PERBANDINGAN PROGRAM DINAMIS DAN ALGORITMA *GREEDY* DALAM MENYELESAIKAN MASALAH *CHINESE POSTMAN PROBLEM*

Yudhi P M, Wamiliana dan Fitriani.

Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Lampung
Yudhi_pm@yahoo.com

Abstrak. Permasalahan *Chinese Postman Problem* adalah bagaimana seorang tukang pos akan mengantarkan surat ke alamat-alamat sepanjang jalan di suatu daerah dan bagaimana ia merencanakan rute perjalanannya supaya ia melewati setiap jalan tepat sekali dan kembali lagi ke tempat awal keberangkatannya dengan jarak / waktu / ongkos seminimal mungkin. Lintasan dan sirkuit yang digunakan dalam penelitian ini adalah lintasan dan sirkuit *Eule*r. Pada penelitian ini algoritma *Greedy* dibandingkan dengan Program Dinamis untuk menyelesaikan permasalahan *Chinese Postman* dengan menggunakan peta wilayah kelurahan Kedaton Bandar Lampung sebagai contoh. Hasil penelitian menunjukan bahwa metode Program Dinamis menghasilkan nilai yang lebih baik.

Kata Kunci. *Chinese Postman Problem*, Sirkuit *Euler*, algortima Greedy, program dinamis.

PENDAHULUAN

Masalah *Chinese Postman Problem* (CPP) pertama kali diformulasikan dalam bentuk masalah untuk menentukan jalan terpendek bagi seorang tukang pos untuk melewati semua jalan yang ada dan kembali ke tempat semula. Masalah ini dikemukakan oleh Kwan Mei-Ko di awal 1960-an dalam jurnal *Chinese Mathematics*. Dalam istilah graf definisi CPP adalah mencari lintasan pada suatu graf berbobot yang terhubung yang melewati semua sisi (minimal sekali) dengan jumlah bobot minimum dari suatu simpul kembali ke simpul awal . Pada permasalahan *Chinese Postman Problem* terdapat beberapa metode untuk menentukan jarak terpendek untuk memberikan solusi optimum pada permasalah tersebut seperti metode Program Dinamis dan Algoritma *Greedy*.

Graf G = (*V*,*E*) terdiri dari *V* = $\{v_1, v_2, \dots\}$ yang disebut *vertex / node* (titik) yang tidak kosong, dan *E* = $\{e_{ij}$; i, j $\in V\}$ yang unsur-unsurnya disebut *edge* (garis) yang boleh kosong .*Eulerian* erupakan graf yang mempunyai derajat genap untuk semua *vertex*nya dan setiap *edge*nya dilewati tepat satu kali. [1]

Lintasan *Euler* ialah lintasan yang melalui masing-masing *edge* di dalam graf tepat satu kali. Bila lintasan itu kembali ke simpul asal maka membentuk lintasan tertutup (sirkuit), sehingga lintasan tertutup itu dinamakan sirkuit *Euler*. [6]

Postman Tour dalam suatu graf *G* adalah *closed walk* yang melewati setiap *edge* nya dari G paling sedikit satu kali. [2]

Algoritma Greedy merupakan metode untuk menemukan solusi optimum dalam persoalan optimasi dengan solusi langkah per langkah.[4]

Program Dinamis (*dynamic programming*) adalah metode penyelesaian masalah dengan cara menguraikan solusi menjadi sekumpulan langkah (*step*) atau tahapan (*stage*) sedemikian sehingga solusi dari persoalan dapat dipandang dari serangkaian keputusan yang saling berkaitan.

Pada penyelesaian persoalan dengan metode ini terdapat sejumlah berhingga pilihan yang mungkin. Solusi pada setiap tahap dibangun dari hasil solusi tahap sebelumnya. Persyaratan optimasi dan kendala digunakan untuk membatasi sejumlah pilihan yang harus dipertimbangkan pada suatu tahap.

Tulisan ini disusun sebagai berikut : Bagian 1 berisi Pendahuluan, Bagian 2 berisi Metode Penelitian, Bagian 3 merupakn Hasil dan Pembahasan dan Bagian 4 merupakan Kesimpulan.

METODE PENELITIAN

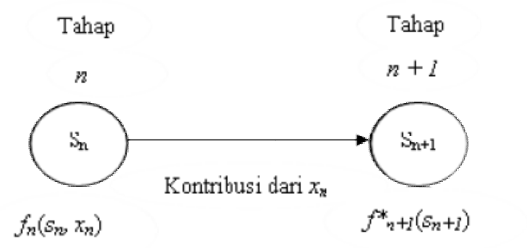
Adapun langkah-langkah yang digunakan dalam penelitian adalah sebagai berikut :

1. Tentukan semua *vertex* berderajat ganjil dalam graf.
2. Tentukan semua kemungkinan *pairing* pada *vertex* berderajat ganjil.
3. Tentukan *matching* yang memiliki bobot minimum dengan menggunakan algoritma *Greedy* dan program dinamis.
4. Tambahkan *edge* yang minimum pada langkah 3 terhadap graf
5. Hitung panjang lintasan yang optimal dari permasalahan *Chinese Postman* adalah jumlah dari semua *edge* graf awal ditambah dengan penambahan *edge* pada Langkah 4.
6. Tentukan / konstruksikan sirkuit yang menyatakan *Eulerian tour* dalam permasalahan *Undirected Chinese Postman Problem.*

1. Himpunan Kandidat
Himpunan ini berisi elemen-elemen yang memiliki peluang untuk pembentuk solusi
2. Himpunan Solusi
Himpunan ini berisi solusi dari permasalahan yang diselesaikan dan elemennya terdiri dari elemen dalam himpunan kandidat, namun tidak semuanya atau dengan kata lain himpunan solusi ini adalah bagian dari himpunan kandidat
3. Fungsi Seleksi
Fungsi seleksi adalah fungsi yang memilih kandidat yang paling memungkinkan dari himpunan kandidat untuk dimasukkan ke dalam himpunan solusi agar solusi optimal terbentuk. Kandidat yang sudah terpilih pada suatu langkah tidak akan dipertimbangkan lagi pada langkah selanjutnya
4. Fungsi Kelayakan Fungsi kelayakan adalah fungsi yang memeriksa apakah suatu calon yang terpilih akan menimbulkan solusi yang layak, yaitu kandidat tersebut bersama dengan himpunan solusi yang terpilih tidak akan melanggar kendala yang berlaku pada masalah.
5. Fungsi Obyektif

Fungsi obyektif adalah fungsi yang memaksimalkan atau meminimalkan nilai solusi. Tujuannya adalah memilih satu saja solusi terbaik dari masing-masing anggota himpunan solusi.[7]

Pada program dinamis, rangkaian keputusan yang optimal dibuat dengan menggunakan prinsip optimalitas. Prinsip optimalitas ini adalah : jika solusi total optimal, maka bagian solusi sampai tahap ke-k juga optimal. Prinsip optimalitas berarti bahwa jika bekerja dari tahap k ke tahap $k + 1$, maka dapat menggunakan hasil optimal dari tahap k tanpa harus kembali ke tahap awal. Ongkos pada tahap k +1 = (ongkos yang dihasilkan pada tahap k) + (ongkos dari tahap k ke tahap k + 1).[5]



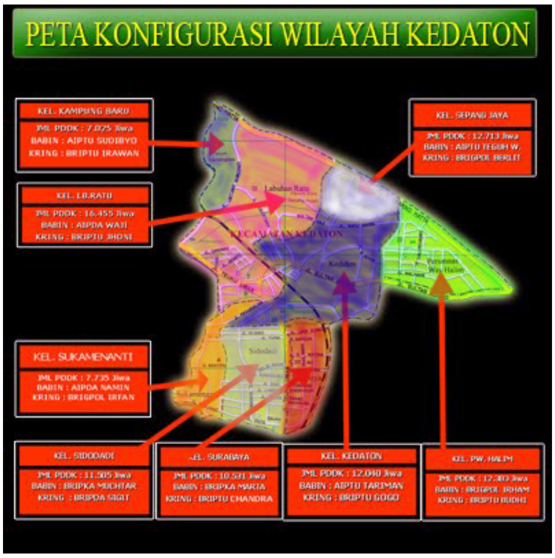
Pada tahap n proses akan berada pada suatu keadaan s_n. Pembuatan keputusan kebijakan x_n selanjutnya menggerakan proses keadaan s_{n+1} pada tahap (n + 1). Kontribusi sesudahnya terhadap fungsi tujuan di bawah kebijakan optimal telah dihitung sebelumnya sebagai $f_{n+1}*(s_{n+1})$. Keputusan kebijakan x_n juga memberi beberapa kontribusi kepada funngsi tujuan. Kombinasi kedua nilai ini dengan benar akan memberikan $f_n(s_n, x_n)$, yaitu kontribusi n tahap kedepan kepada fungsi tujuan .
Pengoptimalan terhadap x_n $f_n*(s_n) = f_n(s_n, x_n*)$. Setelah ditemukan x_n*dan $f_n*(s_n)$ untuk setiap nilai s_n, prosedur penyelesaian sekarang siap bergerak mundur satu tahap. Salah satu cara untuk mengkategorikan masalah pemrograman dinamis deterministik adalah dengan bentuk fungsi tujuan. Sebagai contoh, tujuannya mungkin adalah meminimumkan jumlah kontribusi dari tahap-tahap individual, atau untuk memaksimumkan jumlah tersebut, atau untuk meminimumkan suatu produk, dan seterusnya.[3]

HASIL DAN PEMBAHASAN

Berikut ini diberikan contoh kasus penyelesaian masalah *Chinese Postman Problem* pada graf tak berarah peta wilayah konfigurasi Kedaton Bandar Lampung

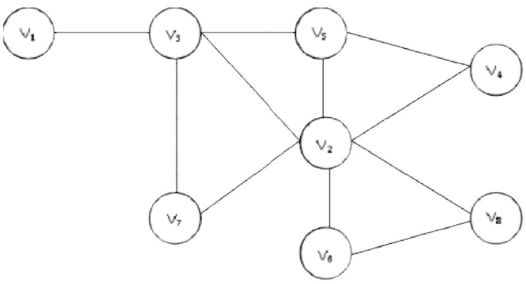
Tabel 1. Kelurahan Kedaton

No	Kode Pos	Kelurahan	Kecamatan
1	35143	Kampung Baru	Kedaton
2	35141	Kedaton	Kedaton
3	35142	Labuhan Ratu	Kedaton
4	35141	Perumnas Way Halim	Kedaton
5	35148	Sepang Jaya	Kedaton
6	35147	Sidodadi	Kedaton
7	35146	Sukamenanti	Kedaton
8	35148	Surabaya	Kedaton



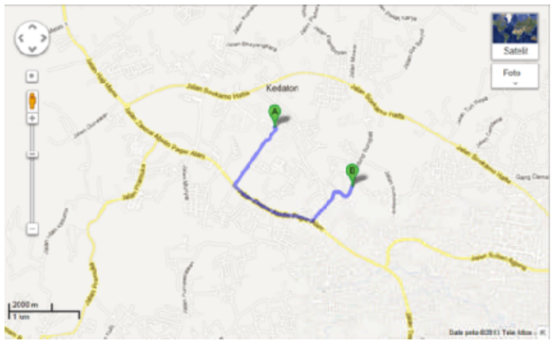
Gambar 1

Gambar di atas dapat direpresentasikan kedalam graf yang merepresentasikan peta konfigurasi wilayah Kedaton sebagai berikut



Gambar 2

Dengan menggunakan bantuan pencitraan *Google Map*, dapat diketahui jarak antara kelurahan yang terhubung berikut ini.



Gambar 3

Kelurahan Kampung Baru – Kelurahan Labuhan Ratu = 3.2 km

Kelurahan Sukamenanti – Kelurahan Labuhan Ratu = 3.5 km

Kelurahan Sepang Jaya – Kelurahan Labuhan ratu = 3.2 km

Kelurahan Kedaton – Kelurahan Labuhan ratu = 2.6 km

Kelurahan Sepang Jaya – Kelurahan Permunas Way halim = 1.6 km

Kelurahan Sepang jaya – Kelurahan Kedaton = 1.4 km

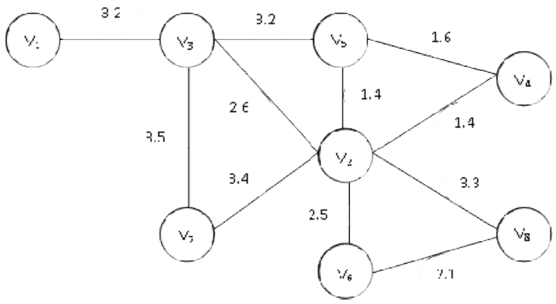
Kelurahan Perumnas Way halim – Kelurahan Kedaton = 1.4 km

Kelurahan Sidodadi – Kelurahan Kedaton = 2.5 km

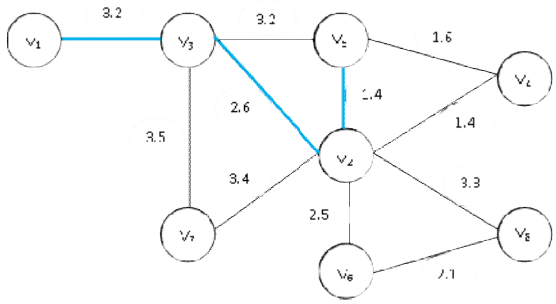
Kelurahan Surabaya – Kelurahan Kedaton = 3.3 km

Kelurahan Sukamenanti – Kelurahan kedaton = 3.4 km

Kelurahan Sidodadi – Kelurahan Surabaya = 2.1 km



Gambar 4



Gambar 6

Tahap-tahap Penyelesaian Masalah dengan Algoritma *Greedy*

1. Pada setiap langkah, ambil sisi yang berbobot minimum yang menghubungkan sebuah *vertex* yang sudah terpilih dengan sebuah *vertex* lain yang belum terpilih.
2. Lintasan dari *vertex* asal ke *vertex* yang baru haruslah merupakan lintasan yang terpendek diantara semua lintasannya ke *vertex-vertex* yang belum terpilih.

Berikut ini adalah tahap-tahap penelusuran jalur dengan menggunakan Algoritma *Greedy*:

Tahap 1 : Penelusuran dimulai dari *vertex* v_1 pada Gambar 16. Pada *vertex* v_1 terdapat 1 *vertex* tunggal yaitu v_3. Sehingga, perjalanan harus melalui *vertex* v_3 yang memiliki bobot yang bernilai 3,2

Tahap 2 : Dari *vertex* v_3 terdapat 3 *edge* yang terhubung pada *vertex* v_3 yaitu v_2 , v_5 , v_7. Dari ketiga *vertex* tersebut *vertex* yang memiliki bobot terkecil adalah *vertex* v_2 bernilai 2,6.

Tahap 3 : Dari *vertex* v_2 terdapat 6 *edge* yang terhubung pada *vertex* v_3 yaitu v_3 , v_4 , v_5 , v_6 , v_7 , v_8 . Dari keenam *vertex* tersebut yang dapat diseleksi utnuk dilalui adalah 5 *vertex,* karena terdapat satu *vertex* yang telah dilalui yaitu *vertex* v_8. Dari kelima *vertex* yang memiliki bobot terkecil adalah *vertex* v_4 dan v_5 bernilai 1,4, sehingga *vertex* yang dipilih adalah *vertex* v_5, karena *vertex* v_5 adalah *vertex* tujuan.

Total jarak yang dapat ditempuh dari *vertex* v_1 ke *vertex* v_5 dengan menggunakan algoritma Greedy adalah 7,2 dengan *path* v_1-v_3-v_2 – v_5.

Penyelesaian Masalah dengan Metode Program Dinamis

Misalkan $v_1, v_2, \ldots, v_4$ adalah simpul-simpul yang dikunjungi pada tahap k (k = 1, 2, 3, 4).
Pada persoalan ini,*Tahap* (k) adalah proses memilih simpul tujuan berikutnya

Status (s) yang berhubungan dengan masing-masing tahap adalah simpul-simpul di dalam graf. Relasi rekurensi berikut menyatakan lintasan terpendek :

$$f_1\ (s) = c_{x_1 s}$$

$$f_k\ (s) = \min_{x_k}\{c_{x_k s} + f_{k-1}(x_k)\}\ ,\ k = 2, 3, 4$$

Keterangan:

x_k : peubah keputusan pada tahap k (k = 2, 3, 4).

c_{sx_k} : bobot (*cost*) sisi dari s ke x_k

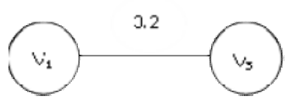
$f_k(s, x_k)$: total bobot lintasan dari s ke x_k

$f_k(s)$: nilai minimum dari $f_k(s, x_k)$

Tujuan program dinamis maju adalah untuk mendapatkan $f_2(v_5)$ dengan cara mencari f_1(s), f_2(s) terlebih dahulu.

Tahap 1 : $f_1(s) = c_{x1s}$

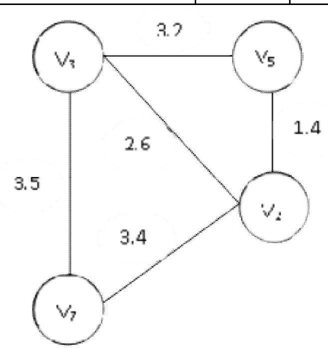
$s \backslash x_1$	Solusi optimum	
	$f_1(s)$	x_1^*
v_3	*3,2*	v_1



Gambar 7

Tahap 2 : $f_x(s) = \min_{x_2} \{c_{x2s} + f_1(x_2)\}$,

$s \backslash x_2$	$f_2(x_2, s) = cx_2s + f_1(x_2)$	Solusi optimum	
	v_3	$f_2(s)$	x_2^*
v_2	*5,8*	*5,8*	v_3
v_5	*6,4*	*6,4*	v_2
v_7	*6,7*	*6,7*	v_2



Gambar 8

Tabel diatas diperoleh dari hasil perhitungan berikut ini

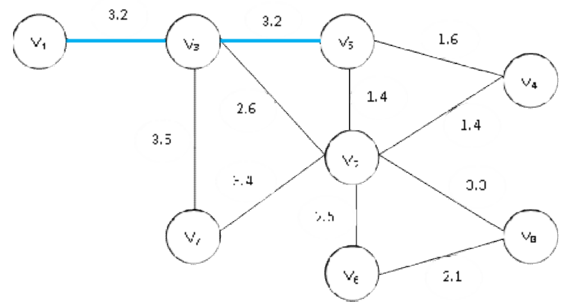
$$\boldsymbol{f_2(x_2, s) = cx_2s + f_1(x_2)}$$

$$\boldsymbol{x_2 = v_3 : \ f_2(v_3, v_2) = 2,6 + 3,2 = 5,8}$$

$$\boldsymbol{x_2 = v_3 : \ f_2(v_3, v_5) = 3,2 + 3,2 = 6,4}$$

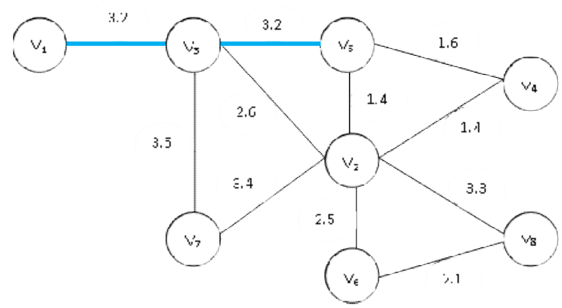
$$\boldsymbol{x_2 = v_3 : \ f_2(v_3, v_7) = 3,2 + 3,2 = 6,7}$$

Sehingga, didapat total bobot dengan menggunakan metode program dinamis adalah 6,4 dengan *path* $\boldsymbol{v_1}$-$\boldsymbol{v_3}$-$\boldsymbol{v_5}$.



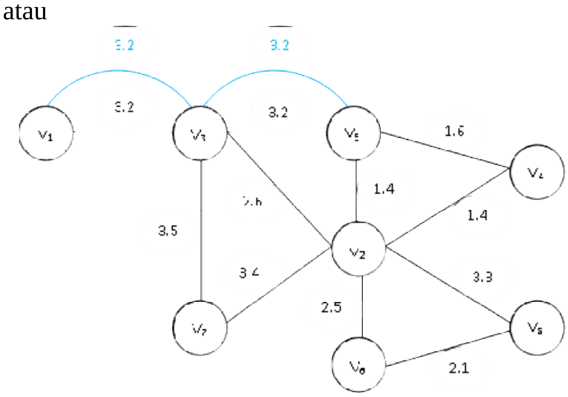
Gambar 9

Dari perbandingan kedua metode tersebut didapat *path* dengan bobot minimum adalah dengan menggunakan program dinamis dengan total jarak 34,6 km, yang dapat dilihat pada gambar berikut :



Gambar 10

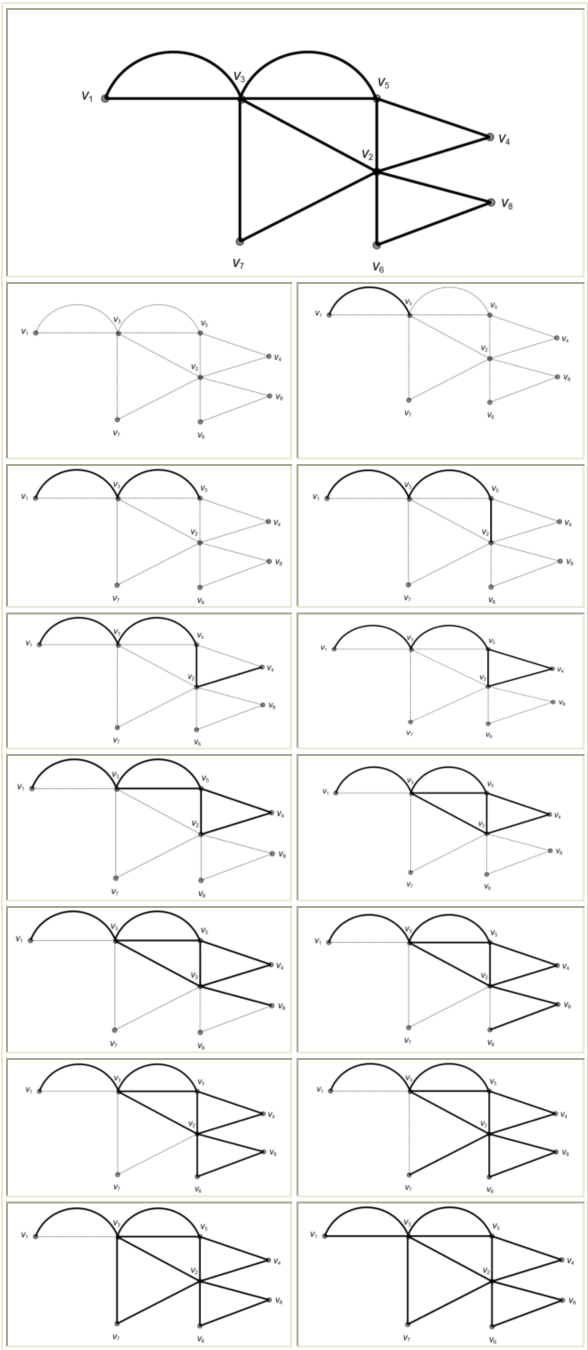
atau



Gambar 11

Berikut adalah konstruksi yang menyatakan jalan/rute perjalanan *Chinese Postman Problem* pada peta Kedaton Bandar Lampung

Gambar 12. Konstruksi garis *Euler* pada graf peta wilayah Kedaton.



Solusi sirkuit dari *Chinese Postman Problem* telah diperoleh dengan menyelesaikan konstruksi garis *Euler* pada graf tidak berarah dengan bobot 34,6 km dengan sirkuit

$v_1\ v_3\ \ v_5\, v_2\ \ v_4\ \ v_4\ \ v_5\ v_3\, v_2\ \ v_8\ \ v_6\ \ v_2\ \ v_7\ \ v_3\ \ v_1$

KESIMPULAN

Untuk kasus penentuan CPP pada kelurahan Kedaton didapat hasil bahwa Program Dinamis lebih baik dalam mengambil keputusan dan menghasilkan nilai optimal karena metode ini menguraikan solusi menjadi sekumpulan langkah atau tahapan sedemikian sehingga solusi dari persoalan dapat dipandang dari serangkaian keputusan yang saling berkaitan.

DAFTAR PUSTAKA

[1] Deo, N. (1989). *Graph Theory with Applications to Engineering and Computer Science*, Prentice-Hall of India Private Limited, New Delhi.

[2] Gross and Yellen. 2004. *Graph Theory*, CRC Press, LLC, New York.

[3] Hillier, F.S and Lieberman, G.J. (1994). *Pengantar Riset Operasi Jilid 1*. Diterjemahkan Oleh Tim Editor Penerbit Erlangga, Jakarta.

[4] Kurniasari, Yeni. 2006. *Penerapan Algoritma Greedy*. PT Gramedia Pustaka Utama, Jakarta.

[5] Munir, R. (2004). *Diktat Kuliah IF2251 : Program Dinamis*, Bandung.

[6] Munir, R. (2008). *Diktat Kuliah IF2091 : Struktur Diskrit 4th*, Bandung.

[7] Setiadi, Robert. (2008). *Algoritma Itu Mudah*. PT. Prima Infosarana Media, Jakarta