E., M.Si, Akt.  
NIP. 196412231988031001

Sekretaris

Drs. Arief Agoestanto, M.Si.  
NIP. 196807221993031005

Ketua Penguji

Drs. Edy Soedjoko, M.Pd.  
NIP. 195604191987031001

Anggota Penguji/  
Pembimbing I

Dr. Isnaini/Rosyida, S.Si., M.Si.  
NIP. 197302191998022001

Anggota Penguji/  
Pembimbing II

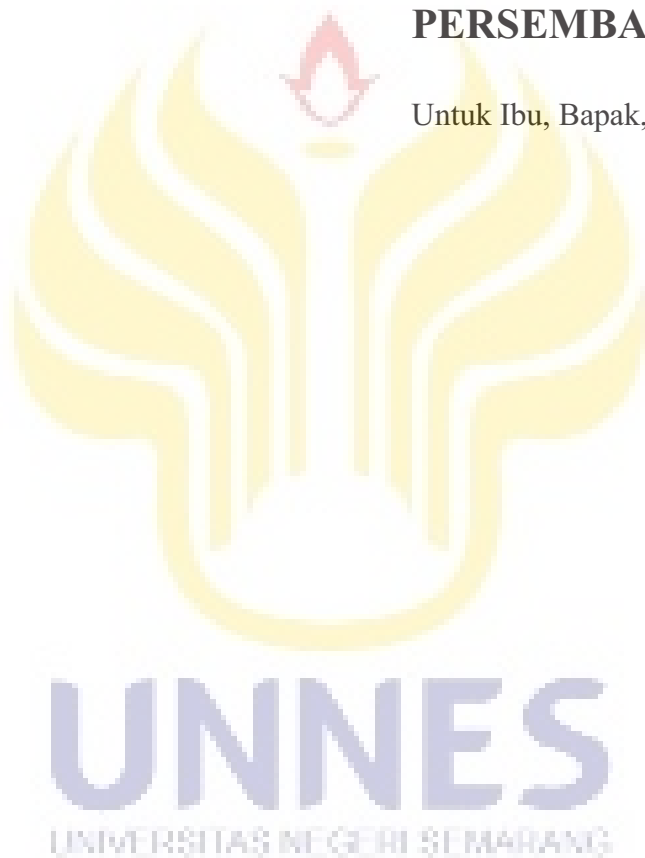
Dr. Mulyono, M.Si.  
NIP. 197009021997021001

## **MOTTO**

Happiness can be found even in the darkest of times if one only remembers to turn on the light (J.K. Rowling)

## **PERSEMBAHAN**

Untuk Ibu, Bapak, Erick



## KATA PENGANTAR

Puji syukur senantiasa penulis panjatkan kehadirat Allah Swt. yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Penerapan Algoritma Kuhn-Munkres dalam Penyelesaian Masalah Penugasan *Multi-objective*”.

Penulisan skripsi ini dapat terselesaikan semata-mata karena kekuatan dari Allah melalui bimbingan, bantuan, dan dukungan dari berbagai pihak baik secara langsung maupun tidak langsung. Untuk itu, pada kesempatan ini penulis ingin menyampaikan terimakasih kepada:

1. Prof. Dr. Fathur Rokhman, M.Hum., Rektor Universitas Negeri Semarang.
2. Prof. Dr. Zaenuri Mastur, S.E., M.Si., Akt., Dekan FMIPA Universitas Negeri Semarang.
3. Drs. Arief Agoestanto, M.Si., Ketua Jurusan Matematika FMIPA Universitas Negeri Semarang.
4. Drs. Mashuri, M.Si., Koordinator Program Studi Matematika FMIPA Universitas Negeri Semarang.
5. Dr. Isnaini Rosyida, S.Si., M.Si., selaku Dosen Pembimbing I yang telah memberikan bimbingan dan dukungan selama penyusunan skripsi ini.
6. Dr. Mulyono, M.Si., selaku Dosen Pembimbing II yang telah memberikan bimbingan dan dukungan selama penyusunan skripsi ini.
7. Drs. Edy Soedjoko, M.Pd., selaku Dosen Penguji yang telah memberikan penilaian dan saran dalam perbaikan skripsi ini.



8. Drs. Sugiman, M.Si., selaku Dosen Wali yang telah memberikan bimbingan dan arahan dari segi akademik dan kemahasiswaan.
9. Seluruh Dosen Jurusan Matematika Universitas Negeri Semarang yang telah mendidik dan menyampaikan ilmu yang bermanfaat kepada penulis.
10. Ibu, Bapak, dan Adik tercinta yang selalu memberikan dukungan dan doa yang tiada putusnya.
11. Bidikmisi Universitas Negeri Semarang yang telah memberikan dukungan secara materiil maupun non-materiil.
12. Sahabat-sahabatku dari Kudus, Kos Wanodyatama, Rombel Matematika 2013, Himatika 2014/2015, dan MEC 2014/2015 yang telah memberikan bantuan, semangat, dan keceriaan kepada penulis.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat kekurangan, untuk itu penulis mengharapkan kritik dan saran yang bersifat membangun demi perbaikan di masa yang akan datang.

**UNNES**  
UNIVERSITAS NEGERI SEMARANG

Semarang, 9 Agustus 2017

Penulis

## ABSTRAK

Utami, Tiara Budi. 2017. *Penerapan Algoritma Kuhn-Munkres dalam Penyelesaian Masalah Penugasan Multi-objective*. Skripsi, Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang. Pembimbing I: Dr. Isnaini Rosyida, S.Si., M.Si. dan Pembimbing II: Dr. Mulyono, M.Si.

Kata Kunci: *Matching*, Algoritma Kuhn-Munkres, penugasan *multi-objective*.

Masalah umum penugasan meliputi  $x$  tugas yang harus ditetapkan kepada  $y$  pekerja di mana setiap pekerja memiliki kompetensi yang berbeda dalam menyelesaikan setiap tugas. Beberapa penelitian telah dikembangkan untuk memecahkan masalah penugasan. Sebagian besar metode yang dikembangkan hanya mempertimbangkan satu tujuan pengoptimalan. Kebanyakan kasus penugasan yang mungkin diinginkan tidak ditentukan hanya berdasar pada satu pertimbangan. Dibutuhkan optimasi secara serempak dari beberapa kriteria sehingga diperoleh penugasan yang sesuai. Masalah penugasan yang mempunyai beberapa tujuan pengoptimalan terhadap beberapa jenis sumber daya yang dimiliki oleh pekerja dalam menyelesaikan tugas disebut sebagai masalah penugasan *multi-objective*.

Pada penelitian ini, diberikan suatu prosedur untuk menyelesaikan masalah penugasan *multi-objective* dengan menggunakan Algoritma Kuhn-Munkres. Algoritma Kuhn-Munkres hanya dapat diterapkan pada kasus masalah penugasan sederhana (*single-objective*), sedangkan pada masalah penugasan *multi-objective* membutuhkan optimasi secara serempak dari beberapa tujuan pengoptimalan (misal: biaya, waktu, dan kualitas). Untuk menyelesaikan masalah tersebut, dilakukan transformasi data *multi-objective* ke dalam bentuk *single-objective* dengan menggunakan pendekatan vektor bobot. Kemudian, prosedur yang diusulkan diilustrasikan dengan contoh numerik pada kasus nyata.

Hasil perhitungan dengan Algoritma Kuhn-Munkres diperoleh solusi masalah penugasan optimal berupa penjadwalan sempurna dengan bobot maksimum pada graf bipartisi komplit berbobot. Selain menggunakan Algoritma Kuhn-Munkres, dilakukan pencarian pasangan penugasan optimal menggunakan Solver. Hasil perhitungan dengan Solver memberikan pasangan penugasan yang sama dengan hasil perhitungan dengan Algoritma Kuhn-Munkres.



# DAFTAR ISI

	Halaman
HALAMAN JUDUL .....	i
PERNYATAAN .....	iii
PENGESAHAN.....	iv
MOTTO DAN PERSEMBAHAN.....	v
KATA PENGANTAR.....	vi
ABSTRAK.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL .....	xiii
DAFTAR GAMBAR .....	xiv
DAFTAR LAMPIRAN.....	xvi
BAB I PENDAHULUAN	
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	4
1.3 Batasan Masalah .....	5
1.4 Tujuan Penelitian.....	5
1.5 Manfaat Penelitian .....	6
1.6 Sistematika Penulisan .....	6
BAB II TINJAUAN PUSTAKA	
2.1 Masalah Penugasan ( <i>Assignment Problem</i> ).....	8
2.1.1 Masalah Umum Penugasan.....	8

2.1.2	Masalah Penugasan Sederhana ( <i>Single-objective</i> ).....	9
2.2	Teori Graf .....	10
2.2.1	Definisi Graf.....	11
2.2.2	Terminologi Dasar Graf .....	11
2.2.2.1	Bertetangga ( <i>Adjacent</i> ) dan Terkait ( <i>Incident</i> ).....	12
2.2.2.2	Gelung, Sisi Rangkap, dan Graf Sederhana .....	12
2.2.2.3	Derajat ( <i>Degree</i> ).....	13
2.2.2.4	Titik Terasing dan Titik Ujung.....	13
2.2.2.5	Jalan, Jejak, Lintasan, Sirkuit, dan Sikel.....	14
2.2.3	Graf Bagian ( <i>Subgraph</i> ) .....	15
2.2.4	Graf Terhubung ( <i>Connected</i> ) dan Komponen Graf .....	15
2.2.5	Graf Komplit.....	16
2.2.6	Graf Beraturan- $k$ ( $k$ -Regular) .....	17
2.2.7	Graf Berbobot.....	17
2.2.8	Graf Bipartisi .....	18
2.2.9	Pohon ( <i>Tree</i> ).....	19
2.2.10	Representasi Graf dalam Matriks .....	19
2.3	Penjodohan ( <i>Matching</i> ) .....	21
2.3.1	Definisi Penjodohan ( <i>Matching</i> ).....	21
2.3.2	Tertutup oleh Penjodohan ( <i>Matching-Saturated</i> ).....	22
2.3.3	Lintasan Alternatif dan Lintasan Augmentasi .....	22
2.3.4	Pohon Alternatif.....	23
2.3.5	Penjodohan Maksimum .....	24

2.3.6	Penjodohan Sempurna (Perfect Matching) .....	26
2.3.7	Penjodohan Bobot Maksimum.....	27
2.3.8	Penjodohan pada Graf Bipartisi .....	28
2.4	Aplikasi Penjodohan .....	30
2.4.1	Aplikasi Penjodohan pada Masalah Penugasan Personal .....	30
2.4.2	Aplikasi Penjodohan pada Masalah Penugasan Optimal .....	31
2.5	Algoritma Kuhn-Munkres .....	34
2.5.1	Deskripsi Algoritma Kuhn-Munkres .....	34
2.5.2	Algoritma Kuhn-Munkres .....	34

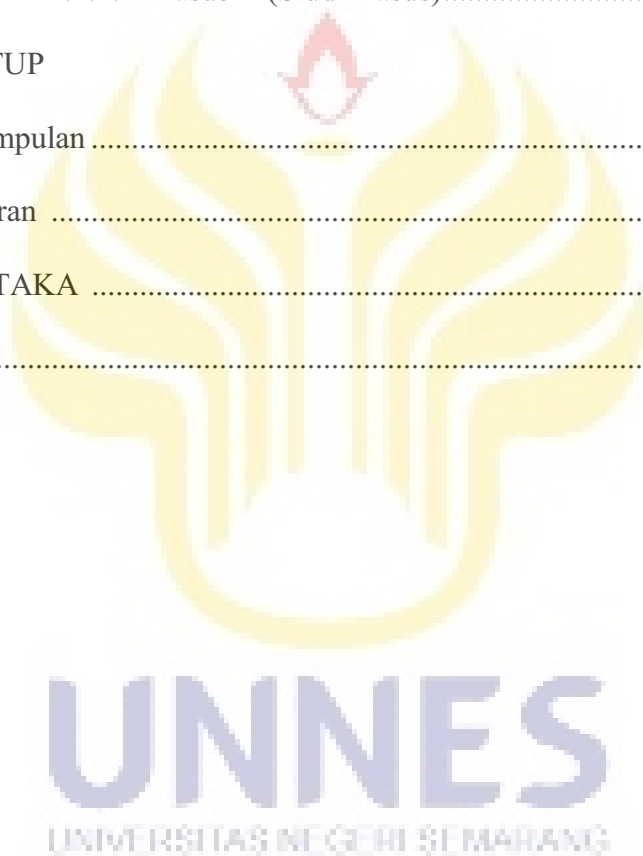
### BAB III METODE PENELITIAN

3.1	Identifikasi Masalah .....	44
3.1.1	Penemuan Masalah.....	44
3.1.2	Perumusan Masalah.....	44
3.2	Persiapan Penelitian.....	44
3.2.1	Studi Pustaka .....	44
3.2.2	Perancangan .....	45
3.3	Analisis dan Pemecahan Masalah.....	45
3.4	Penarikan Simpulan.....	46

### BAB IV HASIL DAN PEMBAHASAN

4.1	Masalah Penugasan <i>Multi-objective</i> .....	47
4.1.1	Definisi Masalah Penugasan <i>Multi-objective</i> .....	47
4.1.2	Model Masalah Penugasan <i>Multi-objective</i> .....	47

4.2 Penyelesaian Masalah Penugasan <i>Multi-objective</i> dengan Algoritma Kuhn-Munkres .....	49
4.2.1 Langkah-langkah Penyelesaian.....	49
4.2.2 Contoh Kasus.....	52
4.2.2.1 Kasus I.....	52
4.2.2.2 Kasus II (Studi Kasus).....	67
BAB V PENUTUP	
5.1 Simpulan.....	87
5.2 Saran .....	90
DAFTAR PUSTAKA .....	91
LAMPIRAN.....	94



## DAFTAR TABEL

Tabel	Halaman
Tabel 2.1. Masalah penugasan sederhana.....	9
Tabel 4.1. Riwayat kinerja operator I.....	53
Tabel 4.2. Riwayat kinerja operator II (rekonstruksi).....	55
Tabel 4.3. Riwayat kinerja operator III (normalisasi).....	57
Tabel 4.4. Riwayat kinerja operator IV ( <i>single-objective</i> ).....	59
Tabel 4.5. Hasil Perhitungan dengan Solver (Kasus I).....	66
Tabel 4.6. Data riwayat produksi I.....	68
Tabel 4.7. Data riwayat produksi II (rekonstruksi).....	70
Tabel 4.8. Data riwayat produksi III (normalisasi).....	72
Tabel 4.9. Data riwayat produksi IV ( <i>single-objective</i> ).....	74
Tabel 4.10. Hasil Perhitungan dengan Solver (Kasus II).....	86

## DAFTAR GAMBAR

Gambar	Halaman
Gambar 2.1. (a) Graf $G$ dengan 4 titik dan 5 sisi;	
(b) graf $H$ dengan 6 titik dan 8 sisi .....	12
Gambar 2.2. Graf $G$ : jalan, jejak, lintasan, sirkuit, dan sikel .....	14
Gambar 2.3. $H_1$ graf bagian $H$ dan $H_2$ graf bagian perentang .....	15
Gambar 2.4. (a) Graf terhubung $G$ ;	
(b) graf tidak terhubung $H$ dengan 3 komponen $G_1, G_2, G_3$ .....	16
Gambar 2.5. $K_5$ graf komplit dengan 5 titik .....	17
Gambar 2.6. Graf $G$ beraturan $-3, \delta(G) = 3 = \Delta(G)$ .....	17
Gambar 2.7. Graf berbobot $G$ dengan $w(G) = 12$ .....	18
Gambar 2.8. (a) Graf bipartisi berbobot $B_1$ ; (b) graf bipartisi komplit $B_2$ .....	18
Gambar 2.9. Pohon $T$ .....	19
Gambar 2.10. Graf $G$ dengan titik-titik yang bertetangga dan terkait .....	20
Gambar 2.11. Graf dasar $G$ (a); (b) graf $G$ dengan penjodohan $M_1$ .....	21
Gambar 2.12. Graf $G$ dengan Penjodohan $M_3$ .....	22
Gambar 2.13. Graf $G$ dengan Penjodohan $M$ .....	23
Gambar 2.14. (a) Graf pohon $T$ dengan penjodohan $M$ ;	
(b) $T_1$ dan $T_2$ pohon alternatif dari graf pohon $T$ .....	24
Gambar 2.15. Penjodohan maksimum di graf $G$ .....	24
Gambar 2.16. Penjodohan sempurna di graf $G$ .....	27
Gambar 2.17. Graf Berbobot $H$ dengan Penjodohan $M$ .....	27



Gambar 2.18. Graf bipartisi $G$ yang memenuhi $ N(S)  \geq  S $ .....	29
Gambar 2.19. Diagram Alir Algoritma Kuhn-Munkres .....	35
Gambar 3.1. Diagram Alir Penelitian .....	43



## DAFTAR LAMPIRAN

Lampiran	Halaman
1. Ekspansi persamaan (4.10).....	94
2. Ekspansi persamaan (4.14).....	96
3. Ekspansi persamaan (4.15).....	99
4. Ekspansi persamaan (4.19).....	101



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Karakteristik dunia usaha saat ini ditandai oleh pesatnya perkembangan di segala bidang yang menuntut kepiawaian manajemen dalam mengantisipasi setiap perubahan. Dalam pengambilan keputusan yang memiliki risiko besar, intuisi naluriah saja belum cukup untuk dijadikan pertimbangan. Oleh karena itu, perlu dilakukan pengoptimalan dalam pengambilan keputusan agar risiko kerugian dapat diminimalkan.

Masalah di kehidupan sehari-hari yang berkaitan dengan pengambilan keputusan salah satunya adalah masalah penugasan (*assignment problem*). Masalah umum penugasan meliputi  $x$  tugas yang harus ditetapkan kepada  $y$  pekerja di mana setiap pekerja memiliki kompetensi yang berbeda dalam menyelesaikan setiap tugas. Tujuan dari masalah penugasan adalah untuk menetapkan tugas yang sesuai pada pekerja sehingga total pengeluaran sumber daya untuk menyelesaikan semua tugas dapat dioptimalkan.

Beberapa penelitian telah dikembangkan untuk memecahkan masalah penugasan. Sebagian besar metode yang dikembangkan hanya mempertimbangkan satu tujuan pengoptimalan. Sebagai contoh, masalah penugasan dengan biaya minimum atau masalah penugasan dengan waktu penyelesaian minimum. Meminimumkan biaya pada masalah penugasan

terfokus pada bagaimana memberikan tugas kepada pekerja sehingga total biaya operasi yang dikeluarkan minimum, begitu juga dalam meminimumkan waktu penyelesaian hanya terfokus pada bagaimana memberikan tugas kepada pekerja dengan total waktu operasi minimum. Masalah penugasan ini yang disebut sebagai masalah penugasan sederhana (*single-objective*).

Kebanyakan kasus penugasan yang mungkin diinginkan tidak ditentukan hanya berdasar pada satu pertimbangan. Dibutuhkan optimasi secara serempak dari beberapa kriteria sehingga diperoleh penugasan yang sesuai. Dengan kata lain, penugasan harus memenuhi lebih dari satu tujuan pengoptimalan. Masalah penugasan yang mempunyai beberapa tujuan pengoptimalan terhadap beberapa jenis sumber daya yang dimiliki oleh pekerja dalam menyelesaikan tugas, baik berupa kuantitatif maupun kualitatif disebut sebagai masalah penugasan *multi-objective* (Przybylski *et al.*, 2009:1).

Penelitian sebelumnya mencoba menyelesaikan masalah penugasan *multi-objective* yang terkait dengan biaya dan waktu (Geetha *et al.*, 1999). Selanjutnya, Bao *et al.* (2007) memberikan pendekatan baru dalam menyelesaikan masalah penugasan *multi-objective* dengan metode pemrograman 0-1 di mana masalah penugasan *multi-objective* diterjemahkan menjadi masalah linier. Garrett *et al.* (2007) menyelesaikan masalah penugasan *multi-objective* pada pelaut (*Sailor Assignment Problem/ SAP*) yang dikenai kondisi bahwa banyaknya pelaut dan tugas tidak sama, yaitu

dengan Algoritma Hibrida. Metode lain untuk menyelesaikan masalah penugasan *multi-objective* adalah dengan metode dua fase (Przybylski *et al.*, 2010) dan Algoritma Branch & Bound (Belhoual *et al.*, 2014) di mana dalam metode ini seluruh kemungkinan solusi diperhitungkan sebagai kandidat solusi. Selain metode-metode yang telah disebutkan di atas, terdapat beberapa pendekatan pada kasus optimasi *multi-objective* yang dapat diterapkan untuk masalah penugasan, di antaranya dengan Algoritma Genetika, Pendekatan Pareto, dan Algoritma Evolusioner *Multi-Objective*. Metode-metode yang bekerja dengan Algoritma Genetika antara lain: metode penyortiran tanpa pengaruh pada Algoritma Genetika (*Non-dominated Sorting in Genetic Algorithm/ NSGA*) (Srinivas & Deb, 1994), metode penyortiran cepat tanpa pengaruh pada Algoritma Genetika (*Fast Non-dominated Sorting in Genetic Algorithm/ NSGA-II*) (Deb *et al.*, 2002) dan Algoritma Genetika berbasis peringkat kepadatan (*Rank-Density-Based Genetic Algorithm/ RDGA*) (Lu & Yen, 2003). Metode-metode yang bekerja dengan Pendekatan Pareto adalah Algoritma Evolusioner Pareto Berdaya (*Strength Pareto Evolutioner Algorithm/ SPEA*) (Zitzler & Thiele, 1999) dan Strategi Evolusi Terarsip Pareto (*Pareto Archived Evolution Strategy/ PAES*) (Knowles & Corne, 2000). Sedangkan, metode yang bekerja dengan Algoritma Evolusioner *Multi-Objective* antara lain: Algoritma Evolusioner *Multi-Objective* (*Multi-Objective Evolutionary Algorithm/ MOEA*) (Sarker *et al.*, 2002) dan Algoritma Evolusioner *Multi-Objective* Dinamik (*Dinamic Multi-Objective Evolutionary Algorithm/ DMOEA*) (Yen & Lu, 2003).

Dalam skripsi ini, penulis mengkaji masalah penugasan *multi-objective* dengan kondisi banyaknya pekerja sama dengan banyaknya tugas.

Teori graf adalah cabang dari matematika yang dapat diterapkan dalam penyelesaian masalah penugasan optimal. Masalah penugasan dinyatakan dengan graf bipartisi komplit berbobot, kemudian dicari solusi optimalnya berupa penjadohan sempurna dengan bobot maksimum (Mills-Tettey *et al.*, 2007). Algoritma yang digunakan untuk menemukan penjadohan sempurna dengan bobot maksimum adalah Algoritma Kuhn-Munkres yang dikembangkan oleh Kuhn (1955) dan Munkres (1957).

Selama ini Algoritma Kuhn-Munkres hanya diterapkan pada kasus masalah penugasan sederhana (*single-objective*). Pada penelitian ini penulis mencoba menerapkan Algoritma Kuhn-Munkres dalam menentukan solusi optimal dari contoh kasus masalah penugasan *multi-objective*, sehingga melalui langkah penyelesaian yang diusulkan dapat memberikan hasil penugasan yang optimal.

## 1.2 Rumusan Masalah

Berdasarkan kondisi yang muncul pada latar belakang, rumusan masalah yang akan dibahas dalam penelitian ini adalah sebagai berikut:

1. Bagaimana model matematis masalah penugasan *multi-objective*?
2. Bagaimana menyelesaikan masalah penugasan *multi-objective* dengan Algoritma Kuhn-Munkres?



### 1.3 Batasan Masalah

Pada penelitian ini, pembahasan masalah dibatasi oleh hal-hal sebagai berikut:

1. Masalah penugasan *multi-objective* yang diteliti mempunyai tujuan optimasi yang sama.
2. Masalah penugasan *multi-objective* yang diteliti dikenai kondisi banyaknya pekerja sama dengan banyaknya tugas.
3. Asumsi bobot tujuan dari masing-masing tujuan yang akan dioptimalkan adalah sama.
4. Metode yang digunakan dalam penyelesaian masalah penugasan optimal yang direpresentasikan dalam graf bipartisi komplit berbobot adalah dengan Algoritma Kuhn-Munkres.

### 1.4 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka tujuan dari penelitian ini adalah:

1. Dapat mengetahui model matematis dari penugasan *multi-objective*.
2. Dapat menyelesaikan masalah penugasan *multi-objective* dengan Algoritma Kuhn-Munkres untuk memperoleh solusi optimal.

## 1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat berupa pengetahuan tentang metode penyelesaian masalah penugasan *multi-objective* dengan Algoritma Kuhn-Munkres. Selain itu, solusi yang diperoleh diharapkan dapat menyelesaikan permasalahan dalam kehidupan nyata dan melalui algoritma yang diterapkan dapat menambah referensi tentang aplikasi teori graf dalam hal manajemen dan pengambilan keputusan.

## 1.6 Sistematika Penulisan

Secara garis besar sistematika penulisan skripsi ini dibagi menjadi beberapa bab, yaitu:

### BAB I PENDAHULUAN

Bab ini memuat latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

### BAB II TINJAUAN PUSTAKA

Berisi kajian teoritik dari sumber pustaka cetak dan elektronik yang terpercaya.

### BAB III METODE PENELITIAN

Bab ini berisi langkah-langkah yang dilakukan dalam penelitian yang meliputi penemuan masalah, perumusan masalah, studi

pustaka, perancangan, analisis dan pemecahan masalah, dan penarikan simpulan.

#### BAB IV HASIL DAN PEMBAHASAN

Bab ini menguraikan pembahasan tentang penugasan *multi-objective* dan penyelesaiannya dengan Algoritma Kuhn-Munkres.

#### BAB V PENUTUP

Bab ini berisi simpulan dan saran.



## BAB II

### TINJAUAN PUSTAKA

Pada bab ini diberikan beberapa definisi, teorema, dan pengertian yang berhubungan dengan masalah umum penugasan, masalah penugasan sederhana (*single-objective*), terminologi dasar graf, jenis-jenis graf, penjadohan (*matching*), dan Algoritma Kuhn-Munkres.

#### 2.1 Masalah Penugasan (*Assignment Problem*)

##### 2.1.1 Masalah Umum Penugasan

Masalah penugasan (*assignment problem*) merupakan jenis khusus pemrograman linier yang mengalokasikan sumber-sumber kepada kegiatan-kegiatan atas dasar satu-satu (*one-to-one basis*) (Hillier, 2008:242). Setiap sumber (*assignee*) (misalnya, karyawan, mesin, atau satuan waktu) ditugasi secara khusus kepada suatu kegiatan atau tugas (misalnya, suatu pekerjaan, lokasi, atau kejadian). Sebagai contoh, permasalahannya berupa penugasan terbaik atas pekerja dengan pekerjaannya, pemain sepak bola dengan posisinya di lapangan, peralatan dengan lokasi konstruksi, dan sebagainya.

Dalam masalah penugasan, selain melakukan penempatan secara efisien, keefektifan total juga perlu dipertimbangkan. Dalam kasus ini, penugasan yang bertujuan menetapkan setiap tugas yang sesuai pada pekerja dengan memaksimalkan keefektifan total dari para pekerja dikenal sebagai masalah penugasan optimal (Bondy & Murty, 1976:86).

Beberapa metode telah dikembangkan untuk menyelesaikan masalah penugasan, diantaranya adalah metode simpleks transportasi, metode Hungaria, Algoritma Genetika, Algoritma Brute-Force, dan Algoritma Kuhn-Munkres. Adapun perangkat lunak yang dapat digunakan, diantaranya adalah Solver, Lindo, CPLEX, dan Matlab.

### 2.1.2 Masalah Penugasan Sederhana (*Single-Objective*)

Masalah penugasan sederhana (*single-objective*) adalah masalah penugasan yang mempunyai satu tujuan optimasi, yaitu memaksimalkan atau meminimumkan suatu sumber daya (pendapatan, biaya, jarak, atau waktu) yang digunakan untuk menyelesaikan tugas (White, 1984:759).

Suatu masalah penugasan sederhana yang hanya berkaitan dengan biaya operasi dapat dipresentasikan pada Tabel 2.1 dengan  $n$  tugas yang ditetapkan untuk  $m$  pekerja,  $c_{ij}$  adalah biaya operasi pekerja  $i$  ( $i = 1, 2, \dots, m$ ) untuk melaksanakan tugas  $j$  ( $j = 1, 2, \dots, n$ ). Tujuan akhirnya adalah menetapkan setiap pekerja pada tepat satu tugas sehingga total biaya operasi dapat diminimumkan.

Tabel 2.1. Masalah Penugasan Sederhana

Tugas \ Pekerja	1	2	....	$n$
1	$c_{11}$	$c_{12}$	....	$c_{1n}$
2	$c_{21}$	$c_{22}$	....	$c_{2n}$
....	....	....	....	....
$m$	$c_{m1}$	$c_{m2}$	....	$c_{mn}$

Oleh sebab setiap tugas ditetapkan hanya untuk pekerja tertentu. Model masalah matematis penugasan pada Tabel 2.1 dapat ditulis sebagai berikut (Hillier, 2008:243).

Minimumkan:

$$C = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (2.1)$$

dengan kendala

$$\sum_{j=1}^n x_{ij} = 1 \quad \text{untuk setiap } i = 1, 2, \dots, m \quad (2.2)$$

$$\sum_{i=1}^m x_{ij} = 1 \quad \text{untuk setiap } j = 1, 2, \dots, n \quad (2.3)$$

atau jika dijabarkan, persamaan (2.2) dan (2.3) menjadi,

$$x_{11} + x_{12} + \dots + x_{1n} = 1$$

$$x_{21} + x_{22} + \dots + x_{2n} = 1$$

.....

$$x_{m1} + x_{m2} + \dots + x_{mn}$$

$$x_{11} + x_{21} + \dots + x_{m1} = 1$$

$$x_{12} + x_{22} + \dots + x_{m2} = 1$$

.....

$$x_{1n} + x_{2n} + \dots + x_{mn} = 1$$

$$x_{ij} = \begin{cases} 1, & \text{jika tugas } j \text{ ditetapkan untuk pekerja } i. \\ 0, & \text{jika tugas } j \text{ tidak ditetapkan untuk pekerja } i. \end{cases} \quad (2.4)$$

Keterangan:

$C$  : Fungsi tujuan yang dicari nilai optimalnya (maksimum atau minimum).

$m$  : banyaknya pekerja yang akan diberi tugas.

$n$  : banyaknya tugas yang harus diselesaikan.

$x_{ij}$  : penugasan dari pekerja  $i$  ke tugas  $j$ .

$c_{ij}$  : parameter alokasi (biaya operasi) dari pekerja  $i$  ke tugas  $j$ .

Dengan asumsi  $m = n$ , maka akan ada  $n!$  ( $n$  faktorial) penugasan yang mungkin dalam suatu masalah karena berpasangan satu-satu (Munkres, 1957:32).



## 2.2 Teori Graf

Teori graf adalah cabang ilmu matematika yang mempelajari dan memodelkan permasalahan dalam bentuk graf. Teori graf pertama kali diperkenalkan pada tahun 1736 oleh seorang matematikawan Swiss yang bernama Leonard Euler. Baru pada sekitar tahun 1920 teori graf berkembang pesat terutama di bidang ilmu komputer, kimia, ekonomi, dan riset operasi.

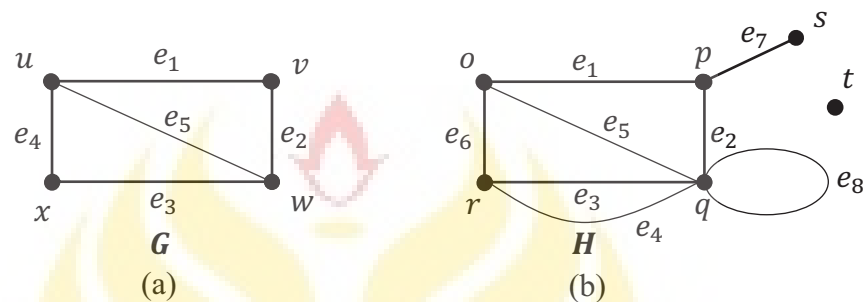
### 2.2.1 Definisi Graf

Sebuah graf  $G$  berisi dua himpunan yaitu himpunan berhingga tak kosong  $V(G)$  dari obyek-obyek yang disebut titik (*vertex*, *node*) dan himpunan berhingga (mungkin kosong)  $E(G)$  yang elemen-elemennya disebut sisi (*edge*) sehingga setiap elemen  $e$  dalam  $E(G)$  merupakan pasangan tak berurutan dari titik-titik  $V(G)$  (Budayasa, 2007:1). Himpunan  $V(G)$  disebut himpunan titik  $G$ , dan himpunan  $E(G)$  disebut himpunan sisi  $G$ . Misalkan  $x$  dan  $y$  adalah dua titik di  $G$ , sisi yang menghubungkan titik  $x$  dan  $y$  dinotasikan dengan  $e = \{x, y\}$  atau  $e = xy$ .

### 2.2.2 Terminologi Dasar Graf

Suatu graf  $G$  dapat dipresentasikan dalam bentuk diagram di mana setiap titik  $G$  digambarkan dengan sebuah noktah dan setiap sisi yang menghubungkan dua titik di  $G$  digambarkan dengan sebuah kurva sederhana (ruas garis). Sebagai contoh graf  $G$  dengan  $V(G) = \{u, v, w, x\}$  dan  $E(G) = \{e_1, e_2, e_3, e_4, e_5\}$  dengan  $e_1 = uv$ ,  $e_2 = vw$ ,  $e_3 = wx$ ,

$e_4 = ux$ , dan  $e_5 = uw$  dapat dipresentasikan dalam bentuk diagram seperti pada Gambar 2.1 (a). Contoh lainnya, graf  $H$  dengan  $V(H) = \{o, p, q, r, s, t\}$  dan  $E(H) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$  dengan  $e_1 = op$ ,  $e_2 = pq$ ,  $e_3 = qr$ ,  $e_4 = qr$ ,  $e_5 = oq$ ,  $e_6 = ro$ ,  $e_7 = ps$ , dan  $e_8 = qq$ , dapat dipresentasikan seperti pada Gambar 2.1 (b).



Gambar 2.1. (a) Graf  $G$  dengan 4 titik dan 5 sisi; (b) graf  $H$  dengan 6 titik dan 8 sisi

### 2.2.2.1 Bertetangga (*Adjacent*) dan Terkait (*Incident*)

Dipunyai  $u$  dan  $v$  adalah dua titik di  $G$  dan  $e = uv$  adalah sisi yang menghubungkan (*joining*) titik  $u$  dan titik  $v$ . Dapat dikatakan titik  $u$  dan  $v$  bertetangga (*adjacent*) di  $G$ ,  $u$  dan  $v$  titik-titik akhir sisi  $e$ , sisi  $e$  terkait (*incident*) dengan titik  $u$  dan titik  $v$  (Budayasa, 2007:2). Dua sisi dikatakan bertetangga (*adjacent*) jika dua sisi tersebut terkait (*incident*) pada titik yang sama. Dari Gambar 2.1 (a) tampak bahwa titik  $u$  dan titik  $v$  bertetangga (*adjacent*) di  $G$ . Sisi  $e_1$  menghubungkan (*joining*) titik  $u$  dan titik  $v$  di  $G$ . Titik  $u$  dan titik  $v$  titik-titik akhir sisi  $e_1$ . Sisi  $e_1$  terkait (*incident*) dengan titik  $v$  dan titik  $u$ . Sisi  $e_1$  bertetangga dengan sisi  $e_2$  karena sisi  $e_1$  dan  $e_2$  terkait oleh titik yang sama yaitu titik  $v$ .

### 2.2.2.2 Gelung (*loop*), Sisi Rangkap (*multiple-edges*), dan Graf Sederhana

Sisi graf yang menghubungkan suatu titik dengan dirinya sendiri disebut gelung (*loop*) (Budayasa, 2007:3). Sebagai contoh, sisi  $e_8$  di graf  $H$  pada Gambar 2.1 (b) adalah contoh gelung. Jika terdapat lebih dari satu sisi yang menghubungkan dua titik  $u$  dan  $v$  pada suatu graf, maka sisi-sisi tersebut disebut sisi-rangkap (*multiple-edges*). Sisi-sisi  $e_3$  dan  $e_4$  pada graf  $H$  adalah contoh sisi rangkap. Graf yang tidak mempunyai sisi-rangkap dan gelung disebut graf sederhana. Graf  $G$  pada Gambar 2.1 (a) merupakan contoh graf sederhana.

### 2.2.2.3 Derajat (*Degree*)

Menurut Chartrand dan Oellermann (1993:6), derajat (*degree*) dari titik  $v$  dalam graf  $G$  adalah banyaknya sisi yang terkait (*incident*) dengan  $v$ , yang dinotasikan  $d_G(v)$  atau  $d(v)$ . Tiap gelung dihitung dua kali. Derajat terbesar dari suatu titik dalam graf  $G$  disebut derajat maksimum (*maximum degree*) dan dinotasikan dengan  $\Delta(G)$ . Derajat terkecil dari suatu titik dalam graf  $G$  disebut derajat minimum (*minimum degree*) dan dinotasikan dengan  $\delta(G)$ . Sebagai contoh, graf  $H$  pada Gambar 2.1 (b) dapat dilihat bahwa  $d(t) = 0, d(q) = 6, \Delta(H) = 6, \delta(H) = 0$ .

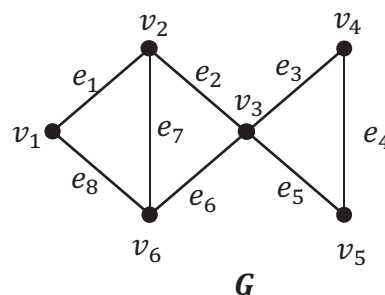
### 2.2.2.4 Titik Terasing dan Titik Ujung

Titik yang mempunyai derajat 0 disebut titik terasing (*isolated vertex*), sedang titik yang mempunyai derajat 1 disebut titik ujung (*end-vertex*) (Chartrand & Oellermann, 1993:6). Sebagai contoh, graf  $H$  pada

Gambar 2.1 (b) dapat dilihat bahwa titik  $t$  adalah titik terasing dan titik  $s$  adalah titik ujung.

### 2.2.2.5 Jalan (Walk), Jejak (Trail), Lintasan (Path), Sirkuit (Circuit), dan Sikel (Cycle)

Misalkan  $G$  adalah sebuah graf,  $u$  dan  $v$  adalah titik dari graf  $G$ . Menurut Budayasa (2007:6), sebuah jalan (*walk*) di graf  $G$  adalah sebuah barisan berhingga (tak kosong)  $W = (v_0, e_1, v_1, e_2, v_2, \dots, e_k, v_k)$  yang suku-sukunya bergantian titik dan sisi, sehingga  $v_{i-1}$  dan  $v_i$  adalah titik-titik akhir sisi  $e_i$ , untuk  $1 \leq i \leq k$ . Dikatakan  $W$  adalah sebuah jalan dari titik  $v_0$  dan titik  $v_k$ , atau jalan- $(v_0, v_k)$ . Titik  $v_0$  dan titik  $v_k$  berturut-turut disebut titik awal dan titik akhir  $W$ . Titik-titik  $v_1, v_2, \dots, v_{k-1}$  disebut titik-titik internal  $W$  dengan  $k$  adalah panjang jalan  $W$ . Jika semua sisi  $e_1, e_2, e_3, \dots, e_k$  dalam jalan  $W$  berbeda, maka  $W$  disebut jejak (*trail*). Jika semua titik  $v_0, v_1, v_2, \dots, v_k$  dalam  $W$  juga berbeda, maka  $W$  disebut lintasan (*path*). Sebuah jalan  $W$  dengan panjang positif disebut tertutup, jika titik awal dan titik akhir dari  $W$  sama. Jejak tutup disebut sirkuit (*circuit*). Sebuah sikel (*cycle*) adalah sebuah jejak tertutup yang titik awal dan semua titik internalnya berbeda.

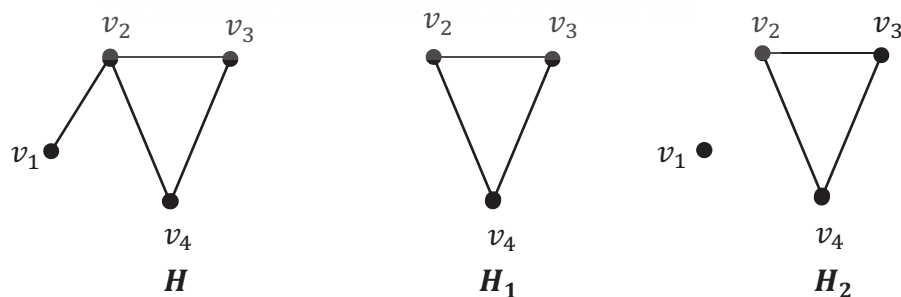


Gambar 2.2. Graf  $G$ : jalan, jejak, lintasan, sirkuit, dan sikel

Sebagai contoh barisan  $(v_3, e_2, v_2, e_7, v_6, e_6, v_3, e_3, v_4, e_4, v_5, e_4, v_4)$  dari graf  $G$  pada Gambar 2.2 adalah contoh jalan— $(v_3, v_4)$ . Karena dalam barisan tersebut sisi  $e_4$  muncul lebih dari sekali, jelas barisan tersebut bukan jejak. Barisan  $(v_3, e_2, v_2, e_7, v_6, e_6, v_3, e_3, v_4)$  adalah contoh jejak di graf  $G$  dengan panjang 4. Karena titik  $v_3$  muncul lebih dari sekali maka jejak tersebut bukan lintasan. Barisan  $(v_1, e_1, v_2, e_7, v_6, e_6, v_3, e_3, v_4, e_4, v_5)$  adalah contoh suatu lintasan di graf  $G$  dengan panjang 5. Barisan  $(v_1, e_1, v_2, e_2, v_3, e_3, v_4, e_4, v_5, e_5, v_3, e_6, v_6, e_8, v_1)$  adalah sebuah sirkuit di di graf  $G$  dengan panjang 7. Sirkuit ini bukan sikel sebab titik internal  $v_3$  muncul lebih dari sekali. Barisan  $(v_1, e_1, v_2, e_2, v_3, e_6, v_6, e_8, v_1)$  adalah contoh sebuah sikel di graf  $G$  dengan panjang 4.

### 2.2.3 Graf Bagian (*Subgraph*)

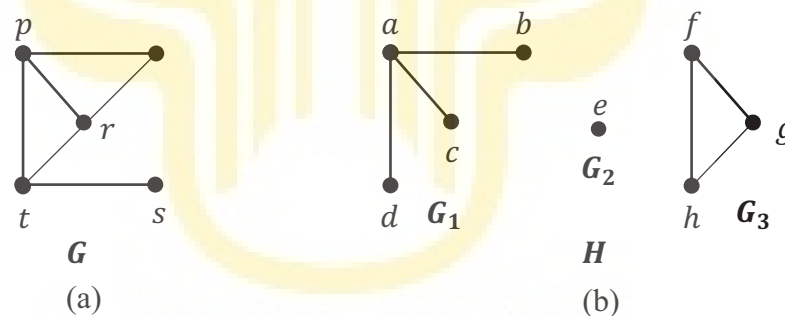
Suatu graf  $H$  disebut graf bagian (*subgraph*) dari graf  $G$ , ditulis  $H \subset G$ , jika  $V(H) \subset V(G)$  dan  $E(H) \subset E(G)$  (Bondy & Murty, 1976:8). Jika  $H \subset G$  dan  $V(H) = V(G)$ , maka  $H$  disebut graf bagian perentang (*spanning subgraph*). Sebagai contoh, pada Gambar 2.3  $H_1$  merupakan contoh graf bagian dari graf  $H$  dan  $H_2$  adalah graf bagian perentang dari  $H$ .



Gambar 2.3.  $H_1$  graf bagian  $H$  dan  $H_2$  graf bagian perentang

### 2.2.4 Graf Terhubung (*Connected*) dan Komponen Graf

Menurut Chartrand dan Lesniak (1996:18), graf  $G$  dikatakan terhubung (*connected*) jika untuk setiap dua titik berbeda  $u$  dan  $v$ , terdapat suatu lintasan  $-(u, v)$  yang menghubungkan titik  $u$  dan  $v$ . Sebaliknya, graf  $G$  disebut tidak terhubung (*disconnected*) jika tidak ada lintasan yang menghubungkan antara kedua titik yang berbeda. Sebuah komponen graf  $G$  adalah sebuah graf bagian terhubung maksimal (titik dan sisi) dari  $G$ . Jadi setiap graf terhubung memiliki tepat satu komponen sedangkan graf tidak terhubung memiliki paling sedikit dua komponen. Sebagai contoh graf  $G$  pada gambar 2.4 (a) adalah graf terhubung dan graf  $H$  pada gambar 2.4 (b) adalah graf tidak terhubung dengan 3 komponen, yaitu  $G_1, G_2, G_3$ .

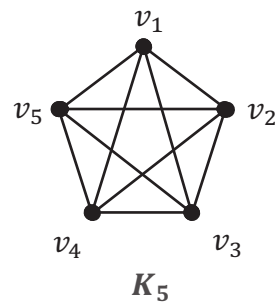


Gambar 2.4. (a) Graf terhubung  $G$ ; (b) graf tidak terhubung  $H$  dengan 3 komponen  $G_1, G_2, G_3$

### 2.2.5 Graf Komplit

Suatu graf komplit dengan  $n$  titik, dilambangkan dengan  $K_n$  adalah graf sederhana dengan  $n$  titik dan setiap dua titik berbeda dihubungkan dengan sebuah sisi (Budayasa, 2007:3). Sebagai contoh, graf komplit dengan 5 titik diperlihatkan pada Gambar 2.5 berikut:

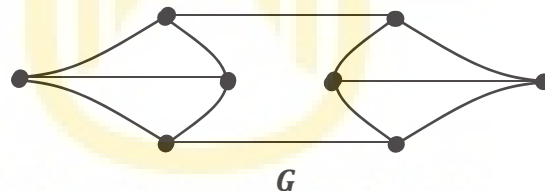




Gambar 2.5  $K_5$  graf komplit dengan 5 titik

### 2.2.6 Graf Beraturan- $k$ ( $k$ -Regular)

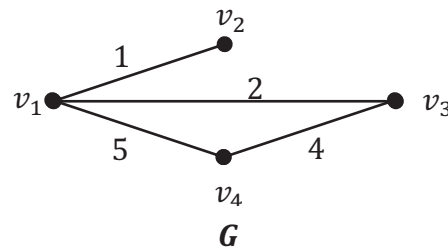
Sebuah graf  $G$  disebut graf beraturan- $k$  jika setiap titik  $G$  berderajat  $k$  (Budayasa, 2007:11). Berdasarkan definisi dapat dikatakan, jika  $G$  graf beraturan maka  $\delta(G) = \Delta(G)$ . Sebagai contoh, graf  $G$  pada gambar 2.6 adalah graf beraturan-3.



Gambar 2.6 Graf  $G$  beraturan-3,  $\delta(G) = 3 = \Delta(G)$

### 2.2.7 Graf Berbobot

Sebuah graf yang setiap sisinya dikaitkan dengan suatu bilangan riil disebut graf berbobot (Budayasa, 2007:37). Bilangan yang dikaitkan ke suatu sisi  $G$  disebut bobot sisi tersebut. Bobot pada graf  $G$  adalah jumlah bobot semua sisi  $G$  dan dinotasikan dengan  $w(G)$ . Sebagai contoh, graf  $G$  pada Gambar 2.7 adalah graf berbobot dengan  $w(G) = 12$ .

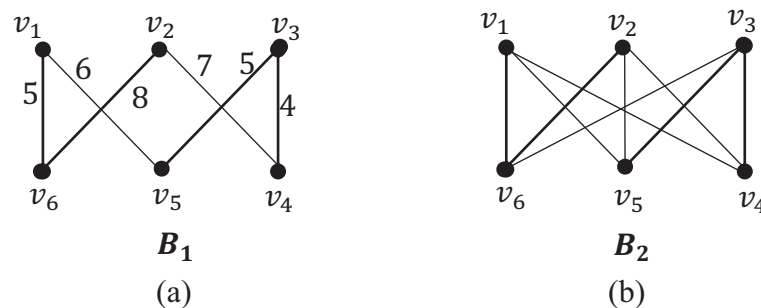


Gambar 2.7. Graf berbobot  $G$  dengan  $w(G) = 12$

### 2.2.8 Graf Bipartisi

Suatu graf  $G$  dikatakan graf bipartisi jika himpunan titik  $V(G)$  dapat dipartisi menjadi dua sub himpunan tak kosong  $V_1(G)$  dan  $V_2(G)$  sehingga tiap sisi dari  $G$  terkait (*incident*) dengan titik di  $V_1(G)$  dan titik di  $V_2(G)$  (Gu *et al.*, 2011:112). Pasangan himpunan  $(V_1(G), V_2(G))$  disebut partisi dari  $G$ .

Selanjutnya menurut Bondy dan Murty (1976:5), apabila graf bipartisi  $G$  dengan partisi  $(V_1(G), V_2(G))$  sehingga setiap titik di  $V_1(G)$  bertetangga (*adjacent*) dengan setiap titik di  $V_2(G)$ , maka  $G$  disebut graf bipartisi komplit, dilambangkan dengan  $K_{m,n}$  dengan  $|V_1(G)| = m$  dan  $|V_2(G)| = n$ . Jika setiap sisi dari graf bipartisi  $G$  diberi nilai atau bobot, maka disebut graf bipartisi berbobot.

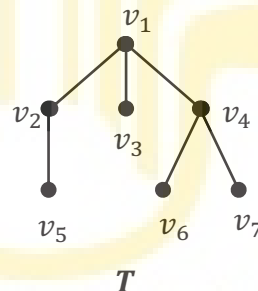


Gambar 2.8. (a) Graf bipartisi berbobot  $B_1$ ; (b) graf bipartisi komplit  $B_2$

Graf  $B_1$  pada Gambar 2.8 (a) menunjukkan graf bipartisi berbobot dengan titik  $V(G)$  dipartisi menjadi sub himpunan  $V_1(G) = \{v_1, v_2, v_3\}$  dan  $V_2(G) = \{v_4, v_5, v_6\}$  serta tiap sisi menghubungkan titik  $V_1(G)$  dan  $V_2(G)$  dengan diberi nilai atau bobot. Contoh lainnya, graf  $B_2$  pada Gambar 2.6 (b) merupakan graf bipartisi komplit.

### 2.2.9 Pohon (*Tree*)

Graf terhubung yang tidak memuat siklus disebut pohon (*tree*) (Budayasa, 2007:22). Sebuah graf pohon  $T$  disebut pohon berakar jika  $T$  mempunyai satu titik akar yaitu  $r$  di mana untuk setiap titik  $v$  di  $T$  terdapat lintasan  $r$  ke  $v$ . Contoh pohon diperlihatkan pada Gambar 2.9 berikut:



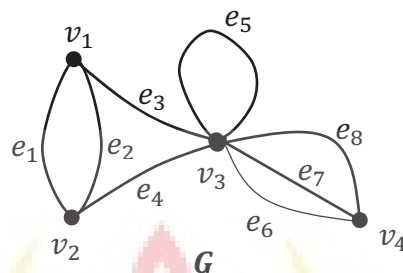
Gambar 2.9 Pohon  $T$

### 2.2.10 Representasi Graf dalam Matriks

Selain dengan diagram, suatu graf dapat disajikan dalam bentuk matriks. Terdapat dua jenis matriks yang dapat digunakan, yaitu matriks ketetanggaan (*adjacence matrix*) dan matriks keterkaitan (*incidence matrix*).

Dipunyai graf  $G$  dengan  $V(G) = \{v_1, v_2, \dots, v_n\}$ . Matriks ketetanggaan (*adjacence matrix*) pada graf  $G$  adalah matriks bujur sangkar

$A(G) = (a_{ij})$  berordo  $n \times n$  yang baris-baris dan kolom-kolomnya dilabel dengan label titik-titik graf  $G$  sehingga elemen  $a_{ij}$  menyatakan banyaknya sisi  $G$  yang menghubungkan titik  $v_i$  dan  $v_j$  (Budayasa, 2007:14).



Gambar 2.10. Graf  $G$  dengan titik-titik yang bertetangga (*adjacent*) dan terkait (*incident*)

Sebagai contoh, diberikan graf  $G$  pada Gambar 2.10 dapat disajikan dalam bentuk matriks ketetanggaan sebagai berikut:

$$A(G) = \begin{matrix} & \begin{matrix} v_1 & v_2 & v_3 & v_4 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{bmatrix} 0 & 2 & 1 & 0 \\ 2 & 0 & 1 & 0 \\ 1 & 1 & 1 & 3 \\ 0 & 0 & 3 & 0 \end{bmatrix} \end{matrix}$$

Selain dengan matriks ketetanggaan, sebuah graf juga dapat disajikan dalam matriks keterkaitan (*incidence matrix*). Dipunyai graf  $G$  dengan  $n$  buah titik  $V(G) = \{v_1, v_2, \dots, v_n\}$ , dan  $t$  buah sisi  $E(G) = \{e_1, e_2, \dots, e_t\}$ . Matriks keterkaitan (*incidence matrix*) dari graf  $G$  adalah matriks  $M(G) = (m_{ij})$  berordo  $n \times t$  yang baris-baris dilabel dengan label titik-titik graf  $G$  dan kolom-kolomnya dilabel dengan label sisi-sisi graf  $G$  sehingga

$$m_{ij} = \begin{cases} 0, & \text{jika sisi } e_j \text{ tidak terkait dengan } v_i. \\ 1, & \text{jika sisi } e_j \text{ terkait dengan } v_i \text{ dan } e_j \text{ bukan gelung.} \\ 2, & \text{jika sisi } e_j \text{ terkait dengan } v_i \text{ dan } e_j \text{ gelung.} \end{cases}$$

(Budayasa, 2007:16).

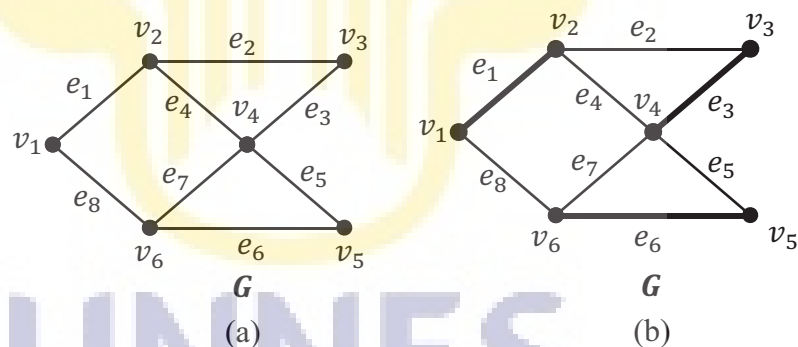
Sebagai contoh, diberikan graf  $G$  pada Gambar 2.10 dapat disajikan dalam bentuk matriks keterkaitan sebagai berikut:

$$M(G) = \begin{matrix} & c \\ & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \\ v_1 & \left[ \begin{array}{cccccccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ v_2 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ v_3 & 0 & 0 & 1 & 1 & 2 & 1 & 1 & 1 \\ v_4 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{array} \right] \end{matrix}$$

## 2.3 Penjodohan (*Matching*)

### 2.3.1 Definisi Penjodohan (*Matching*)

Suatu himpunan bagian  $M$  dari himpunan sisi  $E(G)$  disebut sebagai penjodohan (*matching*) pada graf  $G$  jika sisi-sisinya memiliki titik ujung yang berbeda dan tidak ada dua sisi yang bertetangga (*adjacent*) (Bondy & Murty, 1976:70).



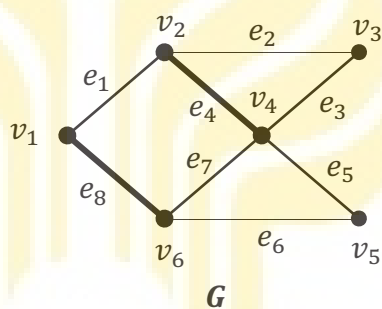
Gambar 2.11. Graf dasar  $G$  (a); (b) graf  $G$  dengan penjodohan  $M_1$

Sebagai contoh, graf  $G$  pada Gambar 2.11 (b), himpunan sisi  $M_1 = \{e_1, e_3, e_6\}$  atau dapat ditulis dengan notasi  $M_1 = \{v_1v_2, v_3v_4, v_5v_6\}$  (diilustrasikan dengan sisi yang dicetak tebal) adalah sebuah penjodohan berukuran 3, ditulis  $|M_1| = 3$ . Contoh lain penjodohan berukuran 3 dari graf  $G$  pada Gambar 2.11 (a) adalah himpunan  $M_2 = \{e_2, e_5, e_8\}$ . Himpunan  $M_3 = \{e_4, e_8\}$  dan  $M_4 = \{e_4, e_6\}$  adalah penjodohan berukuran

2 pada graf  $G$ . Tetapi, himpunan  $E = \{e_1, e_2, e_5\}$  bukan penjodohan pada graf  $G$  karena sisi  $e_1$  dan  $e_2$  terkait ke titik yang sama yaitu titik  $v_2$ .

### 2.3.2 Tertutup oleh Penjodohan (*Matching-Saturated*)

Suatu titik  $v$  di graf  $G$  dikatakan tertutup oleh penjodohan  $M$  ( $M$ -saturated) jika titik  $v$  merupakan titik akhir atau titik ujung dari salah satu sisi di  $M$ . Sebaliknya, jika  $M$  tidak menutup  $v$  di graf  $G$  maka titik  $v$  dikatakan tidak tertutup oleh penjodohan  $M$  ( $M$ -unsaturated) (Budayasa, 2007:185).



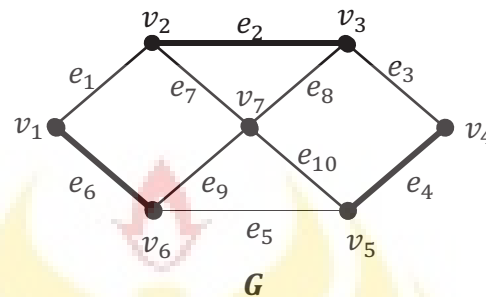
Gambar 2.12. Graf  $G$  dengan Penjodohan  $M_3$

Sebagai contoh, graf  $G$  pada Gambar 2.12 memuat penjodohan  $M_3 = \{e_4, e_8\}$  yang menutup titik-titik  $v_1, v_6, v_2, v_4$ ; tetapi  $M_3$  tidak menutup titik  $v_3$  dan  $v_5$ . Dengan kata lain, titik-titik  $v_1, v_6, v_2, v_4$  adalah tertutup oleh penjodohan  $M_3$  ( $M_3$ -saturated), tetapi titik-titik  $v_3$  dan  $v_5$  tidak tertutup oleh penjodohan  $M_3$  ( $M_3$ -unsaturated).

### 2.3.3 Lintasan Alternatif (*Alternating Path*) dan Lintasan Augmentasi (*Augmenting Path*)

Misalkan  $M$  adalah penjodohan di graf  $G$ ,  $E$  himpunan sisi di  $G$  yang bukan penjodohan. Lintasan alternatif- $M$  ( $M$ -alternating path)

adalah lintasan yang sisinya bergantian di  $E \setminus M$  dan di  $M$ . Sebuah lintasan alternatif- $M$  yang titik awal dan titik akhirnya tidak tertutup oleh  $M$  dinamakan lintasan augmentasi- $M$  ( $M$ -augmenting path) (Budayasa, 2007:185).



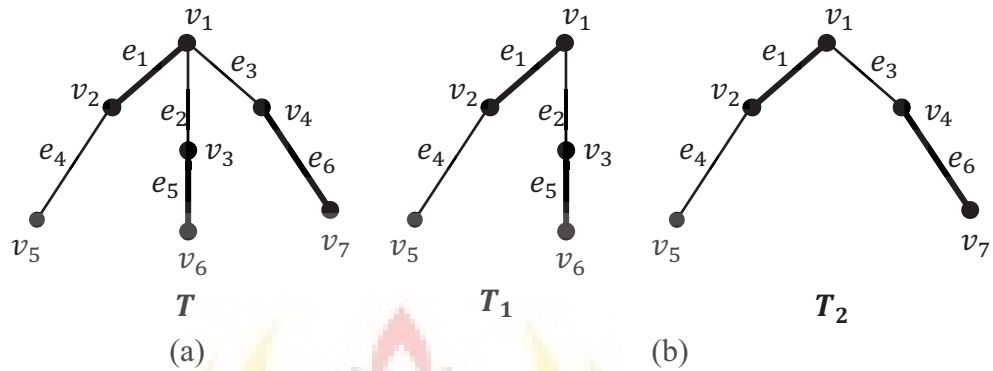
Gambar 2.13. Graf  $G$  dengan Penjodohan  $M$

Sebagai contoh, penjodohan  $M = \{e_2, e_4, e_6\}$  pada Gambar 2.13, lintasan  $P = (v_7, e_{10}, v_5, e_4, v_4, e_3, v_3, e_2, v_2, e_1, v_1, e_6, v_6)$  adalah sebuah lintasan alternatif- $M$ . Karena titik  $v_6$  tertutup oleh  $M$ , maka  $P$  bukan lintasan augmentasi- $M$ . Selanjutnya perhatikan penjodohan  $M_1 = \{e_2, e_4\}$  pada graf  $G$ . Dapat dilihat bahwa titik-titik ujung lintasan  $P_1 = (v_7, e_7, v_2, e_2, v_3, e_3, v_4, e_4, v_5, e_5, v_6)$  pada graf  $G$  yaitu  $v_7$  dan  $v_6$  tidak tertutup oleh  $M_1$ , maka lintasan  $P_1$  merupakan lintasan augmentasi- $M_1$  pada graf  $G$ .

#### 2.3.4 Pohon Alternatif

Dipunyai sebuah graf pohon  $T$  dengan penjodohan  $M$ . Titik  $v$  adalah titik yang tidak tertutup oleh penjodohan  $M$ . Pohon alternatif- $M$  adalah sebuah pohon yang berakar di titik  $v$  yang mempunyai sifat semua

lintasan yang berawal dari titik  $v$  adalah lintasan alternatif- $M$  (Chartrand & Oellermann, 1993:169).

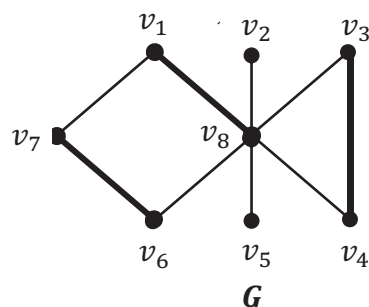


Gambar 2.14. (a) Graf pohon  $T$  dengan penjadohan  $M$ ; (b)  $T_1$  dan  $T_2$  pohon alternatif dari graf pohon  $T$

Sebagai contoh, graf pohon  $T$  pada Gambar 2.14 dapat dibangun sebuah pohon alternatif  $T_1$  dan  $T_2$  yang berakar dari titik  $v_5$  dengan lintasan  $P_1 = (v_5, e_4, v_2, e_1, v_1, e_2, v_3, e_5, v_6)$  untuk  $T_1$  dan lintasan  $P_2 = (v_5, e_4, v_2, e_1, v_1, e_3, v_4, e_6, v_7)$  untuk  $T_2$ .

### 2.3.5 Penjadohan Maksimum

Sebuah penjadohan  $M$  di graf  $G$  dikatakan penjadohan maksimum jika  $G$  tidak mempunyai penjadohan yang lain dengan ukuran lebih besar dari penjadohan  $M$ . Dengan kata lain, jika  $M'$  penjadohan pada graf  $G$  maka  $|M| \geq |M'|$  (Bondy & Murty, 1976:70).



Gambar 2.15. Penjadohan maksimum di graf  $G$



Sebagai contoh, graf  $G$  pada Gambar 2.15, penjadohan  $M_1 = \{v_1v_8, v_3v_4, v_6v_7\}$  dengan  $|M_1| = 3$  adalah penjadohan maksimum pada  $G$  sebab tidak ada penjadohan lain di graf  $G$  yang berukuran 4.

*Teorema 2.1:* Jika  $M$  sebuah penjadohan dan  $P$  adalah lintasan augmentasi- $M$  pada graf  $G$ , maka ada penjadohan  $M'$  pada  $G$  dengan  $|M'| = |M| + 1$ . (2.4)

*Bukti:* Dipunyai  $M$  adalah sebuah penjadohan di graf  $G$ . Misalkan  $P = (v_1, e_1, v_2, e_2, v_3, \dots, e_{2m-1}, v_{2m})$  sebuah lintasan augmentasi- $M$  pada graf  $G$ , maka  $E_1 = \{e_2, e_4, e_6, \dots, e_{2m-2}\} \subseteq M$ . Sehingga  $E_1^* = \{e_1, e_3, e_5, \dots, e_{2m-1}\}$  juga sebuah penjadohan baru pada graf  $G$ . Jelas bahwa,

$$|E_1^*| = |E_1| + 1. \quad (2.5)$$

Selanjutnya, dibentuk penjadohan baru  $M'$  sedemikian hingga,

$$M' = E_1^* \cup (M - E_1). \quad (2.6)$$

Karena  $E_1^* \cap (M - E_1) = \emptyset$ , maka

$$\begin{aligned} |M'| &= |E_1^*| + |M - E_1| \\ &= |E_1^*| + |M| - |E_1| \\ &= |E_1| + 1 + |M| - |E_1|, \text{ karena } E_1 \subseteq M \\ &= |M| + 1. \end{aligned}$$

Dengan demikian teorema terbukti (Budayasa, 2007:185).

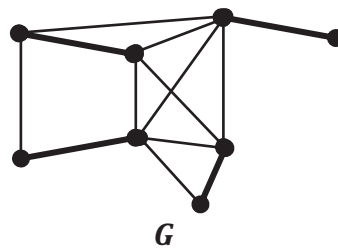
*Teorema 2.2 (Berge, 1957)* Sebuah penjadohan  $M$  di graf  $G$  adalah penjadohan maksimum jika dan hanya jika  $G$  tidak memuat lintasan augmentasi- $M$ .

*Bukti:* Dipunyai  $M$  adalah penjadohan maksimum pada graf  $G$ . Menurut Teorema 2.1, terdapat penjadohan  $M'$  pada  $G$  dengan  $|M'| = |M| + 1$ , kontradiksi bahwa  $M$  penjadohan maksimum pada  $G$ .

Sebaliknya, misalkan graf  $G$  tidak memuat lintasan augmentasi- $M$ . Andaikan  $M$  bukan penjadohan maksimum pada  $G$ . Misalkan  $M'$  penjadohan maksimum pada  $G$ , maka  $|M'| > |M|$ . Misalkan  $H$  adalah graf bagian  $G$  yang dibangun oleh  $M \Delta M' = (M \cup M') - (M' \cap M)$ . Setiap titik di  $H$  mempunyai derajat 1 atau 2, maka setiap komponen  $H$  merupakan sikel yang panjangnya genap yang sisi-sisinya bergantian di  $M$  dan  $M'$  atau lintasan dengan sisi-sisinya bergantian di  $M$  dan  $M'$ . Karena  $|M'| > |M|$ , maka  $H$  memuat lebih banyak sisi  $M'$  dari pada sisi  $M$ . Sehingga ada komponen di  $H$  berupa lintasan yang sisi awal dan sisi akhirnya di  $M'$ . Dengan demikian titik awal dan titik akhir lintasan tersebut tidak tertutup oleh  $M$ . Oleh karena itu lintasan tersebut adalah lintasan augmentasi- $M$  pada  $G$ , kontradiksi. Dengan demikian bukti teorema lengkap (Budayasa, 2007:186).

### 2.3.6 Penjadohan Sempurna (*Perfect Matching*)

Sebuah penjadohan  $M$  di graf  $G$  dikatakan penjadohan sempurna (*perfect matching*) jika  $M$  memuat semua titik  $G$  (Budayasa, 2007:184). Dengan kata lain, jika  $M$  penjadohan sempurna pada graf  $G$ , maka setiap titik di  $G$  tertutup oleh penjadohan ( $M$ -saturated). Sebagai contoh, penjadohan  $M$  di graf  $G$  pada Gambar 2.16 merupakan penjadohan sempurna karena  $M$  memuat semua titik  $G$ .

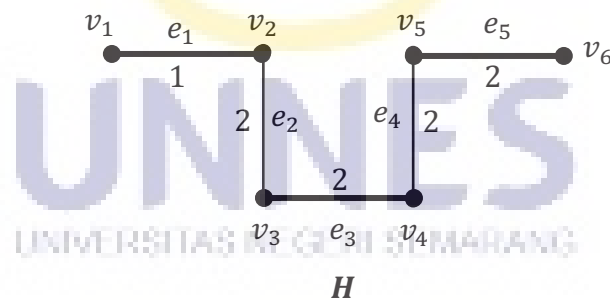


Gambar 2.16. Penjodohan sempurna di graf  $G$

Berdasarkan definisi, setiap penjodohan sempurna adalah penjodohan maksimum, tetapi tidak berlaku sebaliknya, artinya jika  $M$  adalah penjodohan maksimum, maka belum tentu  $M$  penjodohan sempurna (Budayasa, 2007:184).

### 2.3.7 Penjodohan Bobot Maksimum

Penjodohan bobot maksimum pada graf berbobot adalah penjodohan yang memiliki jumlah bobot sisi maksimum (Chartrand & Oellermann, 1993:162).



Gambar 2.17. Graf Berbobot  $H$  dengan Penjodohan  $M$

Sebagai contoh, graf  $H$  pada Gambar 2.17,  $M = \{e_1, e_3, e_5\}$  adalah penjodohan bobot maksimum.

### 2.3.8 Penjodohan pada Graf Bipartisi

Dipunyai  $G$  sebuah graf dan  $S \subseteq V(G)$ , dapat didefinisikan himpunan persekitaran (*neighbour set*) dari  $S$  di  $G$  adalah himpunan semua titik  $G$  yang bertetangga (*adjacent*) dengan titik-titik di  $S$ , dan dinotasikan dengan  $N_G(S)$  atau  $N(S)$  (Bondy & Murty, 1976:72). Selanjutnya, untuk menemukan penjodohan dari graf bipartisi  $G$  dengan partisi  $(X, Y)$  yang menutup semua titik di  $X$  telah diberikan syarat perlu dan syarat cukup oleh Hall (1935).

*Teorema 2.3 (Hall, 1935)* Misalkan  $G$  adalah graf bipartisi dengan partisi  $(X, Y)$ . Graf  $G$  memuat sebuah penjodohan yang menutup semua titik  $X$  jika dan hanya jika untuk setiap  $S \subseteq X$  berlaku

$$|N(S)| \geq |S|. \quad (2.7)$$

*Bukti:* Misalkan graf bipartisi  $G$  dengan partisi  $(X, Y)$  memuat penjodohan  $M$  yang menutup semua titik  $X$ ,  $S \subseteq X$ . Karena titik-titik di  $S$  dipasangkan oleh  $M$  ke titik yang berbeda di  $N(S) \subseteq Y$  maka  $|N(S)| \geq |S|$ .

Sebaliknya, misalkan  $G$  adalah graf bipartisi dengan partisi  $(X, Y)$  yang memenuhi (2.7), untuk setiap  $S \subseteq X$ . Andaikan  $G$  tidak memuat penjodohan yang menutup semua titik  $X$ . Misal  $M^*$  adalah penjodohan maksimum di  $G$ , karena pengandaian,  $M^*$  tidak menutup semua titik  $X$ , berarti ada titik di  $X$  yang tidak ditutupi oleh  $M^*$ , namakan titik  $u$ . Misalkan  $Z$  menyatakan himpunan semua titik yang terhubung ke  $u$  oleh lintasan-lintasan alternatif- $M^*$ . Karena  $M^*$  penjodohan maksimum maka menurut Teorema 2.2, hanya titik  $u$  yang tidak tertutup oleh  $M^*$  di  $Z$ .

Namakan himpunan  $S = Z \cap X$  dan  $T = Z \cap Y$ . Jelas titik-titik di  $S \setminus \{u\}$  dipasangkan oleh  $M^*$  dengan titik di  $T$ . Oleh karena itu

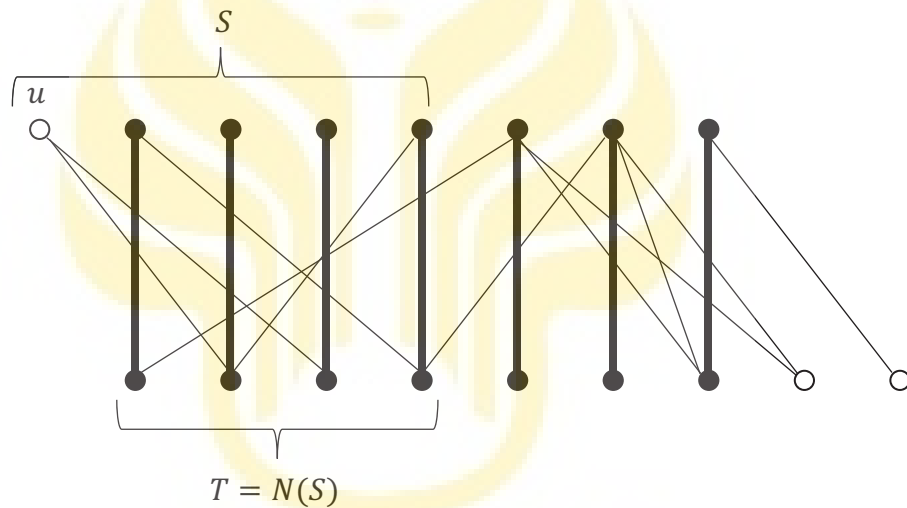
$$|T| = |S| - 1 \quad \dots \dots \dots (1)$$

dan  $T \subseteq N(S)$ .

Karena setiap titik di  $N(S)$  terhubung ke  $u$  oleh lintasan alternatif- $M^*$  maka  $N(S) \subseteq T$ , sehingga diperoleh

$$N(S) = T \quad \dots \dots \dots (2)$$

Sebagai ilustrasi, diperlihatkan pada Gambar 2.18 berikut:



Gambar 2.18. Graf bipartisi  $G$  yang memenuhi  $|N(S)| \geq |S|$

Dari (1) dan (2) didapatkan

$$|N(S)| = |S| - 1 < |S| \quad (2.8)$$

adalah suatu kontradiksi. Dengan demikian teorema terbukti (Budayasa, 2007:187).

*Akibat Teorema 2.4* Jika  $G$  adalah graf bipartisi beraturan- $k$  dengan  $k > 0$ , maka  $G$  memuat penjodohan sempurna.

*Bukti:* Misalkan  $G$  adalah graf bipartisi beraturan- $k$  dengan partisi  $(X, Y)$ .

Karena  $G$  adalah beraturan  $k$ , maka

$$k|X| = |E(G)| = k|Y|, \quad (2.9)$$

dan karena  $k > 0$ ,

$$|X| = |Y|. \quad (2.10)$$

Sebaliknya, dipunyai  $S \subseteq X$  dan  $E_1$  dan  $E_2$ , secara berturut-turut adalah himpunan-himpunan dari sisi-sisi yang terkait dengan titik  $S$  dan  $N(S)$ .

Berdasarkan definisi dari  $N(S)$ ,  $E_1 \subseteq E_2$ , oleh karena itu

$$k|N(S)| = |E_2| \geq |E_1| = k|S|. \quad (2.11)$$

Akibatnya,

$$|N(S)| \geq |S|,$$

sehingga berdasarkan Teorema 2.3, graf  $G$  memuat penjodohan  $M$  yang menutup semua titik  $X$ . Karena  $|X| = |Y|$  maka penjodohan  $M$  menutup semua titik  $Y$ . Jadi  $M$  adalah penjodohan sempurna pada graf  $G$ . Akibat Teorema 2.4 sering disebut sebagai Marriage Theorem (Bondy & Murty, 1976:72).

## 2.4 Aplikasi Penjodohan

### 2.4.1 Aplikasi Penjodohan pada Masalah Penugasan Personal

Dalam suatu perusahaan, misalkan  $x_1, x_2, x_3, \dots, x_n$  menyatakan  $n$  pekerja dan  $y_1, y_2, y_3, \dots, y_n$  menyatakan  $n$  pekerjaan. Setiap pekerja memiliki kemampuan untuk melakukan satu atau lebih pekerjaan tersebut. Dapatkah dibuat pasangan antara pekerja dan pekerjaan yang ada sehingga semua pekerja mendapat pekerjaan yang sesuai dengan kemampuannya?

Permasalahan tersebut dapat dimodelkan dengan membentuk graf bipartisi  $G$  dengan partisi  $(X, Y)$  di mana  $X = \{x_1, x_2, x_3, \dots, x_n\}$ ,  $Y = \{y_1, y_2, y_3, \dots, y_n\}$  dan  $x_i$  bertetangga dengan  $y_j$  jika dan hanya jika pekerja  $x_i$  dapat melakukan pekerjaan  $y_j$ . Selanjutnya, ditentukan ada atau tidaknya penjadohan sempurna pada graf  $G$ . Menurut Teorema 2.3 (Teorema Hall), adanya penjadohan sempurna pada graf  $G$  jika untuk setiap  $S \subseteq X$  berlaku  $|N(S)| \geq |S|$ . Sebaliknya, jika ada  $S \subset X$  sehingga  $|N(S)| < |S|$  maka  $G$  tidak mempunyai penjadohan sempurna. Dengan kata lain, tujuan akhir dari pemecahan masalah penugasan personal adalah menemukan penjadohan dari  $G$  yang menutup semua titik  $X$  atau jika tidak menemukan penjadohan yang demikian, akan ditemukan himpunan bagian  $S$  dari  $X$  sehingga  $|N(S)| < |S|$ .

#### 2.4.2 Aplikasi Penjadohan pada Masalah Penugasan Optimal

Pada masalah penugasan personal yang dimodelkan dengan membentuk graf bipartisi  $G$ , dalam kasus penugasan optimal dibatasi pada graf bipartisi komplit berbobot. Diberikan graf bipartisi komplit berbobot  $(X, Y)$  dengan  $X = \{x_1, x_2, x_3, \dots, x_n\}$ ,  $Y = \{y_1, y_2, y_3, \dots, y_n\}$  dan sisi  $x_i y_j$  mempunyai bobot  $w_{ij} = w(x_i y_j)$  yang menyatakan tingkat kemampuan dari pekerja  $x_i$  dalam menyelesaikan pekerjaan  $y_j$ . Solusi masalah penugasan optimal adalah menemukan penjadohan sempurna dengan bobot maksimum dalam graf bipartisi komplit berbobot atau yang disebut dengan penjadohan optimal (Bondy & Murty, 1976:86).

Dalam penyelesaian masalah penugasan optimal, pada graf  $G = K_{n,n}$  dimungkinkan untuk dibuat  $n!$  penjadohan sempurna dan menemukan satu yang optimal di antara penjadohan sempurna lainnya. Pada kasus  $n$  besar, mendaftar semua penjadohan sempurna pada graf  $K_{n,n}$  dan menemukan satu yang optimal tidaklah efisien.

Pada bagian ini akan dijelaskan algoritma untuk menemukan penjadohan optimal pada graf bipartisi komplit berbobot. Terlebih dahulu diperkenalkan sebuah pelabelan titik yang layak (*feasible vertex labelling*) pada graf  $G$ , yaitu sebuah fungsi  $\ell$  yang memetakan setiap titik  $G$  ke bilangan riil sehingga untuk setiap  $x \in X$  dan  $y \in Y$  berlaku

$$\ell(x) + \ell(y) \geq w(xy). \quad (2.12)$$

Dengan kata lain, pelabelan titik dikatakan layak (*feasible*) jika jumlah label dua titik pada graf  $G$  minimal sama dengan bobot sisi yang menghubungkan dua titik tersebut. Selanjutnya, untuk suatu titik  $v$  pada  $G$ , bilangan riil  $\ell(v)$  disebut label dari titik  $v$ . Pada graf bipartisi komplit berbobot, selalu terdapat pelabelan titik yang layak. Sebagai contoh, fungsi  $\ell$  yang didefinisikan seperti berikut:

$$\ell(x) = \text{maksimum } \{w(xy) | y \in Y\}, \text{ untuk setiap } x \in X$$

$$\ell(y) = 0, \text{ untuk setiap } y \in Y$$

maka  $\ell$  adalah pelabelan titik yang layak di  $G$  dengan

$$E_\ell = \{xy \in E(G) | x \in X, y \in Y \text{ dan } \ell(x) + \ell(y) = w(xy)\}, \quad (2.13)$$

dengan kata lain  $E_\ell$  adalah himpunan sisi  $G$  yang bobotnya sama dengan jumlah label titik-titik ujungnya.



*Teorema 2.5* Dipunyai  $\ell$  adalah sebuah pelabelan titik yang layak pada graf bipartisi komplit berbobot  $G$  dan  $G_\ell$  adalah graf bagian perentang  $G$  yang dibangun oleh sisi-sisi  $E_\ell$ . Jika graf  $G_\ell$  memuat penjadohan sempurna  $M^*$ , maka  $M^*$  adalah penjadohan optimal pada graf  $G$ .

*Bukti:* Dipunyai  $G$  adalah graf bipartisi komplit berbobot,  $G_\ell$  memuat penjadohan sempurna  $M^*$ . Karena  $G_\ell$  adalah graf bagian perentang dari  $G$ , maka  $M^*$  juga penjadohan sempurna pada graf  $G$ . Karena setiap  $e \in M^*$  terletak pada  $G_\ell$  dan titik-titik ujung sisi-sisi  $M^*$  menutup setiap titik tepat satu kali, maka

$$w(M^*) = \sum_{e \in M^*} w(e) = \sum_{v \in V} \ell(v) \dots \dots \dots (1)$$

Sementara, jika  $M$  sebarang penjadohan sempurna pada graf  $G$ , berdasarkan (2.12), maka

$$w(M) = \sum_{e \in M} w(e) \leq \sum_{v \in V} \ell(v) \dots \dots \dots (2)$$

Dari (1) dan (2) diperoleh,

$$w(M) \leq w(M^*). \quad (2.14)$$

Ini berarti bahwa  $M^*$  merupakan penjadohan sempurna dengan bobot maksimum (penjadohan optimal) pada graf  $G$ .

Teorema 2.5 dijadikan dasar dalam pembuatan algoritma oleh Kuhn (1955) dan Munkres (1957) untuk menentukan sebuah penjadohan optimal dalam graf bipartisi komplit berbobot (Budayasa, 2007:202).

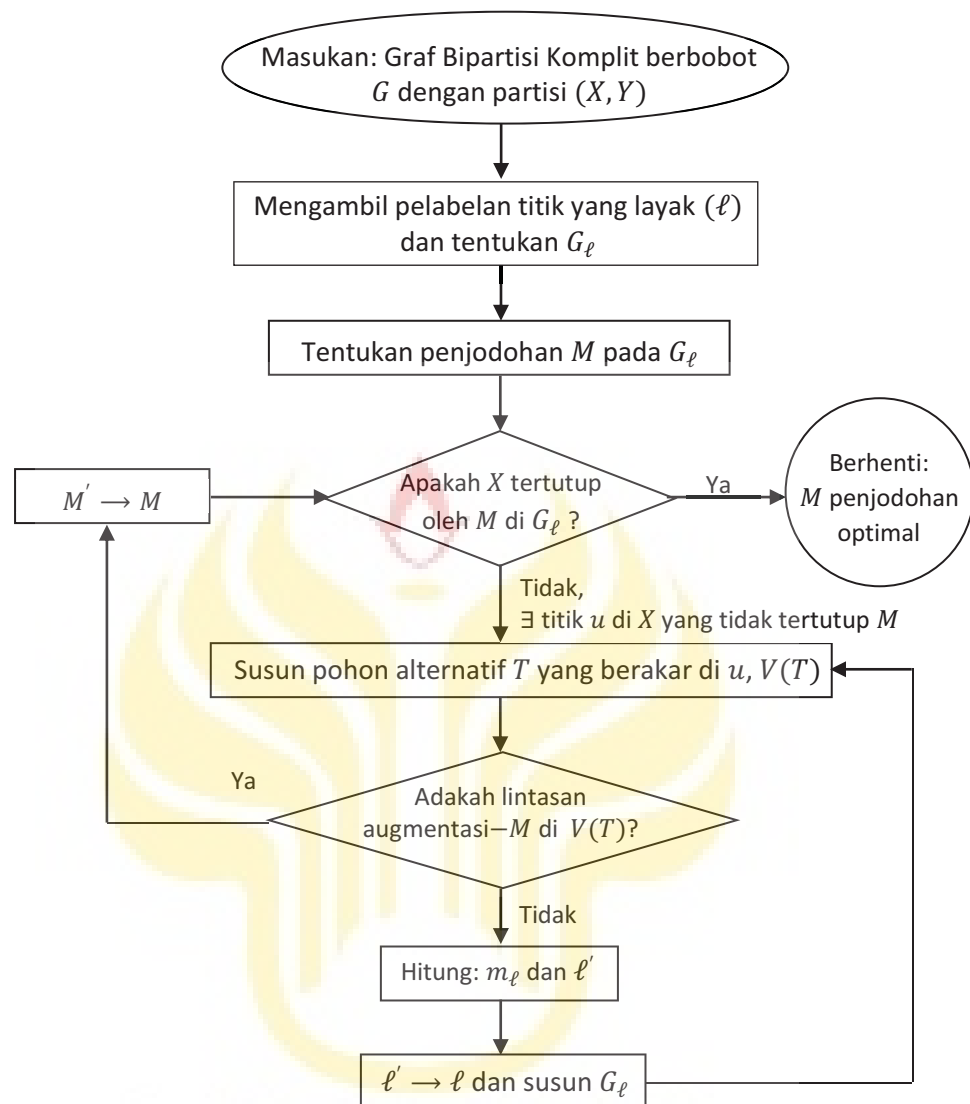
## 2.5 Algoritma Kuhn-Munkres

### 2.5.1 Deskripsi Algoritma Kuhn-Munkres

Diberikan sebarang graf bipartisi  $G$  dengan partisi  $(X, Y)$ , tujuan algoritma ini adalah menemukan penjadohan dari  $G$  yang menutup setiap titik  $X$  atau jika tidak menemukan penjadohan yang demikian, akan ditemukan himpunan bagian  $S$  dari  $X$  sehingga  $|N(S)| < |S|$ . Algoritma tersebut dikenal dengan nama Algoritma Hungarian. Algoritma Hungarian dipublikasikan oleh seorang matematikawan asal Amerika bernama Harold Kuhn pada tahun 1955 untuk menghormati dua orang matematikawan asal Hungaria, yaitu D. König dan E. Egerváry. Selanjutnya pada tahun 1957, oleh James Raymond Munkres seorang Profesor Emeritus matematika dari MIT merevisi algoritma Kuhn. Oleh karena itu, algoritma ini sering disebut Algoritma Kuhn-Munkres. Algoritma Kuhn-Munkres dapat digunakan untuk menentukan sebuah penjadohan optimal dalam graf bipartisi komplit berbobot (Budayasa, 2007:203).

### 2.5.2 Algoritma Kuhn-Munkres

Prosedur Algoritma Kuhn-Munkres dirangkum dalam diagram alir pada Gambar 2.19 sebagai berikut:



Gambar 2.19. Diagram Alir Algoritma Kuhn-Munkres

Secara sistematis langkah-langkah untuk mendapatkan penjadohan sempurna dengan bobot maksimum pada graf bipartisi komplit berbobot dengan Algoritma Kuhn-Munkres sebagai berikut (Chartrand & Oellermann, 1993:175):

Masukan : Graf bipartisi komplit berbobot  $G$  dengan partisi  $(X, Y)$ .

Langkah 1 :

Dimulai dengan sebuah pelabelan titik  $G$ , namakan pelabelan  $l$ .

- a) Untuk setiap  $x \in X$ , misal  $\ell(x) \leftarrow \text{maksimum } \{w(xy) | y \in Y\}$ .
- b) Untuk setiap  $y \in Y$ ,  $\ell(y) \leftarrow 0$ .
- c)  $G_\ell$  adalah graf bagian perentang dari  $G$  dengan himpunan sisi  $E_\ell$ .

Langkah 2 :

Pilih sebarang penjodohan  $M$  pada  $G_\ell$ .

Langkah 3 :

Langkah ini menentukan apakah penjodohan sempurna telah didapatkan.

Jika penjodohan  $M$  bukan penjodohan sempurna, maka disusun sebuah pohon alternatif  $T$  yang berakar dari suatu titik yang tidak tertutup oleh penjodohan  $M$  yang digunakan untuk mendefinisikan sebuah pelabelan titik baru  $\ell'$ .

- a) Jika  $X$  tertutup oleh  $M$ , maka  $M$  adalah penjodohan sempurna (karena  $|x| = |y|$ ), berdasarkan Teorema 2.5, penjodohan  $M$  optimal pada  $G_\ell$ , dan algoritma berhenti. Jika penjodohan  $M$  bukan penjodohan sempurna di  $G_\ell$ , maka dilanjutkan.
- b) Misal  $u$  adalah titik yang tidak tertutup oleh  $M$ ,  $u \in X$ .
- c) Susun pohon alternatif  $T$  dari  $M$  yang berakar di  $u$ . Jika terdapat lintasan augmentasi- $M$ , maka perpanjang  $M$  sepanjang lintasan augmentasi- $M$  untuk mendapatkan penjodohan baru dan kembali ke langkah 3.a, jika tidak memuat lintasan augmentasi- $M$ , maka  $T$  tidak dapat diperluas lebih jauh di  $G_\ell$  dan pelabelan titik  $\ell$  diganti dengan sebuah pelabelan titik baru  $\ell'$  dengan sifat bahwa  $M$  dan  $T$  termuat di graf  $G_\ell$ .

Langkah 4 :

Langkah ini menggunakan  $T$  untuk menghitung pelabelan titik baru  $\ell'$ .

Hitung  $m_\ell \leftarrow \min \{ \ell(x) + \ell(y) - w(xy) \mid x \in X \cap V(T) \text{ dan } y \in Y - V(T) \}$ .

Maka berlaku untuk  $\ell'$  :

$$\ell' \leftarrow \begin{cases} \ell(x) - m_\ell & \text{untuk } x \in X \cap V(T) \\ \ell(y) + m_\ell & \text{untuk } y \in Y \cap V(T) \\ \ell(x), \ell(y) & \text{lainya} \end{cases} \quad (2.15)$$

Langkah 5 :

Jika  $\ell \leftarrow \ell'$ , maka susun  $G_\ell$  baru dan kembali ke Langkah 3.c.

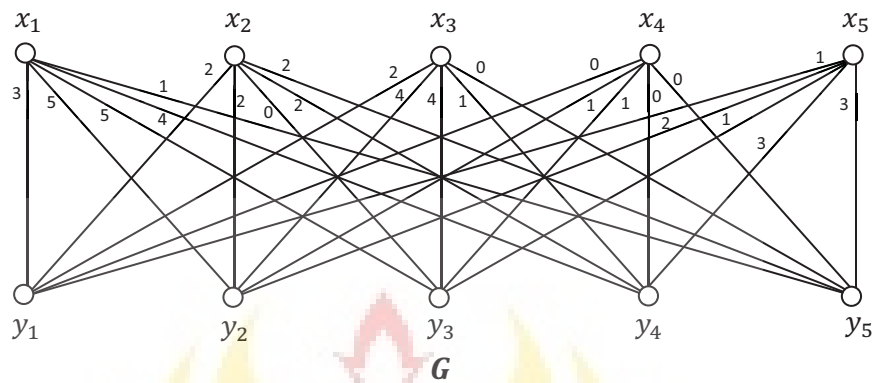
Penerapan Algoritma Kuhn-Munkres dapat diilustrasikan dengan graf bipartisi komplit berbobot  $G$  yang direpresentasikan dalam bentuk matriks  $M = [m_{ij}]$ , dengan  $m_{ij} = w(x_i y_j)$  untuk  $1 \leq i, j \leq 5$ .

Sebagai contoh, dalam suatu perusahaan terdapat 5 calon pekerja  $x_i$  ( $i = 1, 2, 3, 4, 5$ ) untuk 5 jabatan  $y_j$  ( $j = 1, 2, 3, 4, 5$ ). Tiap pelamar memiliki kemampuan yang berbeda-beda untuk setiap jabatan. Kemampuan pelamar kerja adalah bobot sisi yang menghubungkan pelamar kerja  $x_i$  dengan jabatan  $y_j$  yang dibentuk dalam matrik  $G = [m_{ij}]$  dengan  $m_{ij} = w(x_i y_j)$ .

$$G = \begin{matrix} & h & \begin{bmatrix} y_1 & y_2 & y_3 & y_4 & y_5 \end{bmatrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{bmatrix} 3 & 5 & 5 & 4 & 1 \\ 2 & 2 & 0 & 2 & 2 \\ 2 & 4 & 4 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 3 & 3 \end{bmatrix} \end{matrix}$$

Selanjutnya jika dimodelkan dengan graf bipartisi komplit berbobot  $G$  dengan partisi  $(X, Y)$ , maka  $X = \{x_1, x_2, x_3, x_4, x_5\}$ ,

$Y = \{y_1, y_2, y_3, y_4, y_5\}$ . Titik-titik  $x_i$  bertetangga dengan  $y_j$  jika dan hanya jika pekerja  $x_i$  dapat melakukan pekerjaan pada jabatan  $y_j$ .



Permasalahannya adalah bagaimana menempatkan calon pekerja ke jabatan yang sesuai dengan kemampuannya, sehingga diperoleh solusi yang optimal dengan Algoritma Kuhn-Munkres.

Dengan mengikuti langkah-langkah pada Algoritma Kuhn-Munkres didapatkan hasil sebagai berikut:

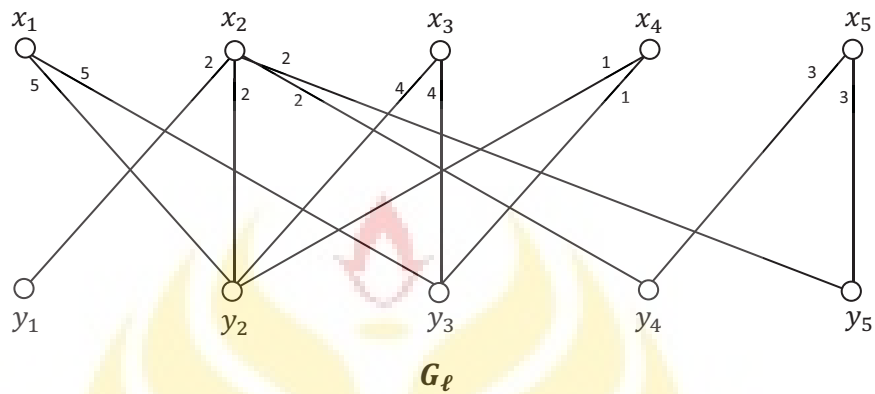
Masukan : Graf bipartisi komplit berbobot  $G$  dengan partisi  $(X, Y)$ .

Langkah 1:

Dimulai dengan sebuah pelabelan titik  $G$ , namakan pelabelan  $\ell$ .

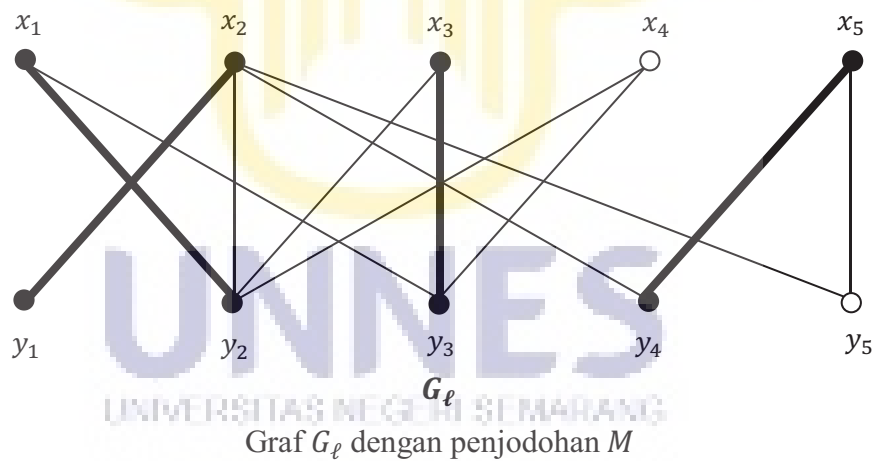
- $\forall x \in X$ , misal  $\ell(x) \leftarrow \text{maksimum } \{w(xy) | y \in Y\}$ , maka nilai pelabelan  $\ell(x_i) = \{5, 2, 4, 1, 3\}$ ,  $i = 1, 2, 3, 4, 5$ .
- $\forall y \in Y$ ,  $\ell(y) \leftarrow 0$ , maka nilai pelabelan  $\ell(y_i) = \{0, 0, 0, 0, 0\}$ ,  $i = 1, 2, 3, 4, 5$ .
- $G_\ell$  adalah graf bagian perentang dari  $G$  dengan himpunan sisi  $E_\ell$ .

$$G = \begin{array}{c} \begin{array}{ccccc} & y_1 & y_2 & y_3 & y_4 & y_5 & \ell(x_i) \\ x_1 & 3 & 5 & 5 & 4 & 1 & 5 \\ x_2 & 2 & 2 & 0 & 2 & 2 & 2 \\ x_3 & 2 & 4 & 4 & 1 & 0 & 4 \\ x_4 & 0 & 1 & 1 & 0 & 0 & 1 \\ x_5 & 1 & 2 & 1 & 3 & 3 & 3 \end{array} \\ \ell(y_i) & 0 & 0 & 0 & 0 & 0 \end{array}$$



Langkah 2 :

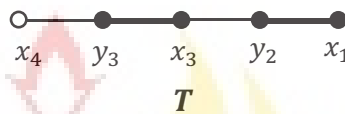
Pilih sebarang penjadohan  $M$  di  $G_\ell$ , misal  $M = \{x_1y_2, x_2y_1, x_3y_3, x_5y_4\}$ .



Langkah 3 :

- a) Ada titik di  $X$  yang tidak tertutup oleh  $M$ , maka lanjutkan.
- b) Titik  $x_4$  adalah titik di  $X$  yang tidak tertutup oleh  $M$ , sehingga  $x_4$  sebagai titik akar.

- c) Susun pohon alternatif  $T$  dari  $M$  yang berakar di  $x_4$ . Pohon alternatif diperoleh dengan  $V(T) = \{x_4, y_3, x_3, y_2, x_1\}$ , karena tidak ada lintasan augmentasi— $M$  dan  $T$  adalah pohon alternatif dari  $M$  yang berakar di  $x_4$  yang tidak dapat diperluas lebih jauh di  $G_\ell$ , maka pelabelan titik  $\ell$  diganti dengan sebuah pelabelan titik baru  $\ell'$ .



Langkah 4:

Menghitung label baru  $\ell'$  :

$$m_\ell \leftarrow \text{minimal } \{\ell(x) + \ell(y) - w(xy) \mid x \in X \cap V(T) \text{ dan } y \in Y - V(T)\}.$$

Anggota titik  $x \in X \cap V(T)$  adalah  $\{x_1, x_3, x_4\}$  dan anggota  $y \in Y - V(T)$  adalah  $\{y_1, y_4, y_5\}$ . Jadi perhitungan nilai  $m_\ell$  sebagai berikut:

$$m_\ell = \ell(x_1) + \ell(y_1) - w(x_1y_1) = 5 + 0 - 3 = 2$$

$$m_\ell = \ell(x_1) + \ell(y_4) - w(x_1y_4) = 5 + 0 - 4 = 1$$

$$m_\ell = \ell(x_1) + \ell(y_5) - w(x_1y_5) = 5 + 0 - 1 = 4$$

$$m_\ell = \ell(x_3) + \ell(y_1) - w(x_3y_1) = 4 + 0 - 2 = 2$$

$$m_\ell = \ell(x_3) + \ell(y_4) - w(x_3y_4) = 4 + 0 - 1 = 3$$

$$m_\ell = \ell(x_3) + \ell(y_5) - w(x_3y_5) = 4 + 0 - 0 = 4$$

$$m_\ell = \ell(x_4) + \ell(y_1) - w(x_4y_1) = 1 + 0 - 0 = 1$$

$$m_\ell = \ell(x_4) + \ell(y_4) - w(x_4y_4) = 1 + 0 - 0 = 1$$

$$m_\ell = \ell(x_4) + \ell(y_5) - w(x_4y_5) = 1 + 0 - 0 = 1$$

dari perhitungan di atas diperoleh  $m_\ell = 1$ , sehingga pelabelan baru  $\ell'$

sebagai berikut:

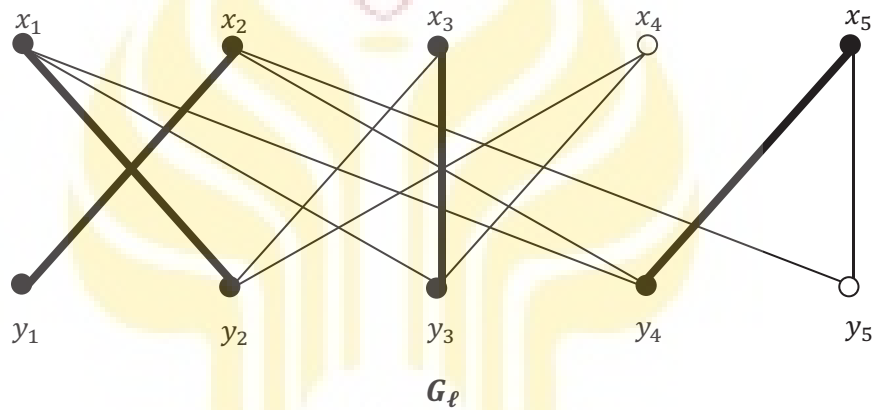


$$\ell' \leftarrow \begin{cases} \ell(x) - m_\ell & \text{untuk } x \in X \cap V(T) \\ \ell(y) + m_\ell & \text{untuk } y \in Y \cap V(T) \\ \ell(x), \ell(y) & \text{lainya} \end{cases}$$

Langkah 5 :

Karena  $\ell \leftarrow \ell'$ , maka buat  $G_\ell$  dan kembali ke Langkah 3.c.

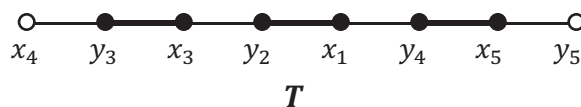
$$G = \begin{array}{c} h \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ \ell'(y_i) \end{array} \begin{array}{c} \left[ \begin{array}{ccccc} y_1 & y_2 & y_3 & y_4 & y_5 \\ 3 & 5 & 5 & 4 & 1 \\ 2 & 2 & 0 & 2 & 2 \\ 2 & 4 & 4 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 3 & 3 \end{array} \right] \ell'(x_i) \\ 4 \\ 2 \\ 3 \\ 0 \\ 3 \\ 0 \\ 3 \end{array}$$

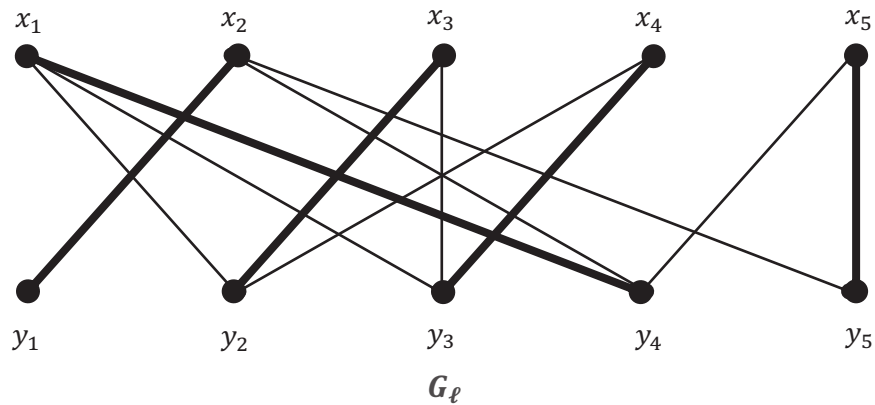


Graf  $G_\ell$  dengan pelabelan  $\ell'$

Langkah 3 :

- c) Susun pohon alternatif dari  $M$  yang berakar di  $x_4$ . Pohon alternatif diperoleh dengan  $V(T) = \{x_4, y_3, x_3, y_2, x_1, y_4, x_5, y_5\}$  karena lintasan augmentasi- $M$  didapatkan, maka perpanjang  $M$  sepanjang lintasan augmentasi- $M$  untuk mendapatkan penjadohan baru  $M'$  dan kembali ke Langkah 3.a.



Graf  $G_\ell$  dengan penjadohan  $M'$ 

Langkah 3 :

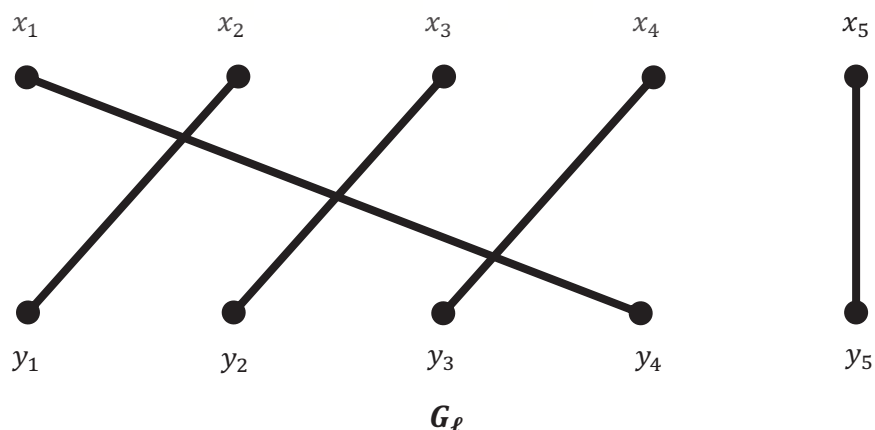
- a) Setiap titik di  $X$  tertutup oleh  $M'$ , maka  $M' = \{x_1y_4, x_2y_1, x_3y_2, x_4y_3, x_5y_5\}$  adalah penjadohan sempurna di  $G$  dan algoritma berhenti.

Keluaran :

Diperoleh  $M' = \{x_1y_4, x_2y_1, x_3y_2, x_4y_3, x_5y_5\}$  adalah penjadohan sempurna di  $G$  dengan besar bobot sebagai berikut:

$$\begin{aligned} w(M') &= \{w(x_1y_4), w(x_2y_1), w(x_3y_2), w(x_4y_3), w(x_5y_5)\} \\ &= (4 + 2 + 4 + 1 + 3) \\ &= 14. \end{aligned}$$

Jadi penempatan calon pekerja ke posisi jabatan pekerjaannya sebagai berikut:

Graf  $G_\ell$  dengan penjadohan  $M'$

## BAB V

### PENUTUP

#### 5.1 Simpulan

Berdasarkan hasil analisis dan pembahasan yang telah diuraikan, maka dapat disimpulkan hal-hal sebagai berikut:

1. Masalah penugasan *multi-objective* merupakan masalah penugasan yang memungkinkan untuk mengoptimalkan beberapa tujuan (*objectives*) secara bersamaan. Model matematis untuk masalah penugasan *multi-objective* dengan  $p$  tujuan ( $pAP$ ) dirumuskan sebagai berikut:

Minimumkan (maksimumkan):

$$z(X) = \{z_1(X), z_2(X), \dots, z_p(X)\}$$

dengan

$$z_k(X) = \sum_{i=1}^n \sum_{j=1}^n c_{ij}^k x_{ij}$$

untuk setiap  $k = 1, 2, \dots, p$

dengan kendala:

$$\sum_{j=1}^n x_{ij} = 1, \text{ untuk setiap } i = 1, 2, \dots, n$$

$$\sum_{i=1}^n x_{ij} = 1, \text{ untuk setiap } j = 1, 2, \dots, n$$

$$x_{ij} = \begin{cases} 1, & \text{jika tugas } j \text{ ditetapkan untuk pekerja } i. \\ 0, & \text{jika tugas } j \text{ tidak ditetapkan untuk pekerja } i. \end{cases}$$

2. Langkah-langkah penyelesaian masalah penugasan *multi-objective* dengan Algoritma Kuhn-Munkres adalah sebagai berikut:

1) Menyusun model matematika

a) Menentukan tipe dari masalah yaitu masalah maksimum atau minimum, dan melakukan rekontruksi bobot khusus untuk masalah minimum. Diketahui  $c_{ij}^k$  adalah nilai atau elemen baris ke- $i$ , kolom ke- $j$  dari data tujuan ke- $k$ , cara melakukan rekonstruksi bobot yaitu misalkan  $m_k = \text{maksimum} \{c_{ij}^k\}$  untuk setiap  $k = 1, 2, \dots, p$ , maka

$$c_{ij}^{k'} = m_k - c_{ij}^k$$

di mana  $c_{ij}^{k'}$  adalah data hasil rekonstruksi.

b) Merumuskan fungsi tujuan, merumuskan kendala, mendefinisikan variabel keputusan, dan persyaratan non-negatif

2) Normalisasi data

Normalisasi semua data dengan cara membagi setiap nilai dari masing-masing tujuan dengan nilai maksimum dari tujuan yang bersangkutan. Diketahui  $c_{ij}^k$  adalah nilai atau elemen baris ke- $i$ , kolom ke- $j$  dari data tujuan ke- $k$  (jika masalah minimum maka  $c_{ij}^k$  diganti dengan  $c_{ij}^{k'}$ ), cara normalisasi data yaitu misalkan  $m'_k = \text{maksimum} \{c_{ij}^k\}$  untuk setiap  $k = 1, 2, \dots, p$ , maka

$$c_{ij}^{k''} = \frac{c_{ij}^k}{m_{k'}}.$$

di mana  $c_{ij}^{k''}$  adalah data hasil normalisasi.

### 3) Modifikasi fungsi tujuan

#### a) Menentukan asumsi bobot tujuan

Pada masalah penugasan dengan  $p$  tujuan, jika  $\alpha_k$  adalah bobot tujuan ke- $k$  dengan

$$\sum_{k=1}^p \alpha_k = 1, \quad \text{untuk } k = 1, 2, \dots, p$$

maka  $\alpha_1 = \alpha_2 = \dots = \alpha_p = \frac{1}{p}$ .

#### b) Transformasi fungsi tujuan *multi-objective* ke dalam bentuk *single-objective*

Fungsi tujuan dimodifikasi menjadi seperti berikut:

Maksimumkan (minimumkan):

$$z(X) = \alpha_1 \sum_{i=1}^n \sum_{j=1}^n c_{ij}^1 x_{ij} + \alpha_2 \sum_{i=1}^n \sum_{j=1}^n c_{ij}^2 x_{ij} + \dots \\ + \alpha_p \sum_{i=1}^n \sum_{j=1}^n c_{ij}^p x_{ij}$$

### 4) Menyelesaikan masalah penugasan *single-objective* dengan Algoritma Kuhn-Munkres.

## 5.2 Saran

Berdasarkan penelitian yang telah dilakukan, saran yang dapat diberikan antara lain:

1. Masalah penugasan *multi-objective* yang dibahas dikenai kondisi banyaknya pekerja sama dengan banyaknya tugas, disarankan untuk mengkaji penyelesaian masalah penugasan *multi-objective* dengan Algoritma Kuhn-Munkres pada kasus banyaknya pekerja dan tugas tidak sama.
2. Pada penelitian ini, Algoritma Kuhn-Munkres yang diterapkan untuk menentukan penjadwalan sempurna pada graf bipartisi komplit berbobot dikerjakan secara manual, disarankan untuk membuat program yang dikombinasikan dengan pendekatan yang digunakan untuk menyelesaikan masalah penugasan *multi-objective*.

## DAFTAR PUSTAKA

- Bao, Chiao-Pin, T. Ming-Chi, & T. Meei-ing. 2007. A New Approach to Study The Multi-Objective Assignment Problem. *WHAMPOA – An Interdisciplinary Journal*, 53: 123-132.
- Belhouli, L., L. Galand, & D. Vanderpooten. 2014. An Efficient Procedure for Finding Best Compromise Solutions to the Multi-Objective Assignment Problem. *In Computers & Operations Research*. Elsevier.
- Bondy, J.A. & U. S. Murty. 1976. *Graph Theory with Applications*. New York: Elsevier Science Publishing Co., Inc.
- Budayasa, I Ketut. 2007. *Teori Graph dan Aplikasinya*. Surabaya: Unesa University Press.
- Chartrand, G. & L. Lesniak. 1996. *Graph and Digraph* (3<sup>rd</sup> ed.). USA: Chapman & Hall.
- Chartrand, G. & O. R. Oellermann. 1993. *Applied and Algorithmic Graph Theory*. New York: Mc Graw Hill, Inc.
- Deb K., A. Pratap, S. Agarwal, T. Meyarivan. 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *Journal of IEEE Transactions on Evolutionary Computation*, 6(2): 182–97.
- Garrett, J.D., J. Vannucci, R. Silva, D. Dasgupta, and J. Simien. 2007. Applying Hybrid Multiobjective Evolutionary Algorithms to the Sailor Assignment Problem. *In Advances in Evolutionary Computing for System Design*. Springer Verlag.
- Geetha, S. & K. P. K. Nair. 1993. A Variation of the Assignment Problem. *European Journal of Operational Research*, 68(3): 422-426.
- Grodzevich1, O. & O. Romanko. 2006. Normalization and Other Topics in MultiObjective Optimization. *Proceedings of the Fields–MITACS Industrial Problems Workshop*, 89-101.

- Gu, T., L. Chang, & Z. Xu. 2011. A Novel Symbolic Algorithm for Maximum Weighted Matching in Bipartite Graphs. *International Journal of Communications, Network and System Sciences*. 4:111-121.
- Hillier, S. F. & J. G. Lieberman. 2008. *Introduction To Operations Research* (9<sup>th</sup> ed.). New York: Mc Graw-Hill, Inc.
- Knowles, J.D. & D.W. Corne. 2000. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Journal of Evolutionary Computation*, 8(2): 149–72.
- Kuhn, H.W. 1955. The Hungarian Method for the Assignment Problem. *Journal of Naval Research Logistics Quarterly*, 2: 83-97.
- Lu, H. & G.G. Yen. 2003. Rank-Density-Based Multiobjective Genetic Algorithm and Benchmark Test Function Study. *Journal of IEEE Transactions on Evolutionary Computation*, 7(4): 325–43.
- Mills-Tettey, G. A., A. Stentz, & M. B. Dias. 2007. *The Dynamic Hungarian Algorithm for the Assignment Problem with Changing Cost*. Pittsburgh: Carnegie Mellon University.
- Munkres, J. 1957. Algorithms for the Assignment and Transportation Problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1): 32-38.
- Przybylski, A., X. Gandibleux, & M. Ehrgott. 2009. Computational Results for Four Exact Methods to Solve the Three-Objective Assignment Problem. In *Lecture Notes in Economics and Mathematical Systems*, 618: 79-88. Edited by V. Barichard, M. Ehrgott, X. Gandibleux, & V. T'Kindt. Berlin: Springer Verlag.
- Przybylski, A., X. Gandibleux, & M. Ehrgott. 2010. A Two Phase Method for Multi-Objective Integer Programming and Its Application to the Assignment Problem with Three Objectives. *Journal of Discrete Optimization*, 7: 149-165.
- Sarker, R., K-H. Liang, C. Newton. 2002. A New Multiobjective Evolutionary Algorithm. *European Journal of Operational Research*, 140(1): 12–23.
- Srinivas, N. & K. Deb. 1994. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Journal of Evolutionary Computation*, 2(3): 221–48.



- White, D.J. 1984. A Special Multi-Objective Assignment Problem. *The Journal of the Operational Research Society*, 35(8): 759-767.
- Yen, G.G. & H. Lu. 2003. Dynamic Multiobjective Evolutionary Algorithm: Adaptive Cell-Based Rank and Density Estimation. *Journal of IEEE Transactions on Evolutionary Computation*, 7(3): 253–74.
- Zitzler, E. & L. Thiele. 1999. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *Journal of IEEE Transactions on Evolutionary Computation*, 3(4): 257–71.

