

**APLIKASI ALGORITMA *BRANCH AND BOUND* UNTUK OPTIMASI JALUR
PEMADAM KEBAKARAN KOTA YOGYAKARTA**

Sri Margiyani¹, Noor Saif Muhammad Mussafi²

^{1,2}Program Studi Matematika Fakultas Sains dan teknologi UIN Sunan Kalijaga

Jl. Marsda Adisucipto No 1. Yogyakarta 55281

Email : om_norsa@yahoo.com

Abstrak

Kasus kebakaran di Indonesia, khususnya Kota Yogyakarta dari tahun ke tahun masih menunjukkan angka yang tinggi. Hal ini mengakibatkan kerugian yang cukup tinggi bagi korban kebakaran. Untuk meminimalisasi terjadinya korban jiwa dan kerugian secara material saat terjadi kebakaran, maka pihak pemadam kebakaran mengupayakan melalui rute terpendek untuk sampai di lokasi kebakaran. Tujuan dari penelitian ini adalah pencarian rute terpendek jalur pemadam kebakaran dari kantor pemadam sampai ke lokasi kebakaran.

Permasalahan pencarian rute terpendek jalur pemadam kebakaran secara abstrak dapat digambarkan dengan suatu graf yang merupakan masalah optimasi dalam pencarian rute terpendek (*Shortest Path Problem*). Pemecahan permasalahan tersebut adalah dengan merepresentasikan peta pemadam kebakaran ke dalam bentuk graf berbobot dan berarah, selanjutnya permasalahan diselesaikan menggunakan Algoritma *Branch and Bound*. Perhitungan dilakukan secara manual dengan jarak (dalam meter) sebagai bobot perhitungan.

Berdasarkan perhitungan menggunakan Algoritma *Branch and Bound* untuk optimasi jalur pemadam kebakaran Kota Yogyakarta untuk wilayah Kecamatan Umbulharjo menghasilkan solusi rute: Kantor pemadam kebakaran – Jln. Ipda Tut Harsono - Jln. Kusumanegara – Jln. Glagahsari – Kantor Kecamatan Umbulharjo) dengan total jarak 5305 meter atau 5,035 km.

Kata kunci : Algoritma *Branch and Bound*, Pencarian Rute Terpendek (*Shortest Path Problem*), optimasi rute

1. PENDAHULUAN

Peristiwa kebakaran merupakan bencana yang tidak bisa diprediksi sebelumnya. Peristiwa ini antara lain disebabkan oleh hubungan arus pendek listrik, puntung rokok yang dibuang di sembarang tempat, ledakan tabung gas LPG, dan lain-lain. Peristiwa kebakaran dapat menyebabkan kerugian yang cukup tinggi bagi korbannya. Sehingga dalam penanganannya harus tepat dan cepat.

Ketika kebakaran terjadi, pihak yang pertamakali dihubungi tentulah Dinas Pemadam Kebakaran. Untuk itu mobil pemadam kebakaran harus memilih jalur terpendek dari kantor pemadam kebakaran ke lokasi terjadinya kebakaran. Kota Yogyakarta mempunyai luas wilayah 32,50 km². Walaupun wilayahnya tidak terlalu luas, namun memiliki tata ruang dan

administrasi yang lengkap. Sehingga dalam penulisan ini wilayah yang akan dipilih adalah Kota Yogyakarta. Pencarian rute terpendek ini, dapat diselesaikan antara lain dengan Algoritma Branch and Bound. Algoritma Branch and Bound (B&B) adalah suatu algoritma umum untuk pencarian solusi optimal dari berbagai masalah optimasi, khususnya optimasi diskrit dan kombinatorial. B&B secara sistematis mengabaikan sekumpulan kandidat solusi yang tidak potensial menuju solusi optimal dengan menggunakan estimasi batas atas dan batas bawah (*upper and lower estimated bounds*) dari kuantitas yang dioptimasi. Metode ini pertama kali diusulkan oleh A. H. Land dan A. G. Doig pada tahun 1960 untuk *linear programming*.

2. LANDASAN TEORI

Algoritma *Branch and Bound* merupakan algoritma yang membagi permasalahan menjadi sub masalah lebih kecil yang mengarah ke solusi dengan pencabangan (*branching*) dan melakukan pembatasan (*bounding*) untuk mencapai solusi optimal. Pencabangan (*branching*) yaitu proses membentuk permasalahan ke dalam bentuk struktur pohon pencarian (*search tree*). Proses Pencabangan dilakukan untuk membangun semua cabang pohon yang menuju solusi, sedangkan proses pembatasan dilakukan dengan menghitung estimasi nilai (*cost*) simpul dengan memperhatikan batas. *Branch and Bound* pertama kali dikemukakan oleh A. H. Land dan A. G. Doig pada tahun 1960 (Suyanto, 2010:81).

Secara umum algoritma *Branch and Bound* dalam melakukan pencarian solusi menggunakan teknik *Least Cost Search* atau pencarian nilai terkecil, teknik ini akan menghitung nilai (*cost*) setiap simpul. Simpul yang memiliki nilai paling kecil dikatakan memiliki kemungkinan paling besar menuju solusi.

Setiap simpul aktif (*current node*) mempunyai sebuah nilai yang menyatakan nilai batas (*bound*). Sebuah simpul aktif merupakan simpul yang mempunyai nilai batas terkecil (karena teknik pencarian solusi *Least Cost Search*) (Eko Budi P.,2008:244).

Misal $c(i)$ adalah nilai estimasi lintasan minimum dari simpul i ke simpul tujuan, jadi $c(i)$ menyatakan batas (*bound*) nilai pencarian solusi dari simpul i . Sehingga dapat dirumuskan fungsi heuristik untuk menghitung nilai estimasi sebagai berikut :

$$c(i) = f(i) + g(i)$$

dengan:

$$c(i) = \text{nilai untuk simpul } i$$

$f(i)$ = nilai lintasan dari simpul akar ke simpul i

$g(i)$ = nilai untuk mencapai simpul tujuan dari simpul i

Untuk permasalahan yang lebih kompleks, sistem digambarkan dengan matriks A ukuran $n \times n$. Dalam hal ini simpul anak beserta nilai batasnya dimasukkan ke dalam sebuah antrian (*queue*) untuk dipilih simpul mana yang akan dieksplorasi. Simpul yang dieksplorasi adalah simpul dengan nilai batas terkecil. Nilai batas didapatkan dari reduksi baris dan kolom matriks A yang merepresentasikan graf. Reduksi dilakukan dengan mengurangi nilai c_{ij} pada baris atau kolom dengan nilai c_{ij} terkecil pada baris atau kolom tersebut, sedemikian sehingga didapatkan matriks tereduksi $A(t)$ dengan sebuah nilai nol pada setiap baris dan kolom. Menurut Suyanto (2010), sebuah matriks dikatakan tereduksi jika setiap kolom dan barisnya mengandung minimal satu nilai nol dan semua elemen lainnya non-negatif. Selanjutnya total nilai pereduksi menjadi nilai batas simpul akar. Untuk setiap simpul anak yang dibangkitkan dengan mengunjungi $A(i, j)$, dilakukan reduksi matriks $A(t)$ untuk mendapatkan matriks tereduksi $A(i, j)$. Sebelum melakukan proses reduksi matriks, nilai pada baris ke i , kolom ke j dan $A(j, \text{simpul awal})$ diubah menjadi ∞ .

Berdasarkan persamaan fungsi heuristik untuk menghitung nilai estimasi yang telah dijelaskan sebelumnya, maka nilai batas simpul anak dihitung dengan rumus:

$$\hat{c}(S) = \hat{c}(R) + A(i, j) + r$$

dimana :

$\hat{c}(S)$ = Nilai perjalanan minimum yang melalui simpul S , dimana S adalah anak dari simpul R .

$\hat{c}(R)$ = Nilai perjalanan minimum yang melalui simpul R , dimana R adalah simpul akar.

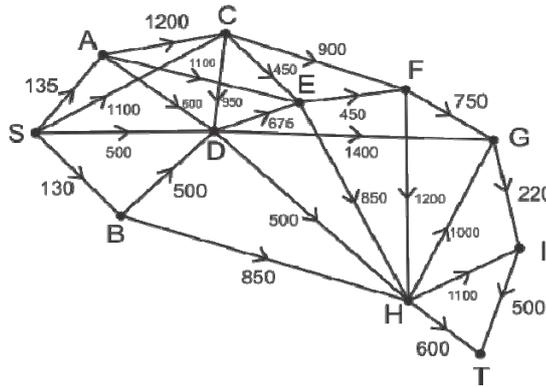
$A(i, j)$ = Bobot sisi (i, j) pada matriks tereduksi

r = Jumlah semua pereduksi pada proses reduksi matriks untuk simpul S .

Reduksi matriks simpul anak didasarkan pada matriks simpul induknya. Langkah ini dikerjakan berulang-ulang sampai semua simpul dikunjungi.

Langkah langkah untuk menjalankan algoritma *Branch and Bound* sebagai berikut:

1. Masukkan simpul akar ke dalam antrian Q . Jika simpul akar adalah simpul solusi (*goal node*), maka solusi telah ditemukan. Stop.
2. Jika Q kosong, tidak ada solusi. Stop.



Gambar 2. Graf Berbobot Kecamatan Umbulharjo

Selanjutnya langkah-langkah penerapan Algoritma *Branch and Bound* Untuk Optimasi Jalur Pemadam Kebakaran Kota Yogyakarta khususnya wilayah kecamatan Umbulharjo dapat disajikan sebagai berikut:

1. Representasikan graf kedalam bentuk matriks
2. Reduksi setiap baris dan setiap kolom, sehingga setiap baris dan setiap kolom mengandung minimal satu nilai 0. Sehingga didapat matriks tereduksi:

	S	A	B	C	D	E	F	G	H	I	T
S	∞	0	0	370	370	∞	∞	∞	∞	∞	∞
A	∞	∞	∞	0	0	500	∞	∞	∞	∞	∞
B	∞	∞	∞	∞	0	∞	∞	∞	350	∞	∞
C	∞	∞	∞	∞	500	0	450	∞	∞	∞	∞
D	∞	∞	∞	∞	∞	175	∞	900	0	∞	∞
E	∞	∞	∞	∞	∞	∞	0	∞	400	∞	∞
F	∞	0	450	∞	∞						
G	∞	0	∞								
H	∞	400	∞	500	0						
I	∞	0									
T	∞										

Dimisalkan matriks tereduksi diatas = matriks **T**

1. Simpul kedua yaitu A, lintasan S-A

- a. Ubah baris S dan kolom A pada matriks **T** menjadi ∞ .
- b. Ubah **T(A,S)** menjadi ∞ (elemen baris A kolom S pada matriks **T**).
- c. Reduksi baris dan kolom dengan nilai c_{ij} terkecil, karena tidak ada proses reduksi maka $r = 0$
- d. $\hat{C}(2) = \hat{c}(\text{root}) + A(S,A) + r = 5305 + 0 + 0 = \mathbf{5305}$

2. Simpul ketiga yaitu B, lintasan S-B

- a. Ubah baris S dan kolom A pada matriks **T** menjadi ∞ .
- b. Ubah **T(A,S)** menjadi ∞ (elemen baris A kolom S pada matriks **T**).
- c. Reduksi baris dan kolom dengan nilai c_{ij} terkecil, karena tidak ada proses reduksi maka $r = 0$
- d. $\hat{C}(3) = \hat{c}(\text{root}) + A(S,B) + r = 5305 + 0 + 0 = \mathbf{5305}$

3. Simpul keempat yaitu C, lintasan S-C

- a. Ubah baris S dan kolom C pada matriks **T** menjadi ∞ .
- b. Ubah **T(C,S)** menjadi ∞ (elemen baris C kolom S pada matriks **T**).
- c. Reduksi baris dan kolom dengan nilai c_{ij} terkecil, karena tidak ada proses reduksi maka $r = 0$
- d. $\hat{C}(4) = \hat{c}(\text{root}) + A(S,C) + r = 5305 + 370 + 0 = \mathbf{5675}$

4. Simpul kelima yaitu D, lintasan S-D

- a. Ubah baris S dan kolom D pada matriks **T** menjadi ∞ .
- b. Ubah **T(D,S)** menjadi ∞ (elemen baris D kolom S pada matriks **T**).
- c. Reduksi baris dan kolom dengan nilai c_{ij} terkecil,
 - Baris B dikurangi 350 sehingga $r = 350$
- d. $\hat{C}(5) = \hat{c}(\text{root}) + A(S,D) + r = 5305 + 370 + 350 = \mathbf{6025}$

5. Simpul keenam yaitu C, lintasan S-A-C

- a. Ubah semua elemen pada baris S, baris A dan kolom A, kolom C pada matriks **T** menjadi ∞ .
- b. Ubah **T(C,A)** menjadi ∞ (elemen baris C kolom A pada matriks **T**).
- c. Reduksi baris dan kolom dengan nilai c_{ij} terkecil, karena tidak ada proses reduksi maka $r = 0$
- d. $\hat{C}(6) = \hat{c}(\text{root}) + A(A,C) + r = 5305 + 0 + 0 = \mathbf{5305}$

6. Simpul ketujuh yaitu E, lintasan S-A-E

- a. Ubah semua elemen pada baris S, baris A dan kolom A, kolom E pada matriks **T** menjadi ∞ .
- b. Ubah **T(E,A)** menjadi ∞ (elemen baris E kolom A pada matriks **T**).
- c. Reduksi baris dan kolom dengan nilai c_{ij} terkecil,
Baris C dikurangi 450, sehingga $r = 450$
- d. $\hat{C}(7) = \hat{c}(\text{root}) + A(A,E) + r = 5305 + 500 + 450 = \mathbf{6255}$

7. Simpul kedelapan yaitu D, lintasan S-A-D

- a. Ubah semua elemen pada baris S, baris A dan kolom A, kolom D pada matriks **T** menjadi ∞ .
- b. Ubah **T(D,A)** menjadi ∞ (elemen baris D kolom A pada matriks **T**).
- c. Reduksi baris dan kolom dengan nilai c_{ij} terkecil, Baris B dikurangi 350, sehingga $r = 350$
- d. $\hat{C}(8) = \hat{c}(\text{root}) + A(A,D) + r = 5305 + 0 + 350 = \mathbf{5655}$

8. Simpul kesembilan yaitu D, lintasan S-B-D

- a. Ubah semua elemen pada baris S, baris B dan kolom B, kolom D pada matriks **T** menjadi ∞ .
- b. Ubah **T(D,B)** menjadi ∞ (elemen baris D kolom B pada matriks **T**).
- c. Reduksi baris dan kolom dengan nilai c_{ij} terkecil, karena tidak ada proses reduksi maka $r = 0$
- d. $\hat{C}(9) = \hat{c}(\text{root}) + A(B,D) + r = 5305 + 0 + 0 = \mathbf{5305}$

9. Simpul kesepuluh yaitu H, lintasan S-B-H

- a. Ubah semua elemen pada baris S, baris B dan kolom B, kolom H pada matriks **T** menjadi ∞ .
- b. Ubah **T(H,B)** menjadi ∞ (elemen baris H kolom B pada matriks **T**).
- c. Reduksi baris dan kolom dengan nilai c_{ij} terkecil, Baris D dikurangi 175, sehingga $r = 175$
- d. $\hat{C}(10) = \hat{c}(\text{root}) + A(B,H) + r = 5305 + 350 + 175 = \mathbf{5830}$

10. Simpul kesebelas yaitu D, lintasan S-A-C-D

- a. Ubah semua elemen pada baris S, baris A dan baris C serta kolom A, kolom C dan kolom D pada matriks **T** menjadi ∞ .

- b. Ubah $T(D,C)$ menjadi ∞ (elemen baris D kolom C pada matriks T).
- c. Reduksi baris dan kolom dengan nilai c_{ij} terkecil, Baris B dikurangi 350 dan kolom E dikurangi 175, sehingga $r = 525$
- d. $\hat{C}(11) = \hat{c}(\text{root}) + A(C,D) + r = 5305 + 500 + 525 = \mathbf{6330}$

11. Simpul duabelas yaitu E, lintasan S-A-C-E

- a. Ubah semua elemen pada baris S, baris A dan baris C serta kolom A, kolom C dan kolom D pada matriks T menjadi ∞ .
- b. Ubah $T(D,C)$ menjadi ∞ (elemen baris D kolom C pada matriks T).
- c. Reduksi baris dan kolom dengan nilai c_{ij} terkecil, karena tidak ada proses reduksi maka $r = 0$
- d. $\hat{C}(12) = \hat{c}(\text{root}) + A(C,E) + r = 5305 + 0 + 0 = \mathbf{5305}$

12. Simpul ketigabelas yaitu F, lintasan S-A-C-F

- a. Ubah semua elemen pada baris S, baris A dan baris C serta kolom A, kolom C dan kolom D pada matriks T menjadi ∞ .
- b. Ubah $T(D,C)$ menjadi ∞ (elemen baris D kolom C pada matriks T).
- c. Reduksi baris dan kolom dengan nilai c_{ij} terkecil, kolom E dikurangi 175, sehingga $r = 175$.
- d. $\hat{C}(13) = \hat{c}(\text{root}) + A(C,F) + r = 5305 + 450 + 175 = \mathbf{5930}$

13. Simpul keempatbelas yaitu E, lintasan S-B-D-E

- a. Ubah semua elemen pada baris S, baris A dan baris C serta kolom A, kolom C dan kolom D pada matriks T menjadi ∞ .
- b. Ubah $T(D,C)$ menjadi ∞ (elemen baris D kolom C pada matriks T).
- c. Reduksi baris dan kolom dengan nilai c_{ij} terkecil, Baris C dikurangi 450, dan Kolom H dikurangi 400, sehingga $r = 850$
- d. $\hat{C}(14) = \hat{c}(\text{root}) + A(D,E) + r = 5305 + 175 + 850 = \mathbf{6330}$

14. Simpul kelimabelas yaitu G, lintasan S-B-D-G

- a. Ubah semua elemen pada baris S, baris A dan baris C serta kolom A, kolom C dan kolom D pada matriks T menjadi ∞ .
- b. Ubah $T(D,C)$ menjadi ∞ (elemen baris D kolom C pada matriks T).
- c. Reduksi baris dan kolom dengan nilai c_{ij} terkecil, Baris F dikurangi 450 dan Kolom G dikurangi 400, sehingga $r = 850$

d. $\hat{C}(15) = \hat{c}(\text{root}) + A(D,G) + r = 5305 + 900 + 850 = \mathbf{7055}$

15. Simpul keenambelas yaitu H, lintasan S-B-D-H

- a. Ubah semua elemen pada baris S, baris B dan baris D serta kolom B, kolom D dan kolom H pada matriks **T** menjadi ∞ .
- b. Ubah **T(D,H)** menjadi ∞ (elemen baris D kolom H pada matriks **T**).
- c. Reduksi baris dan kolom dengan nilai c_{ij} terkecil, karena tidak ada proses reduksi maka $r = 0$
- d. $\hat{C}(16) = \hat{c}(\text{root}) + A(D,H) + r = 5305 + 0 + 0 = \mathbf{5305}$

16. Simpul ketujuhbelas yaitu F, lintasan S-A-C-E-F

- a. Ubah semua elemen pada baris S, baris A, baris C dan baris E serta kolom A, kolom C, kolom E dan kolom F pada matriks **T** menjadi ∞ .
- b. Ubah **T(E,F)** menjadi ∞ (elemen baris E kolom F pada matriks **T**).
- c. Reduksi baris dan kolom dengan nilai c_{ij} terkecil, karena tidak ada proses reduksi maka $r = 0$
- d. $\hat{C}(17) = \hat{c}(\text{root}) + A(E,F) + r = 5305 + 0 + 0 = \mathbf{5305}$

17. Simpul kedelapanbelas yaitu H, lintasan S-A-C-E-H

- a. Ubah semua elemen pada baris S, baris A, baris C dan baris E serta kolom A, kolom C, kolom E dan kolom F pada matriks **T** menjadi ∞ .
- b. Ubah **T(E,F)** menjadi ∞ (elemen baris E kolom F pada matriks **T**).
- c. Reduksi baris dan kolom dengan nilai c_{ij} terkecil, Baris D dikurangi 900, sehingga $r = 900$
- d. $\hat{C}(18) = \hat{c}(\text{root}) + A(E,H) + r = 5305 + 400 + 900 = \mathbf{6055}$

18. Simpul kesembilanbelas yaitu I, lintasan S-B-D-H-I

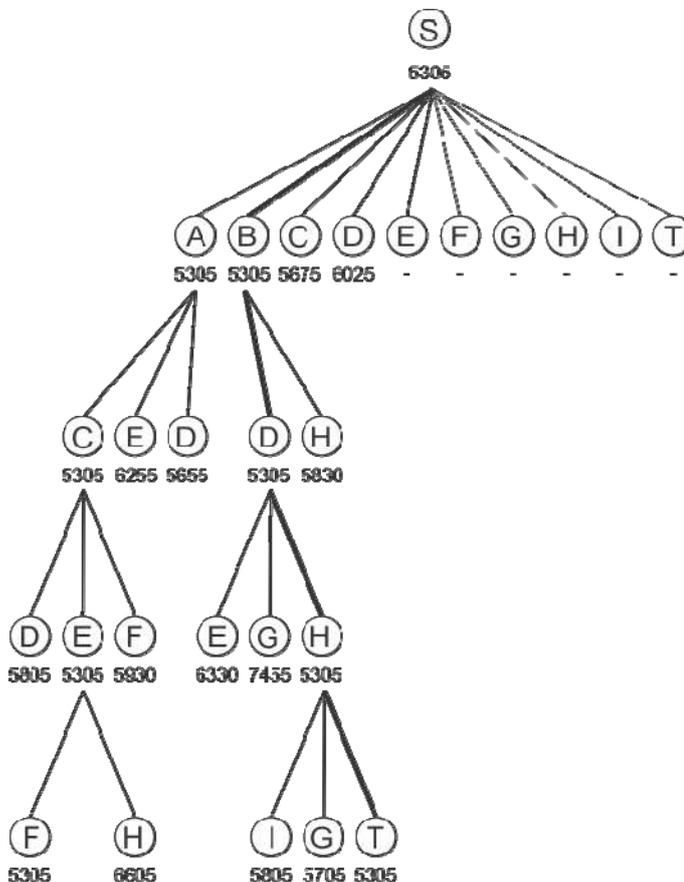
- a. Ubah semua elemen pada baris S, baris B, baris D dan baris H serta kolom B, kolom D, kolom H dan kolom I pada matriks **T** menjadi ∞ .
- b. Ubah **T(H,I)** menjadi ∞ (elemen baris H kolom I pada matriks **T**).
- c. Reduksi baris dan kolom dengan nilai c_{ij} terkecil, karena tidak ada proses reduksi maka $r = 0$
- d. $\hat{C}(19) = \hat{c}(\text{root}) + A(H,I) + r = 5305 + 500 + 0 = \mathbf{5805}$

19. Simpul kedua puluh yaitu G, lintasan S-B-D-H-G

- Ubah semua elemen pada baris S, baris B, baris D dan baris H serta kolom B, kolom D, kolom H dan kolom I pada matriks **T** menjadi ∞ .
- Ubah **T(H,I)** menjadi ∞ (elemen baris H kolom I pada matriks **T**).
- Reduksi baris dan kolom dengan nilai c_{ij} terkecil, karena tidak ada proses reduksi maka $r = 0$
- $\hat{C}(20) = \hat{c}(\text{root}) + A(H,G) + r = 5305 + 400 + 0 = 5705$

20. Simpul kedua puluh satu yaitu G, lintasan S-B-D-H-T

- Ubah semua elemen pada baris S, baris B, baris D dan baris H serta kolom B, kolom D, kolom H dan kolom T pada matriks **T** menjadi ∞ .
- Ubah **T(H,T)** menjadi ∞ (elemen baris H kolom T pada matriks **T**).
- Reduksi baris dan kolom dengan nilai c_{ij} terkecil, karena tidak ada proses reduksi maka $r = 0$
- $\hat{C}(21) = \hat{c}(\text{root}) + A(H,T) + r = 5305 + 0 + 0 = 5305$



Gambar 3. Branch and Bound Kecamatan Umbulharjo

Jadi jalur terpendek menurut Algoritma *Branch and Bound* dari kantor Pemadam Kebakaran Kota Yogyakarta sampai Kecamatan Umbulharjo menurut *branch and bound* gambar 3 adalah $5305 \text{ m} = 5,305 \text{ km}$.

4. KESIMPULAN DAN SARAN

Penerapan Algoritma *Branch and Bound* untuk optimasi jalur pemadam kebakaran Kota Yogyakarta dilakukan dengan cara merepresentasikan peta rute pemadam kebakaran dalam bentuk graf berbobot dan berarah. Sehingga dihasilkan solusi rute pemadam kebakaran dengan jarak minimal. Berdasarkan perhitungan dengan menggunakan Algoritma *Branch and Bound* diperoleh solusi rute pemadam kebakaran, Kantor pemadam kebakaran sampai Kecamatan Umbulharjo Rute S-B-D-H-T (Kantor pemadam kebakaran – Jln. Ipda Tut Harsono - Jln. Kusumanegara – Jln. Glagahsari – Kantor Kecamatan Umbulharjo) dengan total jarak 5305 meter atau 5,035 km. Berdasarkan penelitian yang telah dilakukan, ada beberapa saran untuk peneliti selanjutnya:

1. Penelitian ini hanya menggunakan Teknik FIFO B&B, diharapkan penelitian selanjutnya bisa menggunakan teknik lain seperti Teknik LIFO B&B dan Teknik LC- *Search* B&B.
2. Bobot yang digunakan dalam penelitian ini berupa jarak (dalam meter) dan mengabaikan kepadatan lalu lintas serta kondisi jalan. Diharapkan penelitian selanjutnya dapat menggunakan kepadatan lalu lintas atau kondisi jalan sebagai bobot dalam graf rute pemadam kebakaran.
3. Perhitungan yang dilakukan dalam penelitian ini masih secara manual. Diharapkan penelitian selanjutnya dapat melakukan perhitungan dengan pemrograman komputer.

5. DAFTAR PUSTAKA

- [1] Abdusakir dkk. (2009). *Teori Graf*. Malang: UIN-Malang Press.
- [2] Aldous, Joan M. and Wilson, Robin J. (2000). *Graph And Applications: An Introductory Approach*. Great Britain : Springer.
- [3] Anton, Howard (2000). *Dasar-dasar Aljabar* .Edisi 7.Jakarta: Interaksara
- [4] Bondy, J.A. dan Murty, U.S.R. (1976). *Graph Theory with Applications*. London: The Macmillan Press ltd.
- [5] Budi Purwanto, Eko (2008). *Perancangan dan Analisis Algoritma*. Yogyakarta : Graha Ilmu.
- [6] Levitin, Anany (2010). *Pengantar Desain dan Analisis Algoritma*. Edisi ke 2. Diterjemahkan oleh: Efrizal Zaida. Jakarta : Salemba Infotek.
- [7] Lipschutz, Seymour and Lipson, Marc (2008). *Schaum's Outlines MATEMATIKA DISKRET*. Edisi ke 3. Diterjemahkan oleh Thombi Layukallo. Jakarta: Erlangga.
- [8] Purwanto, Eko Budi (2008). *Perancangan dan Analisis Algoritma*. Yogyakarta: Graha Ilmu.
- [9] Rossen, Kenneth H. (2012). *Discrete Mathematics And Its Applications*. 7th ed. New York: The MCGraw-Hill Companies, Inc.
- [10] Saptono, F. dan Taufiq Hidayat. (2007). *Perancangan Algoritma Genetika Untuk Menentukan Jalur Terpendek*. Yogyakarta: SNATI

Pengujian Optimalisasi Jaringan Kabel *Fiber Optic* di Universitas Islam Indonesia Menggunakan *Minimum Spanning Tree*

- [11] Shofi A., Muhammad (2013). Implementasi Algoritma Branch and Bound untuk Optimasi Rute Pengangkutan Sampah Kota Yogyakarta. *Skripsi*. Yogyakarta: Program Studi Matematika Fakultas Sains dan Teknologi UIN Sunan Kalijaga.
- [12] Suryahadi (1990). *Pengantar Teori dan Algoritma Graph*. Jakarta: Penerbit Gunadarma.
- [13] Suyanto (2010). *Algoritma Optimasi (Deterministik atau Probabilistik)*. Yogyakarta: Graha Ilmu.
- [14] Wibisono, Samuel (2008). *Matematika Diskrit*. Yogyakarta: Graha Ilmu.