

PENGEMBANGAN ALGORITMA PENCARIAN RUTE DAN PEMBEBANAN LALULINTAS *FUZZY*

Nindyo Cahyo Kresnanto
Mahasiswa Program S3
Program Studi Teknik Sipil
Fak. Teknik Sipil dan Lingkungan
Institut Teknologi Bandung
Jl. Ganesa No. 10 Bandung
Telp: (022)2502350
Fax: (022) 2502350
nindyo@students.itb.ac.id

Ofyar Z. Tamin
Staf Pengajar
Sekolah Pascasarjana
Program Studi Teknik Sipil
Fak. Teknik Sipil dan Lingkungan
Institut Teknologi Bandung
Jl. Ganesa No. 10 Bandung
Telp: (022) 2502350
Fax: (022) 2502350
ofyar@trans.si.itb.ac.id

Russ Bona F
Staf Pengajar
Sekolah Pascasarjana
Program Studi Teknik Sipil
Fak. Teknik Sipil dan Lingkungan
Institut Teknologi Bandung
Jl. Ganesa No. 10 Bandung
Telp: (022) 2502350
Fax: (022) 2502350
frazilla@trans.si.itb.ac.id

Abstract

The path finding, called the Tree Building, is the main foundation on traffic assignment model. In conventional method, each iteration in route finding process can produce best path or shortest-path. On the contrary, on fuzzy travel cost condition, that process could produce several paths as shortest-path nominations. This paper discussed the route finding using fuzzy travel cost approach. In conventional method, travel cost, as an input in the model, is expressed in deterministic form, while in fuzzy cost, travel cost is represented as range of certain value from under-bound to upper-bound. Path finding process in fuzzy cost was developed based on Dijkstra (1959) algorithm which placed each selected route in certain layers. Layer with selected route is sequenced from "first" best route. This algorithm is named as Dijkstra Multi Layer (DML) Algorithm. At the end, flow assignment has done with calculating fuzzy membership value in each selected route toward to the best route.

Keywords: fuzzy route cost, shortest path, layer, Dijkstra Algorithm

PENDAHULUAN

Pencarian rute merupakan fondasi utama dalam model pembebanan jaringan jalan, dan proses ini sering juga disebut dengan proses pembentukan pohon jaringan. Menurut Tamin (2000), secara umum pembentukan pohon adalah tahapan penting dalam setiap model pemilihan rute karena dua alasan utama. Pertama, hal ini sangat sering dilakukan dalam algoritma pemecahannya, minimal sekali per pengulangan. Kedua, algoritma pembentukan pohon yang baik dapat menghemat waktu dan biaya komputer. Algoritma yang baik bukan hanya efisien, tetapi harus ditulis dalam bentuk program komputer, sesuai dengan bahasa komputer yang digunakan. Van Vliet (1978) membahas dengan sangat baik algoritma yang paling sering digunakan.

Secara umum terdapat 2 (dua) algoritma dasar yang sering digunakan untuk mencari rute tercepat (termurah) dalam suatu jaringan jalan. Kedua algoritma itu adalah algoritma Moore (1957) dan algoritma Dijkstra (1959). Keduanya menggunakan notasi berorientasi simpul jarak (biaya) ruas antara dua titik A dan titik B dalam suatu jaringan, yang dinotasikan dengan $d_{A,B}$. Rute didefinisikan dalam bentuk urutan simpul yang saling berhubungan, misalnya A-C-D-H dan seterusnya, sedangkan jarak rute adalah penjumlahan setiap ruas yang ada dalam rute tersebut.

Dalam kasus *fuzzy*, rute tercepat tidak dapat secara langsung ditetapkan, karena jika biaya ruas didefinisikan menggunakan biaya-ruas-*fuzzy* maka ada kemungkinan rute tercepat yang dihasilkan akan lebih dari satu rute. Blue et al (1997) telah mengembangkan

algoritma dasar untuk menentukan rute optimum dalam kasus *fuzzy*. Asumsi dasar yang digunakan adalah: (1) tidak ada dominasi rute tercepat, (2) biaya ruas dinyatakan dalam *Fuzzy-Number*, dan (3) rute tercepat diurutkan berdasarkan rangking.

PENGEMBANGAN ALGORITMA PENCARIAN RUTE

Dalam pencarian rute konvensional, setiap pengulangan akan hanya akan menghasilkan satu rute terbaik atau *shortestpath*. Berbeda dengan pemilihan rute konvensional, dalam kondisi biaya *fuzzy* proses pemilihan rute diharapkan akan menghasilkan beberapa rute yang dapat dijadikan rute nominasi sebagai *shortestpath*. Dengan demikian dalam setiap pengulangan, algoritma pencarian menghasilkan lebih dari satu rute terbaik, mulai dari terbaik pertama, kedua, hingga ke-*n*. Dalam penelitian ini, algoritma pencarian rute yang akan dipakai dalam kondisi *fuzzy* dikembangkan dari algoritma Dijkstra (1959). Pengembangan dilakukan dengan membagi algoritma ke dalam layer-layer yang mewakili urutan rute terbaik (rute terbaik pertama disimpan dalam *layer* satu, rute terbaik kedua disimpan dalam *layer* dua, dan seterusnya). Selanjutnya algoritma baru ini akan disebut sebagai Algoritma Dijkstra Multi Layer (DML).

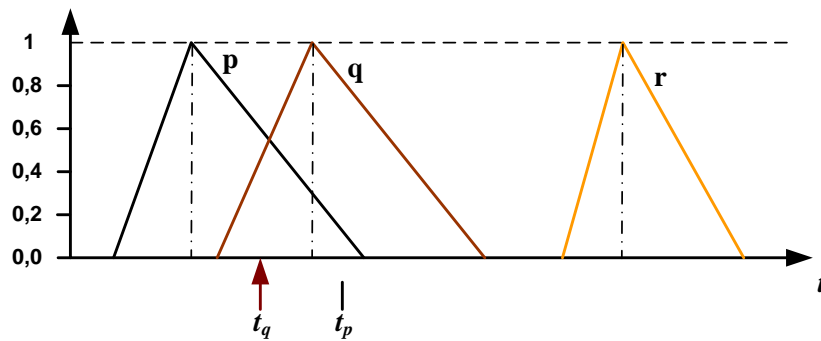
Tabel 1 Perbedaan Algoritma Dijkstra dan DML

Algoritma Dijkstra	DML
Inisialisasi	Inisialisasi (di setiap level mayer)
$d(a)=\infty$; $d(s) = 0$; pred()	$d(a)=\infty$; $d(s) = 0$; pred() ; level()
Set $Q()$ yang akan berisi data simpul yang belum dicapai	Set $Q()$ yang akan berisi data simpul yang belum dicapai
Set $S()$ yang akan berisi data simpul yang sudah dicapai	Set $S()$ yang akan berisi data simpul yang sudah dicapai
Masukkan s ke Q	Masukkan s ke Q
Prosedur	Prosedur (lakukan di setiap level mayer)
Cari nilai minimum di Q tetapkan sbg u	Cari nilai minimum di Q tetapkan sbg u
Cari setiap simpul v yang terhubung dgn u , jika $d(v) > d(u) + [u,v]$ maka $d(v) = d(u) + [u,v]$, masukkan v ke Q	Cari setiap simpul v yang terhubung dgn u , jika $d(v) > d(u) + [u,v]$ maka $d(v) = d(u) + [u,v]$, masukkan v ke Q level n . jika $d(v) < d(u) + [u,v]$ maka $d(v) = d(u) + [u,v]$ masukkan v ke Q level $(n+1)$
Jika Q belum kosong ulangi prosedur sampai Q kosong	Jika Q belum kosong ulangi prosedur sampai Q kosong

PENGEMBANGAN ALGORITMA PEMBEBANAN

Dalam suatu jaringan deterministik, rute optimum dari suatu titik asal ke suatu titik tujuan akan berupa rute tunggal dengan biaya minimum. Dalam situasi *fuzzy*, dengan menggunakan biaya-*fuzzy* sebagai biaya rutennya, tidak dapat ditentukan suatu rute tunggal yang dapat dinyatakan sebagai rute optimum. Sebagai ilustrasi dapat dilihat pada Gambar 1, dengan p, q, r adalah tiga buah rute dengan masing-masing biaya rute τ_p, τ_q, τ_r , dalam biaya-rute-*fuzzy*. Secara intuisi terlihat bahwa rute p dan q lebih "cepat" pada rute r . Tetapi untuk rute p dan q tidak dapat dikatakan secara mutlak bahwa rute p lebih "cepat" daripada

rute r . Alasannya adalah karena $\exists t_p \in \text{Supp}(\tau_p)$ dan $\exists t_q \in \text{Supp}(\tau_q)$ sedemikian sehingga $t_p > t_q$ (Ban et al, 2004).



Gambar 1 Ilustrasi Rute Optimum *Fuzzy* (*fuzzy shortest path*) (Ban et al, 2004)

Untuk menghitung *fuzzy-shortest-path* (FSP) diperlukan beberapa teori dasar yang mendukung hal tersebut. Teori-teori ini akan diuraikan berikut.

Suatu *graph* G terdiri atas satu himpunan *vertices* (*node/vertex/simpul*) V dan satu himpunan *edges* (*Arcs/ruas*) E :

$$G = (V, E) \tag{1}$$

dengan:

$$V = \{v_1, v_2, \dots, v_{ny}\}$$

$$E = \{e_1, e_2, \dots, e_{ng}\}$$

Dalam suatu *graph* terbobot, tiap ruas akan mempunyai suatu bobot (seringkali berupa *impedance*, jarak, waktu tempuh, kapasitas),

$$w_i = W(e_i) \tag{2}$$

Sesuai dengan fungsi W , nilai bobot dapat berupa nilai *crisp* atau nilai-*fuzzy*. Suatu rute (*path*) P merupakan rangkaian dari beberapa ruas:

$$P = (e_{i1}, e_{i2}, \dots, e_{in}) \tag{3}$$

Jika *graph* terbobot dengan jarak, maka rute mempunyai total jarak hasil penjumlahan bobot tiap ruas dalam rute,

$$l_p = \text{length}(P) = \sum_{e_k \in P} w_k \tag{4}$$

Fuzzy Graph dengan Bobot Nilai-Fuzzy

Blue (1997) mengelompokkan tipe *fuzzy graph* dalam 5 (lima) kelompok. Untuk penentuan rute optimum tipe *graph* yang biasa dipakai adalah *graph* dengan simpul *crisp* dan ruas *fuzzy*. Ruas *fuzzy* didefinisikan sebagai berikut:

$$w_i = w_{i,1} \setminus \mu_{i,1} + w_{i,2} \setminus \mu_{i,2} + \dots \quad (5)$$

Jika Π merupakan himpunan semua rute dari simpul v_a menuju v_b , maka panjang-*fuzzy* rute:

$$l_p = \text{length}(P) = \sum_{e_k \in P} w_k, \text{ dengan } P \in \Pi \quad (6)$$

Himpunan *fuzzy* rute-rute optimum adalah himpunan *fuzzy* S dalam Π dengan keanggotaan π_S sebagai berikut:

$$\pi_S(P) = \min_{Q \in \Pi} \{ \hat{\mu}_{l_p \leq l_Q} \}, \text{ dengan } P \in \Pi \quad (7)$$

Support terdiri atas semua rute yang potensial mempunyai panjang minimum. Perhitungan ini didasarkan pada suatu rute yang secara mutlak lebih “pendek” daripada rute yang lain.

$$\text{supp}(S) = \{ P \in \Pi \mid \hat{\mu}_{l_p \leq l_Q} > 0, \forall Q \in \Pi \} \quad (8)$$

Himpunan *fuzzy* rute-rute optimum yang didefinisikan pada Persamaan 7 dapat didefinisikan sebagai *fuzzy-shortest-path*, dengan ruas/edge e_i mempunyai keanggotaan dalam himpunan fuzzy S' :

$$\mu_{S'}(i) = \max_{e_i \in P, P \in \Pi} \{ \pi_S(P) \} \text{ untuk } i = 1, \dots, n_E \quad (9)$$

Persamaan 7 dan Persamaan 9 dapat dituliskan kembali sebagai:

$$\mu_{S'}(i) = \max_{e_i \in P, P \in \Pi} \left\{ \min_{Q \in \Pi} \{ \hat{\mu}_{l_p \leq l_Q} \} \right\} \text{ untuk } i = 1, \dots, n_E. \quad (10)$$

Pertimbangan utama untuk memecahkan masalah algoritma *fuzzy-shortest-path* adalah estimasi sebagai berikut:

$$\{ P \in \Pi \mid \inf \{ \text{supp}(l_p) \} < \kappa \} \subseteq \text{supp}(S) \subseteq \{ P \in \Pi \mid \inf \{ \text{supp}(l_p) \} \leq \kappa \} \quad (11)$$

dengan:

$$\kappa = \min_{P \in \Pi} \{ \sup \{ \text{supp}(l_p) \} \} \quad (12)$$

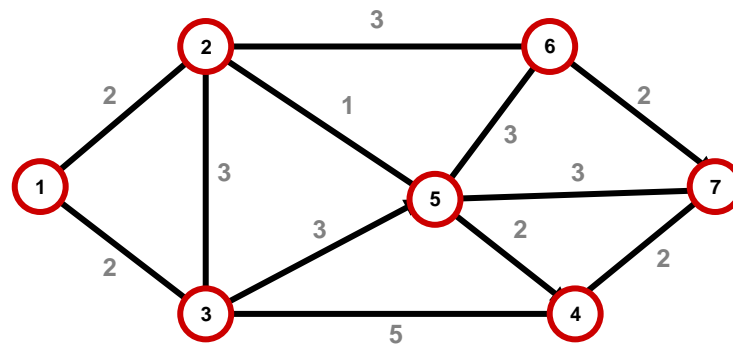
UJI COBA DALAM JARINGAN ARTIFISIAL

Peralatan yang dipakai dalam percobaan ini adalah peralatan berupa perangkat lunak (*software*), bahasa pemrograman, dan perangkat keras (*hardware*). Rincian peralatan yang digunakan dapat dilihat pada Tabel 2.

Tabel 2 Peralatan yang Digunakan

Jenis Peralatan	Uraian	Keterangan
Bahasa Pemrograman	C++	
Software	Microsoft Visual C++ 2008 Express Edition	Software dengan <i>free license</i> yang dapat di download langsung dari website <i>Microsoft</i> .
Hardware	Komputer dengan spesifikasi: <ul style="list-style-type: none"> • Operating System: Windows XP Professional (5.1, Build 2600) Service Pack 2 • Processor: Intel(R) Core(TM)2 CPU T5500 @ 1.66GHz • Memory: 1014 MB 	

Jaringan buatan terdiri atas 7 (tujuh) simpul dan 12 (dua belas) *link*. Masing-masing *link* diberikan bobot yang menggambarkan biaya perjalanan awal hingga ujung simpul. *Link* dapat dilalui dalam dua arah (Gambar 2).

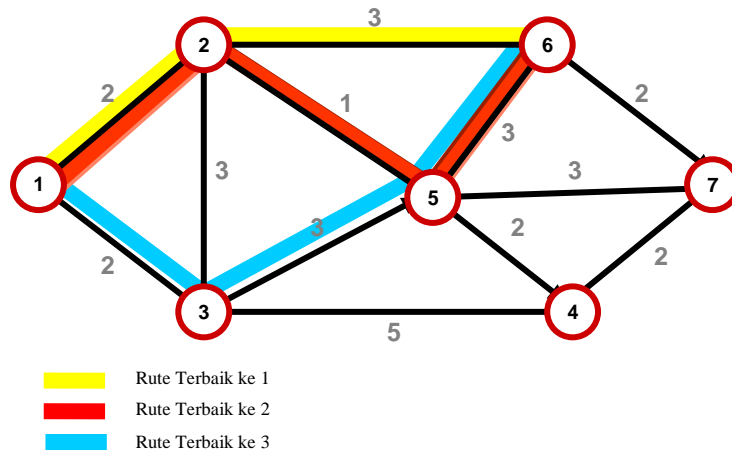


Gambar 2 Jaringan Artifisial

Percobaan dilakukan dalam urutan proses sebagai berikut:

1. Mencari 3 rute dari simpul 1 ke simpul 6 yang dapat digolongkan sebagai kelompok rute terbaik, yang diurutkan berdasarkan terbaik kesatu, kedua, dan ketiga. Hasil proses pertama ini adalah rute terbaik 1 dengan biaya perjalanan 5 (mulai dari simpul 1, 2, 6); rute terbaik 2 dengan biaya perjalanan 6 (mulai dari simpul 1, 2, 5, 6); rute terbaik 3 dengan biaya perjalanan 8 (mulai dari simpul 1, 3, 5, 6). (Gambar 3 dan Gambar 4).
2. Dengan anggapan terdapat 2000 pergerakan dari 1 ke 6, maka akan dilakukan pembebanan ke rute-rute terpilih dengan pendekatan *fuzzy*. Asumsi yang digunakan adalah:
 - a. Nilai bawah biaya perjalanan rute = Biaya Perjalanan Rute - (Biaya Perjalanan Rute x 0.1).
 - b. Nilai atas biaya perjalanan rute = Biaya Perjalanan Rute + (Biaya Perjalanan Rute x 0.1).

Pembebanan menghasilkan arus pada tiap rute seperti pada Gambar 5.



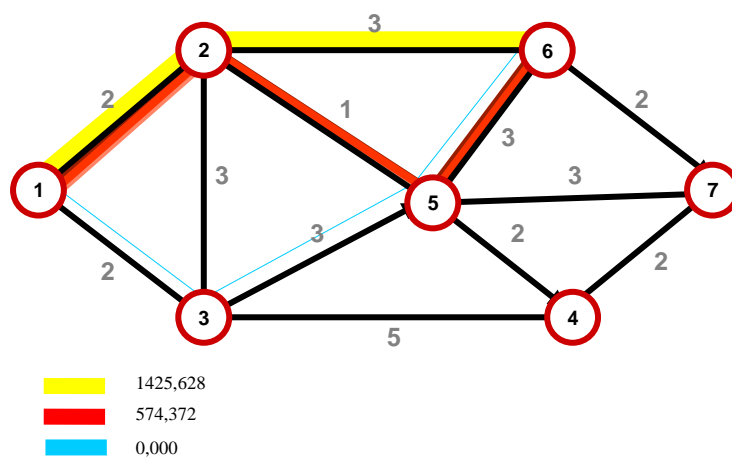
Gambar 3 Hasil Pencarian Rute Terbaik

```

C:\WINDOWS\system32\cmd.exe
Matrik Biaya Perjalanan Aktual
0.000 2.000 2.000 0.000 0.000 0.000 0.000
2.000 0.000 3.000 0.000 1.000 3.000 0.000
2.000 3.000 0.000 5.000 3.000 0.000 0.000
0.000 0.000 5.000 0.000 2.000 0.000 2.000
0.000 1.000 3.000 2.000 0.000 3.000 4.000
0.000 3.000 0.000 0.000 3.000 0.000 2.000
0.000 0.000 0.000 2.000 4.000 2.000 0.000

enter the source vertex
1
enter the sink vertex
6
-> panjang rute terbaik ke 1: <5.000>
-> panjang rute terbaik ke 2: <6.000>
-> panjang rute terbaik ke 3: <8.000>
  
```

Gambar 4 Hasil Keluaran Program



Gambar 5 Hasil Pembebanan Jaringan

KESIMPULAN

Dari hasil pengembangan model dan uji coba yang telah dilakukan dapat diambil beberapa kesimpulan sebagai berikut:

1. Model telah dicoba dengan menggunakan data sederhana, dan dapat dicapai solusi untuk kebutuhan pencarian rute dalam kondisi biaya perjalanan *fuzzy*.
2. Sebelum model yang telah dihasilkan akan dipakai, diperlukan uji selanjutnya, yaitu melakukan uji coba dalam suatu jaringan medium.

DAFTAR PUSTAKA

- Akiyama, T and Tomoko, N. 1998. *The Proposal Of Fuzzy Traffic Assignment Models*. Proceedings of the Eastern Asia Society for Transportation Studies, 3 (6): 263-277.
- Akiyama, T, et al. 1999. *Description of Route Choice Behaviour by Fuzzy Neural Network*. Research Report of The Faculty of Engineering Gifu University, 49: 27-38.
- Ban, X, et al. 2004. *Traffic Assignment Model With Fuzzy Travel Time Perceptions*. 83rd Annual Meeting of the Transportation Research Board. Washington, DC.
- Benetti, M and Marco, D. M. 2002. *Traffic Assignment Model With Fuzzy Travel Cost*. 13th Mini - EURO Conference and 9th Meeting of the Euro Working Group on Transportation June 10-13. Bari, Italy.
- Blue, M, et al. 1997. *Applications of Fuzzy Logic to Graph Theory*. Los Alamos National Laboratory.
- Inokuchi, H. and Kawakami, S. 2002. *Development of the Fuzzy Traffic Assignment Model*. <http://www.trans.civil.kansai-u.ac.jp/inokuchi/study/SCIS2002/153.pdf>.
- Jantzen, J. 2006. *Tutorial On Fuzzy Logic*. Technical University of Denmark. Oersted-DTU, Automation, Bldg 326, 2800. Kongens Lyngby, Denmark.
- Kikuchi, S and Parta, C. 2005. *Place of Possibility Theory in Transportation Analysis*, Transportation Research Part B 40 (2006). Washington, DC.
- Kresnanto, N. C dan Tamin, O. Z. 2006. *Kajian Model Pembebanan Jaringan Dengan Fuzzy Sistem*. Jurnal FSTPT IX. Universitas Brawijaya. Malang.
- Kresnanto, N. C., Tamin, O. Z. 2007. *Biaya Perjalanan Fuzzy Untuk Pembebanan Lalu Lintas*. Jurnal FSTPT X. Universitas Tarumanagara. Jakarta.
- Kusdian, R. D. 2006. *Model Stokastik Untuk Pembebanan Lalulintas Banyak Rute Dengan Mempertimbangkan Persepsi Biaya Perjalanan*. Disertasi FTSL. Institut Teknologi Bandung, Bandung.
- Kusumadewi, S. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Penerbit Graha Ilmu.
- Liu, H. X, et al. 2003. *A Formulation and Solution Algorithm for Fuzzy Dynamic Traffic Assignment Model*. <http://ITSReviewonline/spring2003/trb2003/liu-algorithm.pdf>
- Rusell, S and P. Novig. 2003. *Artificial Intelligence: A Modern Approach - Second Edition*. New Jersey: Prentice Hall – Pearson Education, Inc.
- Suyanto. 2002. *Intelejensia Buatan*. Jurusan Teknik Informatika, Sekolah Tinggi Teknologi Telkom. Bandung.

- Suyoto. 2004. *Intelegensi Buatan: Teori dan Pemrograman*. Yogyakarta: Penerbit Gaya Media.
- Tamin, O. Z. 2000. *Perencanaan dan Pemodelan Transportasi – Edisi Kedua*. Bandung: Penerbit Institut Teknologi Bandung.