

# Penjadwalan Petugas Medis pada Kondisi Darurat dengan Menggunakan *Binary Integer Programming* Berbasis Web

Bryan Alfadhori, Ahmad Saikhu, dan Victor Hariadi

Jurusan Tekni Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

*e-mail:* saikhu@its.sby.edu

**Abstrak**—Bencana alam maupun bencana yang disebabkan kelalaian manusia sering kali menimbulkan kondisi darurat. Penugasan petugas medis pada kondisi darurat merupakan hal yang sangat penting. Terbatasnya petugas dengan keahlian yang dibutuhkan yaitu kombinasi penugasan petugas medis yang tidak tepat dapat membuat penjadwalan yang tidak optimal. Dalam penentuan petugas medis yang memenuhi kondisi, digunakan representasi graf *bipartite* dan algoritma *Ford Fulkerson* dalam proses untuk pemilihan petugas medis yang memenuhi kondisi tersebut. *Binary integer programming* digunakan untuk menentukan kombinasi penugasan yang optimal. Berdasarkan hasil uji coba dapat disimpulkan bahwa kedua proses yang diimplementasikan dapat membantu dalam pengambilan keputusan penugasan. Representasi dari graf *bipartite* terbukti dapat memberikan hasil yang akurat berupa petugas medis yang memenuhi kondisi. *Binary Integer Programming* juga memberikan hasil yang optimal berupa petugas medis yang ditugaskan dan total jarak yang paling minimal.

**Kata Kunci**—graf *bipartite*, *binary integer programming*, kondisi darurat, penugasan petugas medis

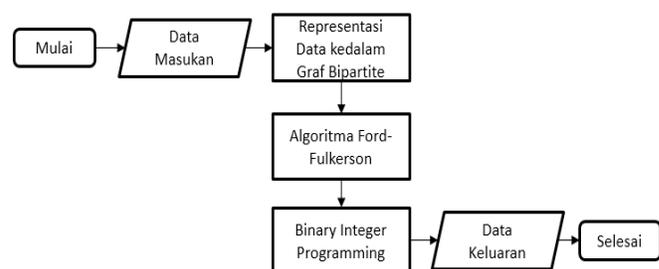
## I. PENDAHULUAN

KETERBATASAN jumlah dokter dengan kualifikasi tertentu sesuai yang diperlukan pada kondisi darurat dan tuntutan harus tiba di lokasi dengan cepat, maka diperlukan suatu sistem koordinasi penugasan yang handal agar situasi darurat dapat ditangani dengan baik. Koordinasi dan penjadwalan tersebut dapat dilakukan oleh pihak yang mempunyai otoritas seperti manajemen rumah sakit dengan mengembangkan sistem penjadwalan dan komunikasi antara pihak administrator dan dokter. Jika terjadi peristiwa yang mendesak seperti bencana alam atau panggilan lainnya menjadi cepat. Untuk pengambilan keputusan penugasan dengan tepat, didalam Sistem Pengambilan Keputusan (SPK) ada dua tahapan yang perlu dijalani yaitu pemilihan petugas medis dengan keahlian yang tepat dengan kondisi wilayah dan jarak serta biaya petugas medis untuk wilayah tersebut optimal.

Pada umumnya penugasan diberikan berdasarkan pada keahlian dari petugas medis. Kualifikasi petugas medis yang ditugaskan disebut kondisi. Kondisi ini berisi daftar keahlian dari petugas medis yang dibutuhkan. Dalam satu kondisi memiliki banyak keahlian. Biaya (*cost*) merupakan biaya yang dibutuhkan untuk petugas medis dalam tiap penugasan pada

wilayah tertentu. Penugasan petugas medis terjadi dalam satu waktu. Tiap petugas medis ditugaskan pada satu kondisi dan tiap kondisi ditangani oleh satu petugas medis. Jika terdapat kondisi darurat, petugas medis yang berjarak paling dekat akan ditugaskan dengan mempertimbangkan kualifikasi/spesialisasi petugas medis yang dibutuhkan pada keadaan khusus.

Dalam tugas akhir ini akan dibuat penerapan sistem tersebut dalam basis *web*. Petugas hanya perlu memberikan kondisi wilayah, keahlian petugas medis, jarak dan biaya untuk petugas medis tersebut pada tiap wilayah darurat. Sistem akan menerima masukan berupa *dataset* inputan yang akan di-*upload*. *Dataset* tersebut terdiri dari 4 bagian yaitu data petugas, data jarak, data keahlian, dan data kondisi. Pada sistem ini, ada beberapa proses yang dilakukan untuk mengambil keputusan. Hal ini meliputi untuk menentukan petugas yang memenuhi kondisi dengan menggunakan model graf *bipartite* dan algoritma *Ford-Fulkerson*. Sedangkan untuk menentukan petugas yang akan ditugaskan dengan menggunakan model *Binary Integer Programming* (Gambar 1).



Gambar 1. *Flow Chart* Sistem

## II. DESAIN DATA

### A. *Desain Data Masukan*

Data yang digunakan dibedakan menjadi dua yaitu data masukan dan data keluaran (Lihat Gambar 1). Data masukan terdiri dari 4 data yaitu data petugas, data jarak, data keahlian, dan data kondisi. Semua data masukan pada Tugas Akhir ini menggunakan ekstensi *file .csv (comma delimiter)*.

Data keahlian berisikan keahlian-keahlian yang tersedia dan setiap awalan kata dimulai dengan huruf besar. Jika terdapat

dua kata atau lebih, maka penulisannya digabung dan setiap awalan kata ditulis dengan huruf besar (lihat Gambar 2).

```

Bedah
Jantung
DokterUtama
Koordinator
    
```

Gambar 2. Data Keahlian

Data Petugas berisikan keahlian yang dimiliki dari setiap petugas. Satu petugas dapat memiliki lebih dari satu keahlian. Format penulisan data petugas dapat dilihat pada Gambar 3.

```

Bedah
Jantung,Bedah
Bedah,DokterUtama
Jantung,Bedah,Dokterutama,Koordinator
    
```

Gambar 3. Data Petugas

Data Kondisi berisikan keahlian-keahlian petugas yang dibutuhkan. Satu kondisi dapat memiliki lebih dari satu keahlian yang dibutuhkan. Bila kondisi memiliki lebih dari satu keahlian, maka penulian dihubungkan dengan kata 'dan' maupun kata 'atau' dalam bentuk *postfix* (Gambar 4).

```

Bedah
Jantung Bedah dan
Bedah DokterUtama dan
Jantung Bedah dan Bedah DokterUtama dan atau
    
```

Gambar 4. Data Kondisi

Data Jarak berisikan jarak petugas dengan lokasi kondisi. Data jarak memiliki bentuk matriks dua dimensi  $a \times b$  dimana  $a$  merupakan jumlah petugas yang tersedia dan  $b$  adalah jumlah kondisi yang ada. Dari contoh data petugas (Gambar 3) dan contoh data kondisi (Gambar 4) dapat diketahui bahwa petugas yang tersedia ada 4 dan jumlah kondisi adalah 4 maka matriks pada data jarak adalah  $4 \times 4$  dalam bentuk *comma delimiter*. Format penulisan data jarak dapat dilihat pada Gambar 5.

```

2,14,10,15
3,15,67,12
4,5,19,20
3,7,9,16
    
```

Gambar 5. Data Jarak

**B. Desain Data Keluaran**

Data keluaran dihasilkan melalui proses penugasan berupa matriks  $a \times b$ , dimana  $a$  merupakan jumlah petugas dan  $b$  adalah jumlah kondisi, yang berisikan 1 bila ditugaskan atau 0 bila tidak ditugaskan. Data keluaran juga menunjukkan biaya atau jarak untuk setiap petugas yang ditugaskan dan total dari jarak atau biaya yang dibutuhkan.

**III. PERANCANGAN ALGORITMA**

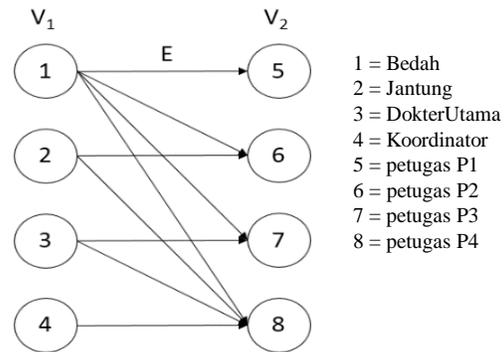
**A. Representasi Graf Bipartite**

Permodelan keahlian dan petugas diubah kedalam bentuk graf *bipartite*. Dimisalkan  $G(V,E)$  merupakan graf *bipartite* untuk keahlian dan petugas, maka :

1.  $V_1$  merupakan himpunan *node* mewakili keahlian.
2.  $V_2$  merupakan himpunan *node* mewakili petugas.
3.  $E$  merupakan himpunan *edge* yang menghubungkan *node*  $V_1$  dengan *node*  $V_2$ .

Untuk mempermudah penjelasan, digunakan Data Petugas (lihat Gambar 3) dan Data Keahlian (Gambar 2). Berdasarkan data pada gambar 3 terdapat 4 orang petugas dengan keahliannya masing-masing dan pada Gambar 2 terdapat 4 keahlian. Permodelan graf *bipartite* yang dilakukan adalah sebagai berikut (Gambar 6):

1.  $V_1$  terdiri dari *node* 1 sampai 4 yang mewakili data keahlian.
2.  $V_2$  terdiri dari *node* 5 sampai 8 yang mewakili data petugas.
3.  $E$  menghubungkan satu keahlian dengan satu petugas yang memiliki keahlian tersebut.



Gambar 6. Permodelan Graf Bipartite

**B. Algoritma Ford Fulkerson**

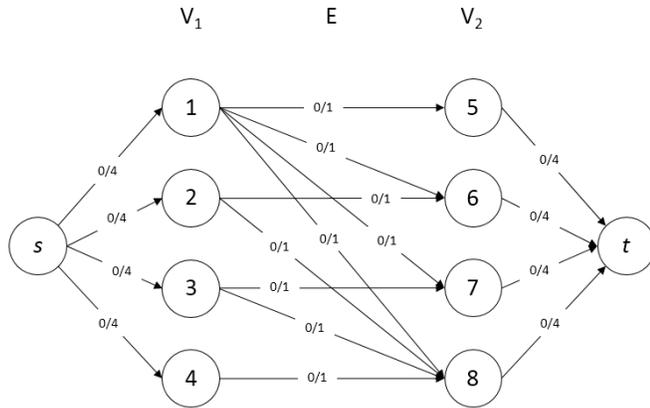
Sebelum algoritma *Ford-Fulkerson* digunakan, permodelan graf *bipartite* diubah menjadi graf yang memiliki *source s*, *sink t*, dan menambahkan kapasitas  $c(u,v)$  dengan aturan sebagai berikut:

1. Setiap *edge* yang dimulai dari *node* pada  $V_1$  ke *node* pada  $V_2$  diberikan kapasitas  $c(u,v)$  dengan bobot 1.
2. Sebuah *source s* dibuat dan dihubungkan ke semua *node* pada  $V_1$  dengan kapasitas  $c(u,v)$  yang memiliki bobot sesuai dengan jumlah petugas pada  $V_2$ .
3. Sebuah *sink t* dibuat dan semua *node* pada  $V_2$  dihubungkan ke *sink t* dengan kapasitas  $c(u,v)$  yang memiliki bobot sesuai dengan jumlah keahlian pada  $V_1$ .

Dari graf *bipartite* pada gambar 6 diketahui bahwa terdapat 4 keahlian dan 4 orang petugas sehingga permodelan diubah menjadi permodelan graf baru sesuai dengan aturan diatas (lihat Gambar 7). Selanjutnya langkah-langkah algoritma *Ford-Fulkerson* dilakukan sebagai berikut:

1. Inisialisasi nilai  $f(u,v) \leftarrow 0$  dan  $f(v,u) \leftarrow 0$ , untuk setiap *edge*  $(u,v)$ .

2. Perulangan selama terdapat lintasan augmenting  $p$  dari  $s$  menuju  $t$  dengan kapasitas tersisa  $c_f(u,v) > 0$  untuk semua  $(u,v) \in p$  :
  - a. Temukan  $c_f(p) = \min[c_f(u,v) | (u,v) \in p]$
  - b. Untuk setiap *edge*  $(u,v) \in p$ 
    - i.  $f(u,v) \leftarrow f(u,v) + c_f(p)$
    - ii.  $f(u,v) \leftarrow -f(v,u)$



Gambar 7. Permodelan Graf Baru

Hasil dari proses algoritma *Ford-Fulkerson* dimasukkan ke vektor dalam bentuk himpunan keahlian dengan elemennya yaitu petugas-petugas yang memiliki keahlian tersebut agar mempermudah proses selanjutnya.

**C. Binary Integer Programming**

Vektor himpunan keahlian dari proses *Ford-Fulkerson* diproses untuk menentukan petugas yang memenuhi kondisi. Karena ada kondisi yang memiliki lebih dari satu keahlian dan dihubungkan dengan kata ‘dan’ dan ‘atau’ maka perlu dilakukan langkah-langkah berikut:

1. Perulangan sebanyak jumlah kondisi yang melakukan proses *load data* kondisi dalam bentuk postfix.
2. Setiap melakukan proses *load, string* kondisi dipisah kedalam *token*.
3. Perulangan sebanyak jumlah *token* yang melakukan pengecekan untuk setiap *token* merupakan *operator* atau *operand*.
  - a. Jika *token* merupakan *operator* maka *token* tersebut diproses untuk menentukan apakah berupa *operator* ‘dan’ atau ‘atau’.
  - b. Jika *token* merupakan *operand* maka dilakukan pencarian petugas yang memiliki keahlian sesuai dengan *token* tersebut dan dimasukkan kedalam vektor.
4. Hasil data petugas yang memenuhi kondisi ditampung kedalam vektor.

Setelah vektor dibuat, pengecekan data jarak dilakukan untuk memuat data jarak petugas dengan kondisi yang telah sesuai. Data tersebut kemudian dibuat dalam bentuk dokumen dengan format *Jarak.txt* untuk mempermudah proses berikutnya yaitu penentuan petugas yang ditugaskan dengan menggunakan *Binary Integer Programming* yang mengadopsi Mansi S. Gaglani’s *algorithm* [1] tentang alternatif *Assignment*

*Problem. Binary Integer Programming* melakukan *load* dokumen *Jarak.txt* dan memuat data tersebut ke dalam sebuah *array Problem*. Selanjutnya *array problem* tersebut diproses untuk menentukan penugasan petugas ke kondisi yang sesuai.

Proses penentuan penugasan petugas dilakukan dengan langkah-langkah sebagai berikut:

1. Matriks *Array mincost* (*jumlah petugas x jumlah kondisi*) dibuat.
2. Matriks *Array penugasan* (*jumlah petugas x I*) dibuat.
3. Lakukan perulangan yang melakukan langkah 4 sampai langkah 9.
4. *Array mincost* diisi dengan bobot 1 untuk jarak yang paling kecil. Terdapat kemungkinan dimana terdapat jarak terkecil yang sama pada satu petugas.
5. Dari *array mincost*, dipilih petugas yang memiliki satu (*unique*) pilihan kondisi.
6. Lakukan pengecekan apakah kondisi tersebut dimiliki oleh petugas lain atau tidak.
7. Jika tidak ada petugas lain yang memiliki kondisi yang sama, maka kondisi tersebut ditambahkan kedalam matriks penugasan sesuai dengan petugasnya dan menandai bahwa petugas dan kondisi tersebut telah ditentukan.
8. Jika ada petugas lain yang memiliki kondisi yang sama, maka dilakukan:
  - a) Cari jarak terkecil kedua pada setiap petugas yang kemudian lakukan pengurangan pada jarak terkecil pertama dengan jarak terkecil kedua untuk mendapatkan selisih dari kedua jarak tersebut.
  - b) Lakukan pengecekan petugas mana yang memiliki selisih jarak terbesar.
  - c) Jika ada dua petugas atau lebih yang memiliki selisih jarak yang sama, maka lakukan langkah a) kembali sampai dapat menemukan satu petugas saja.
  - d) Jika terdapat satu petugas yang memiliki selisih jarak terbesar, maka kondisi tersebut ditambahkan kedalam matriks penugasan sesuai dengan petugasnya dan menandai bahwa petugas dan kondisi tersebut telah ditentukan.
9. Lakukan pengecekan apakah semua kondisi telah memiliki petugas. Jika tidak, maka lakukan langkah 2 sampai langkah 7. Jika iya, maka hentikan perulangan.

**IV. UJI COBA DAN EVALUASI**

Uji coba dan evaluasi dilakukan terhadap implementasi model graf *bipartite* dan *Binary Integer Programming* yang dilakukan. Terdapat dua tahapan dari uji coba yaitu:

1. Uji coba menentukan petugas yang memenuhi kondisi dengan graf *bipartite*.
2. Uji coba terhadap *Binary Integer Programming* yang dilakukan untuk melihat apakah hasil tersebut sesuai dengan hasil graf *bipartite* yang didapat dari algoritma *Ford-Fulkerson*.

Terdapat *dataset* sebagai berikut:

Tabel 1.  
Data Keahlian

| No. | Keahlian    |
|-----|-------------|
| 1   | Bedah       |
| 2   | Jantung     |
| 3   | DokterUtama |
| 4   | Koordinator |

Tabel 2.  
Data Petugas

| Petugas | Keahlian                                 |
|---------|------------------------------------------|
| P1      | Bedah                                    |
| P2      | Jantung, Bedah                           |
| P3      | Bedah, DokterUtama                       |
| P4      | Jantung, Bedah, DokterUtama, Koordinator |

Tabel 3.  
Data Kondisi

| No. | Kondisi                                      |
|-----|----------------------------------------------|
| K1  | Bedah                                        |
| K2  | Jantung Bedah dan                            |
| K3  | Bedah DokterUtama dan                        |
| K4  | Jantung Bedah dan Bedah DokterUtama dan atau |

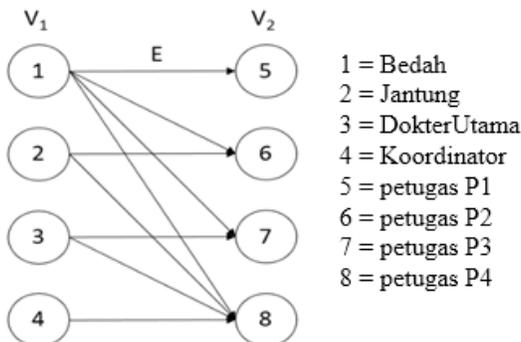
Tabel 4.  
Data Jarak

|    | K1 | K2 | K3 | K4 |
|----|----|----|----|----|
| P1 | 2  | 14 | 10 | 15 |
| P2 | 3  | 15 | 17 | 12 |
| P3 | 4  | 5  | 18 | 20 |
| P4 | 3  | 7  | 9  | 16 |

Proses penentuan petugas yang memenuhi kondisi dilakukan dengan menggunakan model graf *bipartite* ditunjukkan pada Tabel 5, representasi data menjadi model graf *bipartite* ditunjukkan pada Gambar 7, hasil pengelompokan petugas berdasarkan keahliannya.

Tabel 5.  
Pengelompokan Petugas dan Keahlian

| Keahlian    | Petugas |
|-------------|---------|
| Bedah       | 1,2,3,4 |
| Jantung     | 2,4     |
| DokterUtama | 3,4     |
| Koordinator | 4       |



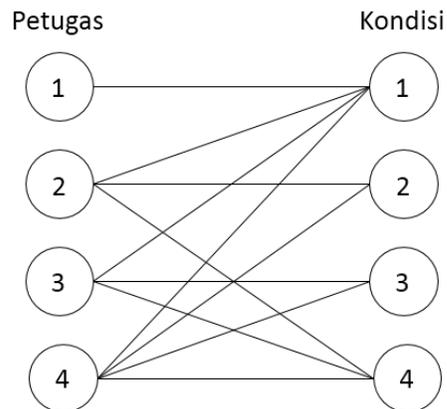
Gambar 7. Graf *bipartite* hasil pengelompokan

Dari hasil permodelan tersebut, dilakukan Algoritma *Ford-*

*Fulkerson* yang menghasilkan data jarak (Tabel 6) keluaran dari proses untuk menentukan petugas yang memenuhi kondisi dengan menggunakan graf *bipartite* (Gambar 8). Petugas yang tidak memenuhi kondisi diberi nilai 100. Pengecekan uji coba tahap pertama dapat dilakukan yaitu dengan melihat keahlian yang dimiliki petugas sudah sesuai dengan kondisi yang ada.

Tabel 6.  
Data Jarak Hasil Algoritma Ford Fulkerson

|    | K1 | K2  | K3  | K4  |
|----|----|-----|-----|-----|
| P1 | 2  | 100 | 100 | 100 |
| P2 | 3  | 15  | 100 | 12  |
| P3 | 4  | 100 | 18  | 20  |
| P4 | 3  | 7   | 9   | 16  |



Gambar 8. Graf *bipartite* hasil *Ford-Fulkerson*

Hasil keluaran dari *Binary Integer Programming* pada Tabel 7 dan Tabel 8 menunjukkan bahwa petugas yang ditugaskan telah sesuai dengan kondisi dan jarak antara petugas dengan kondisi tersebut merupakan jarak yang terkecil. Hal ini dapat dilihat pada graf *bipartite* (Gambar 8) yang merupakan keluaran dari algoritma *Ford-Fulkerson* dengan total jarak 39.

Tabel 7.  
Data Hasil keluaran

|    | K1 | K2 | K3 | K4 |
|----|----|----|----|----|
| P1 | 1  | 0  | 0  | 0  |
| P2 | 0  | 0  | 0  | 1  |
| P3 | 0  | 0  | 1  | 0  |
| P4 | 0  | 1  | 0  | 0  |

Tabel 8.  
Data Kondisi dengan Petugas dan Jaraknya

| Petugas | Kondisi | Jarak/Biaya |
|---------|---------|-------------|
| P1      | K1      | 2           |
| P2      | K4      | 12          |
| P3      | K3      | 18          |
| P4      | K2      | 7           |

Evaluasi berdasarkan hasil uji coba dapat disimpulkan bahwa penugasan petugas ditentukan berdasarkan dari biaya atau jarak yang paling minimal. Hal ini ditunjukkan dengan membandingkan hasil keluaran yang didapat pada tabel 7 dengan graf *bipartite* yang diajukan (Gambar 8). Dari dua hal

tersebut dapat dilihat bahwa hasil keluaran sesuai dengan graf yang dibuat yaitu pemasangan petugas dengan kondisi yang ada telah sesuai dan untuk setiap kondisi yang ditugaskan memiliki biaya atau jarak yang minimal.

## V. KESIMPULAN

Dari hasil ujicoba yang telah dilakukan, dapat diambil kesimpulan berdasarkan evaluasi sebagai berikut:

1. Model Graf *Bipartite* dapat memberikan hasil yang akurat berupa petugas medis yang memenuhi kondisi. Hal ini dapat mempermudah manajemen petugas medis.
2. *Binary Integer Programming* dengan Mansi S. Gaglani's *algoritim* [1] tentang alternatif *assignment problem* dapat menghasilkan hasil yang optimal berupa total jarak terpendek. Hal ini dapat mengefisiensikan biaya atau waktu untuk penugasan petugas medis.
3. Jumlah kemungkinan petugas maupun kondisi tidak berpengaruh secara signifikan terhadap performa dari graf *bipartite*, Algoritma *Ford-Fulkerson* maupun *binary integer programming*. Hal ini dapat mempermudah manajemen petugas medis dalam kondisi darurat.

## UCAPAN TERIMA KASIH

Penulis B.A. mengucapkan puji syukur kepada Allah SWT yang melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan penelitian dengan lancar. Penulis juga mengucapkan terima kasih kepada Bapak Ahmad Saikhu dan Bapak Victor Hariadi yang telah banyak membantu penulis dalam menyelesaikan penelitian ini. Penulis juga mengucapkan terima kasih kepada pihak-pihak lain yang turut membantu terselesaikannya penelitian ini.

## DAFTAR PUSTAKA

- [1] M. S. Gaglani, A Study on Transportation Problem, Transshipment Problem, Assignment Problem, and Supply Chain Management, Gujarat: Saurashtra University Theses Service, 2011.