

## PROGRAM DINAMIS UNTUK PENENTUAN LINTASAN TERPENDEK DENGAN PENDEKATAN ALGORITMA FLOYD-WARSHALL

Enty Nur Hayati, Agus Setiawan  
Dosen Fakultas Teknik Universitas Stikubank Semarang

**DINAMIKA  
TEKNIK**  
Vol. VII, No. 1  
Januari 2013  
Hal 17 - 25

### Abstrak

Akhir-akhir ini pencarian rute perjalanan yang optimum kian menjadi masalah yang semakin penting dipicu oleh kenaikan harga bahan bakar yang hampir naik dua kali lipat, sehingga setiap orang berusaha menempuh perjalanan secepat mungkin untuk sampai ke lokasi tujuan dengan biaya yang minimum. Penelitian ini bertujuan untuk menentukan rute optimum dari Kecamatan Ngaliyan ke Kecamatan Sampangan dengan menggunakan Algoritma Floyd Warshall sehingga mempunyai jarak terpendek dengan asumsi tidak ada kendala seperti kemacetan dan keadaan geografis suatu jalur lalu lintas pada rute yang dilalui oleh orang yang akan melakukan perjalanan tersebut dan tidak memasukkan unsur biaya perjalanan. Pendekatan teknik dalam penyelesaian penelitian ini digunakan program dinamis dengan pendekatan Algoritma Floyd-Warshall, yang akan memilih satu jalur terpendek dan teraman dari beberapa alternatif jalur yang telah dihasilkan dari proses kalkulasi. Dari hasil pengolahan dan analisis data diperoleh bahwa rute optimal untuk perjalanan dari daerah Ngaliyan ke Kendeng menghasilkan jarak terpendek 12 km dengan rute Ngaliyan (A) → SPBU Ngaliyan (B) → Pasadena (D) → Kalipancur (G) → Simongan (I) → SPBU Jembatan Besi (K) → Menoreh (J) → Kendeng (L).

**Kata Kunci :** Program dinamis, Algoritma Floyd-Warshall, stage, state, fungsi rekursif

### 1. Latar Belakang

Luasnya sebuah kota dan banyaknya alternatif jalan raya seringkali menyulitkan seseorang untuk mencari rute optimum, baik dari segi jarak maupun biaya yang dikeluarkan untuk bepergian dari satu lokasi ke lokasi lain. Pada akhir-akhir ini pencarian rute perjalanan yang optimum kian menjadi masalah yang semakin penting dipicu oleh kenaikan harga bahan bakar yang hampir naik dua kali lipat, sehingga setiap orang berusaha menempuh perjalanan secepat mungkin untuk sampai ke lokasi tujuan dengan biaya yang minimum.

Untuk dapat memilih rute yang optimum, maka orang harus mengetahui jarak antar kota dan juga keadaan alam dari rute itu. Kemudian dipilihlah jalur terpendek dari lokasi awal ke lokasi tujuan. Namun, hal ini seringkali kurang membantu karena banyaknya jalan yang ada sehingga menyebabkan banyaknya pilihan jalur yang dapat ditempuh. Penelitian ini bertujuan untuk menentukan rute optimum dari Kecamatan Ngaliyan ke Kecamatan Sampangan dengan menggunakan Algoritma Floyd Warshall sehingga mempunyai jarak terpendek dengan asumsi

tidak ada kendala seperti kemacetan dan keadaan geografis suatu jalur lalu lintas pada rute yang dilalui oleh orang yang akan melakukan perjalanan tersebut dan tidak memasukkan unsur biaya perjalanan.

## 2. Tinjauan Pustaka

### 2.1 Program Dinamis

Program Dinamis (*Dynamic Programming*) adalah teknik manajemen *sain* yang diaplikasikan kepada persoalan dengan melibatkan keputusan berurutan yang saling berkaitan. Sebagai sebuah konsep, DP lebih luwes dibanding program-program optimasi lainnya. Aplikasi DP telah terbukti baik pada pengelolaan persediaan, jaringan, penjadwalan kerja untuk karyawan, pengendalian produksi, perencanaan penjualan dan bidang lain-lain. Formulasi model dilakukan dengan unik sesuai dengan persoalannya.

Karakteristik program dinamis

Program dinamis memiliki karakteristik berikut ini:

- a. Persoalan dapat dipisahkan menjadi beberapa tahap (*stages*), di mana setiap tahap membutuhkan keputusan kebijakan yang standard dan saling berhubungan.
- b. Setiap tahap memiliki sejumlah status (*state*). Secara umum, sekumpulan status ini merupakan berbagai kemungkinan kondisi yang timbul dari sistim persoalannya. Status ini memberikan informasi yang dibutuhkan setiap keputusan dan dampaknya pada tahap berikutnya. Jumlah status pada setiap tahap bisa definit atau infinit.
- c. Setiap keputusan kebijakan yang dibuat pada suatu tahap, status pada tahap tersebut ditransformasi ke dalam status yang berkaitan pada tahap berikutnya. Hubungan antar status pada tahap yang berurutan bisa bersifat deterministik atau probabilistik.

Pada sebuah persoalan dengan n-tahap, ada dua input, yaitu (1) *state* pada tahap ke-n ( $S_n$ ) dan *decision variable* ( $X_n$ ). Sedang outputnya adalah (1) *return* atau akibat dari setiap  $X_n$  yang dipilih,  $f_n(s, X_n)$ ; dan (2) status baru yang menjadi input pada tahap berikutnya ( $S_{n-1}$ ). Hubungan antara  $X_n$  dan  $f_n(s, X_n)$  ditentukan oleh *return function*. Sedang hubungan antar status pada tahap tertentu ditentukan oleh *transition function*.

- d. Solusi pada programa dinamis berprinsip kepada optimalitas yang dikembangkan oleh Bellman<sup>1</sup>: *An optimal policy must have the property that, regardless of the decision to enter a particular state, the remaining decisions must constitute an optimal policy for leaving that state.*

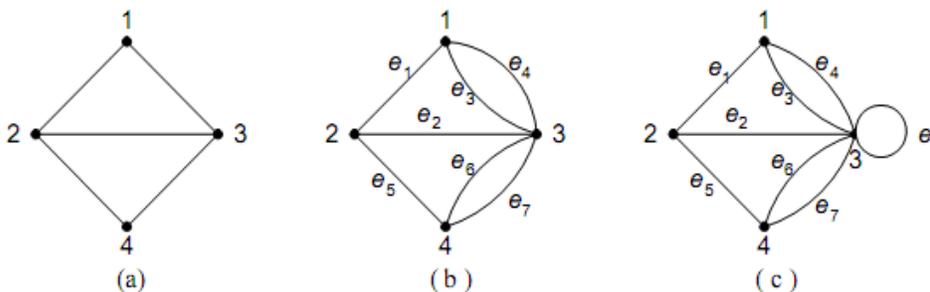
- e. Keputusan pada tahap berikutnya bersifat independen terhadap keputusan sebelumnya. Untuk menyelesaikan persoalan program dinamis, dimulai dari solusi awal pada suatu tahap, dan secara berurutan menuju tahap berikutnya dengan proses yang terbalik (*backward induction process*).
- f. Solusi optimal yang dihasilkan pada setiap tahap berprinsip kepada hubungan dalam bentuk fungsi rekursif (*recursion relationship*). Secara umum bentuk fungsi rekursif adalah:

$$f_n^*(S_n) = \max/\min \{f_n(S_n, X_n)\}$$

dengan  $f_n^*(S_n)$  = adalah hasil optimal dari keputusan pada tahap ke-n.

## 2.2 Graf

Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Representasi visual dari graf suatu objek dinyatakan sebagai noktah, bulatan atau titik, sedangkan hubungan antara objek dinyatakan dengan garis. Seringkali graf digunakan untuk merepresentasikan suatu jaringan. Secara geometri graf digambarkan sebagai sekumpulan noktah (simpul) di dalam bidang dwimitra yang dihubungkan dengan sekumpulan garis (sisi).



Gambar 1 Jenis-jenis graf (a) graf sederhana, (b) graf ganda, dan (c) graf semu

Jenis-jenis Graf dapat dikelompokkan sebagai berikut:

### a. Graf sederhana (*simple graph*)

Graf yang tidak mengandung gelang maupun sisi-ganda dinamakan graf sederhana. Graf pada Gambar 1 (a) adalah contoh graf sederhana yang merepresentasikan jaringan komputer. Simpul menyatakan komputer, sedangkan sisi menyatakan saluran telepon untuk berkomunikasi. Saluran telepon dapat beroperasi pada dua arah.

b. Graf tak-sederhana (*unsimple-graph*)

Graf yang mengandung sisi ganda atau gelang dinamakan graf tak-sederhana (*unsimple graph*). Ada dua macam graf tak-sederhana, yaitu graf ganda (*multigraph*) dan graf semu (*pseudograph*). Graf ganda adalah graf yang mengandung sisi ganda. Sisi ganda yang menghubungkan sepasang simpul bisa lebih dari dua buah. Graf pada Gambar 1 (b) adalah graf ganda. Sisi ganda dapat diasosiasikan sebagai pasangan tak terurut yang sama. Kita dapat juga mendefinisikan graf ganda  $G = (V, E)$  terdiri dari himpunan tidak kosong simpul-simpul dan  $E$  adalah himpunan ganda (*multiset*) yang mengandung sisi ganda.

c. Graf semu

Graf semu adalah graf yang mengandung gelang (*loop*). Graf pada Gambar 1 (c) adalah graf semu. Graf semu lebih umum daripada graf ganda, karena sisi pada graf semu dapat terhubung ke dirinya sendiri.

### 2.3 Lintasan Terpendek

Lintasan terpendek merupakan salah satu dari masalah yang dapat diselesaikan dengan graf. Jika diberikan sebuah graf berbobot, masalah lintasan terpendek adalah bagaimana cara mencari sebuah jalur pada graf yang dapat meminimalkan jumlah bobot sisi pembentuk jalur tersebut. Ada beberapa macam persoalan lintasan terpendek antara lain:

- a. Lintasan terpendek antara dua buah simpul tertentu (*a pair shortest path*)
- b. Lintasan terpendek antara semua pasangan simpul (*all pairs shortest path*)
- c. Lintasan terpendek dari simpul tertentu ke semua simpul yang lain.
- d. Lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu (*intermediate shortest path*)

Beberapa algoritma yang digunakan untuk menyelesaikan persoalan ini adalah algoritma Dijkstra, algoritma Bellman-Ford, dan algoritma Floyd-Warshall. Setiap algoritma penyelesaian persoalan lintasan terpendek memiliki kriteria masing-masing.

### 2.4 Algoritma Floyd-Warshall

Algoritma Floyd-Warshall adalah sebuah algoritma analisis graf untuk mencari bobot minimum dari graf berarah. Dalam pengertian lain Algoritma Floyd-Warshall adalah suatu metode yang melakukan pemecahan masalah dengan memandang solusi yang akan

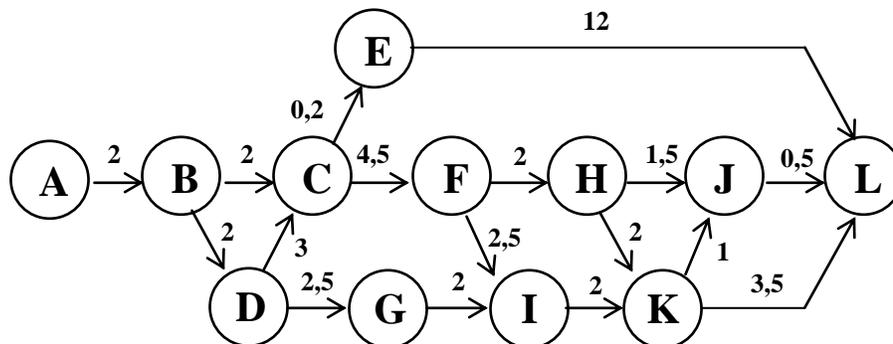
diperoleh sebagai suatu keputusan yang saling terkait. Artinya solusi-solusi tersebut dibentuk dari solusi yang berasal dari tahap sebelumnya dan ada kemungkinan solusi lebih dari satu (Novandi, R.A.D., 2007). Algoritma Floyd-Warshall ini akan memilih satu jalur terpendek dan teraman dari beberapa alternatif jalur yang telah dihasilkan dari proses kalkulasi (Sukrisno A.T dan Rachman A., 2007).

Algoritma Floyd-Warshall memiliki input graf berarah dan berbobot (V,E), yang berupa daftar titik (*node/vertex* V) dan daftar sisi (*edge* E). Jumlah bobot sisi-sisi pada sebuah jalur adalah bobot jalur tersebut. Sisi pada E diperbolehkan memiliki bobot negatif, akan tetapi tidak diperbolehkan bagi graf ini untuk memiliki siklus dengan bobot negatif. Algoritma ini menghitung bobot terkecil dari semua jalur yang menghubungkan sebuah pasangan titik, dan melakukannya sekaligus untuk semua pasangan titik. Prinsip yang dipegang oleh algoritma Floyd-Warshall adalah prinsip optimalitas, yaitu jika solusi total optimal, maka bagian solusi sampai suatu tahap (misalnya tahap ke-i) juga optimal (Novandi, R.A.D., 2007)

### 3. Analisis Data

#### 3.1 Pembuatan Graf

Ada beberapa jalan alternatif dari daerah Ngaliyan ke Kampus Kendeng Unisbank. Alternatif jalan tersebut disajikan dalam Gambar 3.



Gambar 2 Alternatif Jalan dari Daerah Ngaliyan Menuju Daerah Kendeng (Sumber: Hasil Pengolahan Data, 2013)

Keterangan Gambar 2:

*Node A = Ngaliyan, Node B = SPBU Ngaliyan, Node C = Krapyak, Node D = Pasadena, Node E = Tol Krapyak, Node F = APILL Simongan, Node G = Kalipancur, Node H = APILL Sampangan, Node I = Simongan, Node J = Menoreh, Node K = SPBU Jembatan Besi, dan Node L = Kendeng*

Data jarak antar daerah disajikan dalam Tabel 1.

Tabel 1 Jarak yang Ditempuh antar Daerah

Node	Jarak (km)	Node	Jarak (km)
A→B	2	F→I	2,5
B→C	2	G→I	2
B→D	2	I→K	2
C→E	0,2	H→K	2
C→F	4,5	H→J	1,5
D→C	3	K→J	1
D→G	2,5	K→L	3,5
E→L	12	J→L	0,5
F→H	2		

Sumber: Hasil Pengolahan Data, 2013

### 3.2 Pembagian *stage*

Dari Gambar 2 dapat dilihat bahwa perjalanan dari node A ke node L bisa ditempuh melalui beberapa rute. Setiap rute yang mungkin akan dinyatakan dengan *stage* yang mungkin. Hasil pembagian *stage* yang mungkin dilakukan disajikan dalam Tabel 2.

Tabel 2 Pembagian *Stage* untuk dari *Node A* ke *Node L*

<i>Stage (x)</i>	<i>Node/Stages</i>
1	A
2	B
3	D
4	C
5	F, G
6	H, I
7	K
8	E, J, K

Sumber: Hasil Pengolahan Data, 2013

### 3.4 Model Matematika

Pembuatan model matematika dari fungsi rekursif dapat dimulai dari tabel keputusan  $x_n$  ( $n = 1, 2, 3, 4, 5, 6, 7, 8$ ) yang merupakan variabel keputusan yang hendak dicapai pada saat berada di *stage* ke- $n$ . Berdasarkan notasi tersebut dapat dibuat rute yang ditempuh adalah  $1 \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4 \rightarrow x_5 \rightarrow x_6 \rightarrow x_7 \rightarrow x_8$ , dimana  $x_8$  adalah node L. Sebuah fungsi  $f_n(S, x_n)$  merupakan total jarak terbaik untuk menempuh *stage* berikutnya, bila saat ini berada di *stage*  $n$  dengan *state*  $S$  dan memilih  $x_n$  sebagai tujuan berikutnya. Variabel  $x_n^*$  adalah  $x_n$  yang memberikan nilai fungsi  $f_n(S, x_n)$  menjadi minimum, sehingga dapat dibuat

notasi  $f_n^*(S)$  sebagai nilai fungsi dengan harga  $f_n(S, x_n)$  yang paling minimum. Dari penjelasan tersebut maka dapat disusun persamaan sebagai berikut

$$\begin{aligned} f_n(S_n, x_n) &= \text{jarak di stage } n + \text{jarak minimum di stage } n+1 \text{ dengan state } S \text{ ke } x_n \text{ (dengan} \\ &\quad \text{state } x_n) \text{ ke stages berikutnya} \\ &= D_{Sx_n} + f_{n+1}^*(x_n) \end{aligned}$$

Nilai untuk  $D_{Sx_n} = C_{ij}$ , dengan  $i = S$  (state saat ini),  $j = x_n$  (state yang akan dituju). Solusi terbaik untuk menempuh A ke L adalah  $f_1^*(S)$ . Algoritma Floyd-Warshall mencari  $f_1^*(S)$  dengan cara melangkah mundur yakni mencari  $f_8^*(S)$ ,  $f_7^*(S)$ ,  $f_6^*(S)$ ,  $f_5^*(S)$ ,  $f_4^*(S)$ ,  $f_3^*(S)$ ,  $f_2^*(S)$  untuk setiap S yang mungkin dan menggunakan  $f_2^*(S)$  untuk memperoleh  $f_1^*(S)$ .

### 3.5 Hasil

Program dinamis yang digunakan adalah rekursif mundur, sehingga perhitungan dimulai dari tahap/stage ke-8. Pada stage 8, tujuan berikutnya ( $x_8$ ) ditentukan dengan keputusan sebelumnya ( $x_7$ ). Begitu pula pada tahap-tahap yang lainnya, sehingga dapat diperoleh solusinya. Semua proses perhitungan tersebut dapat dijelaskan sebagai berikut:

- a. Stage 8,  $x_8$  hanya mempunyai 1 pilihan yakni L, status yang mungkin di stage 8 adalah E, J, K, sehingga solusi di stage 8 adalah

Tabel 3 Perhitungan Stage ke-8

S	$f_8^*(S) = D_i x_8$		$f_8^*(S)$	$x_8^*$
	L			
E	12		12	L
J	0,5		0,5	L
K	3,5		3,5	L

- b. Stage 7,  $x_7$  mempunyai 2 pilihan yakni J dan L, status yang mungkin di stage 7 adalah E, sehingga solusi di stage 7 adalah

Tabel 4 Perhitungan Stage ke-7

S	$f_7^*(S) = D_i x_7$		$f_7^*(S)$	$x_7^*$
	J	L		
K	1,5	3,5	1,5	J

- c. Stage 6,  $x_6$  mempunyai 2 pilihan yakni J dan K, status yang mungkin di stage 6 adalah H dan I, sehingga solusi di stage 6 adalah

Tabel 5 Perhitungan *Stage* ke-6

S	$f_6^*(S) = D_i x_6$		$f_6^*(S)$	$x_6^*$
	J	L		
H	2	5,5	2	J
I	3,5	3,5	3,5	K

- d. *Stage* 5,  $x_5$  mempunyai 2 pilihan yakni H dan I, status yang mungkin di *stage* 5 adalah F dan G, sehingga solusi di *stage* 5 adalah

Tabel 6 Perhitungan *Stage* ke-5

S	$f_5^*(S) = D_i x_5$		$f_5^*(S)$	$x_5^*$
	J	L		
F	4	6	4	H
G	-	5,5	5,5	L

- e. *Stage* 4,  $x_4$  mempunyai 2 pilihan yakni E dan F, status yang mungkin di *stage* 4 adalah C, sehingga solusi di *stage* 4 adalah

Tabel 7 Perhitungan *Stage* ke-4

S	$f_4^*(S) = D_i x_4$		$f_4^*(S)$	$f_4^*$
	E	F		
C	8,5	12,2	8,5	F

- f. *Stage* 3,  $x_3$  mempunyai 2 pilihan yakni C dan G, status yang mungkin di *stage* 3 adalah D, sehingga solusi di *stage* 3 adalah

Tabel 8 Perhitungan *Stage* ke-3

S	$f_3^*(S) = D_i x_3$		$f_3^*(S)$	$x_3^*$
	C	G		
D	11,5	8	8	G

- g. *Stage* 2,  $x_2$  mempunyai 2 pilihan yakni J dan L, status yang mungkin di *stage* 2 adalah E, sehingga solusi di *stage* 2 adalah

Tabel 9 Perhitungan *Stage* ke-2

S	$f_2^*(S) = D_i x_2$		$f_2^*(S)$	$f_2^*$
	C	D		
B	10,5	10	10	D

- h. *Stage* 1,  $x_1$  mempunyai 1 pilihan yakni B, status yang mungkin di *stage* 1 adalah A, sehingga solusi di *stage* 1 adalah

Tabel 10 Perhitungan Stage ke-1

S	$f_1^*(S) = D_i x_1$	$f_1^*(S)$	$x_1^*$
	B		
A	12	12	B

Berdasarkan langkah terakhir, maka dapat disimpulkan bahwa terdapat 1 jalur yang paling pendek untuk berangkat dari daerah A ke daerah L. Jalur tersebut menghasilkan nilai 12, yakni  $A \rightarrow B \rightarrow D \rightarrow G \rightarrow I \rightarrow K \rightarrow J \rightarrow L$ .

#### 4. Simpulan

Dari hasil pengolahan data dapat ditarik simpulan bahwa melalui perhitungan metode Floyd Warshall didapatkan rute optimal dengan jarak terpendek adalah 12 km yaitu Ngaliyan (A)  $\rightarrow$  SPBU Ngaliyan (B)  $\rightarrow$  Pasadena (D)  $\rightarrow$  Kalipancur (G)  $\rightarrow$  Simongan (I)  $\rightarrow$  SPBU Jembatan Besi (K)  $\rightarrow$  Menoreh (J)  $\rightarrow$  Kendeng (L).

#### 5. Pustaka

- Murty, U.S.R. & J.A. Bondy, 1982. *Graph Theory with Applications*. North-Holland. New York
- Novandi, R.A.D., 2007,
- Sukrisno A.T dan Rachman A., 2007,
- Ristono A. dan Puryani, 2011. *Penelitian Operasional Lanjut*. Edisi Pertama, Graha Ilmu, Yogyakarta
- Wikipedia, 2007. Shortest Path Problem, [http://en.wikipedia.org/wiki/Shortest\\_path\\_problem.htm](http://en.wikipedia.org/wiki/Shortest_path_problem.htm), Tanggal Akses: 26 Desember 2006 pukul 10.32