



PENCARIAN RUTE TERBAIK PEMADAM KEBAKARAN KOTA SEMARANG MENGGUNAKAN ALGORITMA DIJKSTRA DENGAN LOGIKA FUZZY SEBAGAI PENENTU BOBOT PADA GRAF

Nanang Nggufron[✉], Rochmad, Mashuri

Jurusan Matematika, FMIPA, Universitas Negeri Semarang, Indonesia
Gedung D7 Lt. 1, Kampus Sekaran Gunungpati, Semarang 50229

Info Artikel

Sejarah Artikel:
Diterima November 2017
Disetujui Desember 2017
Dipublikasikan Mei 2019

Keywords:
*Rute Terbaik; Dijkstra;
Logika Fuzzy; PHP; Pemadam
Kebakaran; Daerah Rawan
Kebakaran.*

Abstrak

Penelitian ini mengkaji sebuah permasalahan optimasi untuk masalah pencarian rute. Tujuan dari penelitian ini adalah menentukan rute terbaik mobil pemadam kebakaran menuju daerah rawan kebakaran di kota Semarang menggunakan algoritma Dijkstra dan diaplikasikan kedalam bahasa pemrograman PHP. Berdasarkan data primer dan data sekunder berupa data pos pemadam kebakaran, daerah rawan kebakaran, dan peta jalan kota semarang dari Dinas Kebakaran dan Dinas Perhubungan kota Semarang dapat disusun gambar jaringan jalan dan membentuk sebuah graf. Selanjutnya dari gambar graf dapat diperoleh rute terbaik menggunakan algoritma Dijkstra. Terdapat 2 Parameter yaitu panjang jalan dan kepadatan jalan untuk menghasilkan bobot berupa tingkat kemacetan yang dihasilkan menggunakan logika fuzzy. Berdasarkan hasil analisis dengan cara perhitungan manual maupun dengan program, diperoleh 7 rute terbaik untuk masing-masing pos pemadam kebakaran yang direkomendasikan, yang diperoleh dari parameter tingkat kemacetan jalan yang menggabungkan 2 parameter yaitu panjang jalan dan kepadatan jalan.

Abstract

This study examined an optimization problem for route search problems. The purposes of this research were to determine the best route of fire truck to the fire prone area in Semarang city using Dijkstra algorithm and to apply the algorithm with PHP programming language. Based on the primary data and secondary data in the form of fire postal data, fire prone areas, and city road map semarang from the Fire Service and Transportation Office of Semarang city can be drawn pictures of road network and form a graph. Furthermore from the graph images can be obtained the best route using Dijkstra algorithm. There are 2 parameters namely the length of the road and the density of the road to generate the weight of the resulting level of congestion using fuzzy logic. Based on the results of the analysis by manual calculation and with the program, obtained the best 7 routes recommended for each fire station, which were obtained from the traffic jam levels, combines 2 parameters namely the length of the road and the road density.

How to Cite

Nggufron N., Rochmad, & Mashuri. (2019). Pencarian Rute Terbaik Pemadam Kebakaran Kota Semarang Menggunakan Algoritma Dijkstra dengan Logika Fuzzy Sebagai Penentu Bobot Pada Graf. *UNNES Journal of Mathematics* 8(1): 40-49.

PENDAHULUAN

Kebakaran adalah suatu nyala api, baik kecil atau besar pada tempat, situasi dan waktu yang tidak dihendaki, merugikan dan pada umumnya sukar dikendalikan. Sedangkan menurut Depnaker (Triato, 2013) Kebakaran adalah suatu reaksi oksidasi eksotermis yang berlangsung dengan cepat dari suatu bahan bakar yang disertai dengan timbulnya api atau penyalaman. Sumber bahaya kebakaran biasanya berasal dari kelalaian manusia dalam melakukan kegiatan seperti merokok, memasak, penggunaan alat elektronik, bermain sumber api, kebocoran gas, dan sebagainya. Selain oleh faktor manusia, kejadian kebakaran juga dapat disebabkan oleh alam seperti petir, gempa bumi, letusan gunung api, kekeringan dan sebagainya (Pemerintah Republik Indonesia, 2007). Bencana kebakaran sering terjadi di daerah yang memiliki pemicu bencana kebakaran yang tinggi seperti, hutan, dan perkotaan.

Kota Semarang merupakan kota yang rentan terhadap bencana kebakaran. Pada tahun 2016 rentang antara bulan Januari sampai pertengahan Juni total kasus kebakaran mencapai 54 kasus, yang menyebabkan kerugian materil lebih dari 3 miliar. Sebagian besar lokasi kebakaran berada di wilayah yang memiliki resiko kebakaran yang tinggi seperti bangunan umum, bangunan campuran, industri, kendaraan, lahan kosong, pemukiman, dan lain-lain (Dinas Kebakaran Kota Semarang). Angka tersebut menunjukkan tingginya kejadian kebakaran dan besarnya kerugian yang diakibatkan oleh kebakaran di Kota Semarang.

Mengingat potensi kebakaran semakin lama semakin signifikan, bahaya bencana ini harus segera diantisipasi dan dihadapi dengan berbagai upaya penanggulangan yang komprehensif, sistematis, efektif dan berkelanjutan. Salah satu upaya penanggulangan pasca kebakaran adalah dengan memperpendek waktu perjalanan mobil pemadam, sehingga petugas pemadam kebakaran dapat datang lebih cepat menuju lokasi kebakaran. Untuk memperpendek waktu perjalanan, maka haruslah dicari rute terbaiknya.

Pencarian rute terbaik atau dalam teori graf sering disebut pencarian rute terpendek adalah usaha untuk mencari rute yang paling dekat dari titik awal hingga titik akhir tujuan dengan beban paling ringan dibandingkan dengan seluruh rute yang ada (Taufiq, 2015). Pencarian rute terbaik ini telah diterapkan diberbagai bidang untuk mengoptimasi kinerja suatu sistem, baik untuk meminimalkan biaya atau mempercepat jalannya suatu proses (Purwanto *et al.*, 2005).

Hasil pencarian rute terbaik, dapat dimanfaatkan untuk mencapai lokasi yang dituju pada umumnya. Pada kondisi khusus, antara lain seperti mencari lokasi tempat wisata, membantu ambulan untuk menuju rumah sakit, menghindari kemacetan dan pusat keramaian, meminimalisir penggunaan bahan bakar, memprediksi aliran lalulintas yang akan dilewati, dan untuk menentukan posisi router yang tepat untuk meningkatkan kecepatan internet.

Pada kasus pencarian rute, seringkali hanya panjang dari tiap ruas jalan yang dipergunakan sebagai parameter. Dalam kenyataannya banyak faktor lalu lintas yang semestinya digunakan sebagai pertimbangan dalam menentukan rute yang dipilih (Faro & Giordano, 2016). Salah satu faktor yang mempengaruhi adalah kepadatan jalan. Jumlah kendaraan saat ini bertambah dengan pesat tetapi tidak diiringi dengan perbaikan kondisi jalan. Sehingga terjadi penumpukan kendaraan pada ruas jalan tertentu dan terjadi kemacetan ataupun perjalanan tidak lancar. Sehingga pada implementasi analisa dan simulasi, dipergunakan pembangkitan bilangan acak atau masukan manual untuk diproses lebih lanjut.

Logika *fuzzy* digunakan untuk memodelkan kuantitas dari input. Logika ini digunakan untuk situasi model dimana pembuatan keputusan dalam lingkup yang kompleks dan sulit untuk melakukan pengembangan model matematis. Semua nilai keluaran dari logika *fuzzy*, bisa dipakai sebagai input algoritma yang lain. Logika *fuzzy* sangat dekat dengan pemikiran manusia, secara luas diterima dan diaplikasikan pada berbagai permasalahan nyata seperti optimasi produksi yang memanfaatkan aplikasi *fuzzy linear programming* (Yulianto, 2012). Aplikasi logika *fuzzy* juga pernah digunakan dalam masalah *travelling salesman problem* dengan bantuan Matlab (Fitriana, 2015).

Oleh karena itu dibutuhkan sebuah analisis serta perhitungan matematis yang sangat detail dalam proses menentukan rute jalan yang akan dipilih. Peneliti dengan pertimbangan karakteristik jalan serta fenomena yang terjadi, maka dipergunakan logika *fuzzy* untuk memberi pertimbangan yang spesifik dalam memberikan nilai bobot tiap ruas jalan dan algoritma dijkstra untuk mencari rute yang diambil, sehingga didapat rute jalan tercepat.

Algoritma Dijkstra merupakan salah satu metode untuk mencari rute terbaik dari sebuah simpul ke semua simpul lainnya dalam graf yang hanya memiliki bobot positif. Dalam mencari rute terbaik dari suatu simpul ke semua pasangan

simpul, algoritma Dijkstra melalui sejumlah langkah yang menggunakan prinsip *greedy*.

Prinsip *greedy* pada algoritma Dijkstra menyatakan bahwa pada setiap langkah kita memilih sisi yang berbobot minimum dan memasukkannya dalam himpunan solusi (Munir, 2005). Hasil penelitian Peyer *et al.* (2009) menunjukkan bahwa algoritma Dijkstra dapat digeneralisasikan untuk menemukan rute terpendek pada graf berarah dengan panjang sisi non-negatif, sehingga sangat cocok dengan karakteristik perkotaan yang mempunyai jalan satu arah. Algoritma Dijkstra pernah digunakan pada pencarian jalur evakuasi dengan memperhatikan waktu dan beberapa hamabatan jalan, seperti simpangan keramaian, dan lain-lain (Chen *et al.*, 2014).

Terdapat beberapa algoritma dalam pencarian rute terbaik selain algoritma Dijkstra, diantaranya algoritma Bellman Ford dan Floyd-Warshall. Algoritma Dijkstra memiliki waktu eksekusi yang lebih cepat dibandingkan dengan algoritma Bellman Ford dalam menentukan masalah rute terbaik yaitu $O(|V|^2)$ untuk algoritma Dijkstra dan $O(|V| \cdot |E|)$ untuk algoritma Bellman Ford (Patel & Baggar, 2014). Handaka (2011), dalam penelitiannya tentang pencarian Rute Terpendek pada Link-State Routing Protocol dengan membandingkan Algoritma Dijkstra, Bellman-Ford, dan Floyd-Warshall, menyimpulkan bahwa algoritma Dijkstra dapat menjadi pilihan favorit dibandingkan dengan algoritma Bellman-Ford dan Floyd-Warshall, mengingat kebutuhan waktu dan ruangnya sangatlah kecil.

Penggabungan metode *fuzzy* dan algoritma Dijkstra untuk pencarian rute terpendek pernah dilakukan oleh (Ichsan, 2012). Dalam penelitian tersebut, logika *fuzzy* digunakan untuk mendapatkan bobot setiap ruas jalan dengan variabel kepadatan lalu-lintas dan panjang jalan.

Dalam mendapatkan output dari kedua variabel itu, digunakan *fuzzy* model Sugeno. Sementara algoritma Dijkstra digunakan untuk pencarian rute terpendek setelah didapatkan bobot untuk masing-masing ruas jalan. Penelitian tersebut terbatas pada perhitungan secara manual belum sampai tahap pembuatan program.

Persoalan mencari lintasan terpendek didalam graf merupakan salah satu persoalan optimasi (Munir, 2010). Dalam mencari rute terpendek, semakin banyak titik dan garis pada graf akan semakin rumit dan membutuhkan wakru lama ketika melakukan pencarian secara manual. Oleh karena itu diperlukan program pendukung dalam melakukan pencarian lintasan

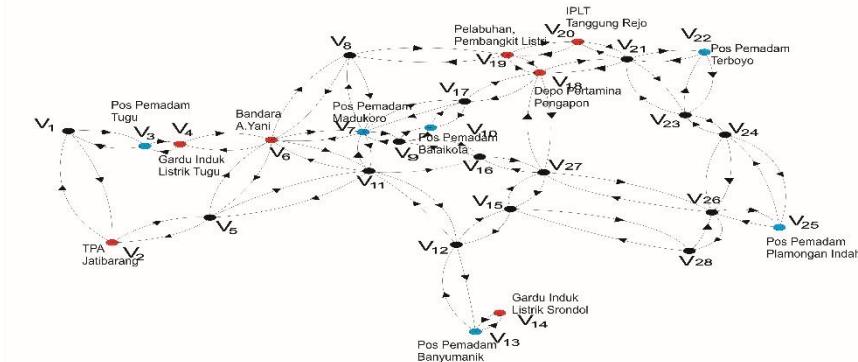
terpendek untuk mempercepat pencarian. Salah satu penerapan algoritma Dijkstra dengan aplikasi adalah optimasi masalah lintasan terpendek dengan menggunakan software TORA pada distribusi air PDAM di Kabupaten Demak (Prasetyo, 2013). Pemrograman Visual basic juga pernah digunakan untuk dalam simulasi algoritma Dijkstra dalam menangani masalah lintasan terpendek (Mardlootillah, 2015). Dalam penelitian ini pencarian rute terbaik akan dibangun menggunakan pemrograman PHP. Pemrograman PHP dipilih karena dapat dijalankan pada semua Sistem Operasi karena PHP berjalan secara web base yang artinya semua sistem operasi yang mempunyai browser dapat menggunakan program PHP tak terkecuali HP. Program PHP juga tidak memerlukan kompilasi/compile dalam penggunaanya.

METODE

Metode yang digunakan pada penelitian ini adalah studi pustaka, pengumpulan data, cara pemecahan masalah, dan penarikan kesimpulan. Studi pustaka menjelaskan tentang sumber yang relevan yang digunakan untuk mengumpulkan informasi yang diperlakukan dalam penelitian. Studi pustaka dengan mengumpulkan sumber pustaka yang dapat berupa buku, teks, makalah dan sebagainya. Setelah sumber pustaka terkumpul dilanjutkan dengan penelaahan sumber pustaka tersebut. Pada akhirnya sumber pustaka itu dijadikan landasan untuk menganalisis permasalahan.

Data yang dikumpulkan pada penelitian ini berupa data primer dan data sekunder. Data primer berupa data tentang volume lalu lintas jalan, waktu tempuh rata-rata, dan kepadatan jalan yang diambil secara langsung di lapangan. Data sekunder diambil dari Dinas Pemadam Kebakaran Kota Semarang berupa pos pemadam kebakaran dan daerah rawan kebakaran kota Semarang serta data sekunder yang diperoleh dari Dinas Perhubungan Komunikasi dan Informatika Kota Semarang yaitu peta jaringan jalan kota Semarang.

Langkah-langkah yang digunakan dalam pemecahan masalah adalah sebagai berikut: (1) membuat graf berbobot pos pemadam kebakaran Kota Semarang dengan bobot tingkat kemacetan yang diperoleh dari *output fuzzy* dari parameter panjang jalan dan kepadatan jalan, (2) menemukan penyelesaian dari penerapan algoritma Dijkstra dalam menentukan rute terbaik dari pos pemadam kebakaran kota Semarang menuju daerah rawan kebakaran kota Semarang,



Gambar 1. Graf yang menggambarkan rute jaringan jalan pemadam kebakaran

(3) pembuatan aplikasi pencarian rute terbaik dengan algoritma Dijkstra menggunakan pemrograman PHP.

Penarikan kesimpulan didasarkan pada studi pustaka dan pemecahan masalah. Simpulan yang diperoleh merupakan hasil analisis dari penelitian. Setelah itu dilakukan pemberian saran atas masalah yang terjadi.

HASIL DAN PEMBAHASAN

Penelitian ini mengkaji tentang rute terbaik untuk petugas pemadam kebakaran. Permasalahan dalam penelitian ini adalah bagaimana penyelesaian dari penerapan algoritma Dijkstra dalam menentukan rute dari pos pemadam kebakaran Kota Semarang menuju daerah rawan kebakaran. Dalam penelitian ini dicari rute yang dilalui petugas pemadam kebakaran dari pos pemadam kebakaran menuju ke daerah rawan kebakaran dengan bobot gabungan dari parameter panjang jalan dan kepadatan jalan yang menghasilkan *fuzzy output* berupa tingkat kemacetan jalan. Selain itu, akan dibuat aplikasi penerapan algoritma Dijkstra untuk mencari rute dari pos pemadam kebakaran menuju daerah rawan kebakaran di kota Semarang menggunakan bahasa pemrograman PHP.

Proses pencarian rute terbaik menggunakan algoritma Dijkstra akan dijelaskan pada kasus dibawah ini dengan 6 pos pemadam kebakaran dan 7 daerah rawan kebakaran dengan resiko kebakaran 3 (Dinas Pemadam Kebakaran Kota Semarang). Berikut adalah gambar jaringan jalan yang menghubungkan pos-pos pemadam kebakaran

menuju semua daerah rawan kebakaran kota Semarang terdapat pada Gambar 1. Terdapat 28 titik dengan 6 titik pos pemadam kebakaran, 7 titik daerah rawan kebakaran dan 15 titik persimpangan jalan. Sisi berarah menunjukkan arah untuk setiap jalan yang dilalui.

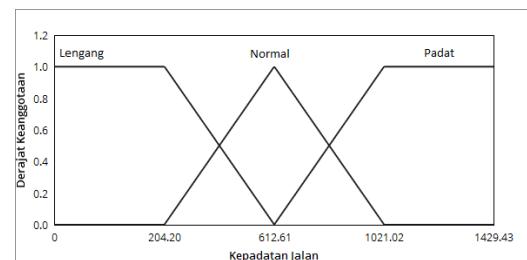
Parameter bobot sisi pada penelitian ini, yaitu tingkat kemacetan yang diperoleh dari *output fuzzy*. *Output fuzzy* diperoleh dari input parameter panjang jalan dan kepadatan jalan. Tabel 1 merupakan besaran panjang jalan dan kepadatan jalan untuk setiap sisi.

Berikut tahapan untuk mendapatkan *output fuzzy* dari parameter panjang jalan dan kepadatan jalan.

Tahap pertama adalah Pembentukan fungsi keanggotaan *fuzzy*.

Berikut fungsi keanggotaan untuk parameter panjang jalan dan kepadatan jalan.

a. Fungsi derajat keanggotaan kepadatan jalan. Pada fungsi keanggotaan kepadatan jalan, menggambarkan derajat keanggotaan yang dimiliki oleh kepadatan jalan, yang memiliki tiga derajat keanggotaan, antara lain: lancar, normal dan padat. Gambar 2 merupakan fungsi keanggotaan untuk kepadatan jalan.



Gambar 2. Fungsi keanggotaan kepadatan jalan

Berikut adalah fungsi linier yang dimiliki oleh kepadatan jalan.

$$\mu_{jlnLENGANG}(L) = \begin{cases} \frac{1}{408,41} ; L \leq 204,20 \\ \frac{612,61 - L}{408,41} ; 204,20 \leq L \leq 612,61 \\ 0 ; L \geq 612,61 \end{cases}$$

$$\mu_{jlnNORMAL}(L) = \begin{cases} 0 ; L \leq 204,20 \text{ atau } L \geq 1021,02 \\ \frac{L - 204,20}{408,41} ; 204,20 \leq L \leq 612,61 \\ \frac{1021,02 - L}{408,41} ; 612,61 \leq L \leq 1021,02 \end{cases}$$

$$\mu_{jlnPADAT}(L) = \begin{cases} 0 ; L \leq 612,61 \\ \frac{L - 612,61}{408,41} ; 612,61 \leq L \leq 1021,02 \\ 1 ; L \geq 1021,02 \end{cases}$$

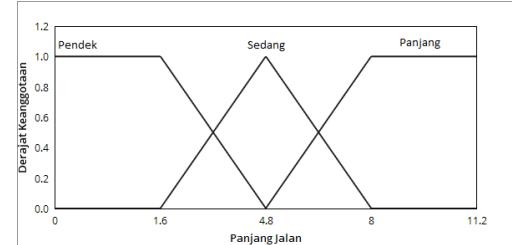
L = kepadatan jalan

Tabel 1. Panjang dan kepadatan jalan

Sisi	Jarak(km)	Kepadatan(smp)	Sisi	Jarak(km)	Kepadatan(smp)
v ₁ v ₂	9,6	1195,01	v ₁₅ v ₂₇	1,6	349,52
v ₁ v ₃	3,8	403,22	v ₁₅ v ₂₈	3,8	283,14
v ₂ v ₁	9,6	1195,01	v ₁₆ v ₉	1,8	427,43
v ₂ v ₅	3,4	361,62	v ₁₆ v ₂₇	1,2	156,16
v ₃ v ₁	3,8	403,22	v ₁₇ v ₇	2,9	258,71
v ₂ v ₄	1,6	169,78	v ₁₇ v ₁₀	1,9	149,53
v ₄ v ₆	2,05	414,37	v ₁₇ v ₁₈	2,2	161,66
v ₄ v ₃	1,6	169,78	v ₁₈ v ₁₇	2,6	152,44
v ₂ v ₂	3,5	372,26	v ₁₈ v ₁₉	3,3	234,63
v ₅ v ₆	4,7	307,8	v ₁₈ v ₂₁	3,8	1160,48
v ₅ v ₁₁	4,5	448,16	v ₁₈ v ₂₇	4,4	961,18
v ₆ v ₄	2,9	586,18	v ₁₉ v ₈	3,35	158,46
v ₆ v ₅	4,4	288,16	v ₁₉ v ₁₈	2,2	156,42
v ₂ v ₇	2,1	238,35	v ₁₉ v ₂₀	2,3	108,79
v ₆ v ₈	4,6	217,58	v ₂₀ v ₁₉	1,8	85,14
v ₆ v ₁₁	2,6	209,4	v ₂₀ v ₂₁	1,8	85,14
v ₇ v ₆	2,1	238,35	v ₂₁ v ₁₈	3,8	1160,48
v ₇ v ₈	3,8	176,13	v ₂₁ v ₂₀	1,8	85,14
v ₇ v ₉	0,85	127,82	v ₂₁ v ₂₂	3,8	1160,48
v ₇ v ₁₁	2,4	113,62	v ₂₁ v ₂₃	3,4	87,48
v ₇ v ₁₇	2,9	258,71	v ₂₂ v ₂₁	3,8	1160,48
v ₈ v ₆	4,6	217,58	v ₂₂ v ₂₃	2,4	61,75
v ₈ v ₇	4,3	199,31	v ₂₃ v ₂₁	4,2	108,07
v ₈ v ₁₉	3,1	146,63	v ₂₃ v ₂₂	2,4	61,75
v ₉ v ₇	1,2	180,46	v ₂₃ v ₂₄	1,1	28,3
v ₉ v ₁₀	0,55	43,29	v ₂₄ v ₂₃	1,2	30,88
v ₉ v ₁₆	1,7	403,68	v ₂₄ v ₂₅	5,2	133,8
v ₁₀ v ₉	0,55	43,29	v ₂₄ v ₂₆	5,6	570,08
v ₁₀ v ₁₇	1,9	149,53	v ₂₅ v ₂₄	5,2	133,8
v ₁₁ v ₅	4,5	448,16	v ₂₅ v ₂₆	2	428,94
v ₁₁ v ₆	2,5	201,35	v ₂₆ v ₂₄	5,3	539,54
v ₁₁ v ₇	2,3	108,88	v ₂₆ v ₂₅	2,1	450,39
v ₁₁ v ₁₂	4,5	1082,88	v ₂₆ v ₂₇	4,8	1029,46
v ₁₁ v ₁₆	2,1	108,91	v ₂₆ v ₂₈	2,7	211,14
v ₁₂ v ₁₁	4,1	986,62	v ₂₇ v ₁₅	1,9	415,06
v ₁₂ v ₁₃	3,8	1108,54	v ₂₇ v ₁₆	1,3	169,17
v ₁₂ v ₁₅	2,3	350,89	v ₂₇ v ₁₈	3,8	830,11
v ₁₃ v ₁₂	4,2	1225,22	v ₂₇ v ₂₆	4,8	1029,46
v ₁₃ v ₁₄	0,17	1,56	v ₂₈ v ₁₅	3,8	283,14
v ₁₄ v ₁₃	0,17	1,56	v ₂₈ v ₂₆	2,8	218,96
v ₁₅ v ₁₂	2,4	366,14			

- b. Fungsi derajat keanggotaan panjang jalan. Pada fungsi keanggotaan panjang jalan, menggambarkan derajat keanggotaan yang dimiliki oleh panjang jalan, yang memiliki

tiga derajat keanggotaan, antara lain, pendek, sedang dan panjang. Gambar 3 merupakan fungsi keanggotaan untuk panjang jalan.



Gambar 3. Fungsi keanggotaan panjang jalan

Berikut adalah fungsi linier yang dimiliki oleh panjang jalan.

$$\mu_{jlnPENDEK}(P) = \begin{cases} \frac{1}{3,2} ; P \leq 1,6 \\ \frac{4,8 - P}{3,2} ; 1,6 \leq P \leq 4,8 \\ 0 ; P \geq 4,8 \end{cases}$$

$$\mu_{jlnSEDANG}(P) = \begin{cases} 0 ; P \leq 1,6 \text{ atau } P \geq 8 \\ \frac{P - 1,6}{3,2} ; 1,6 \leq P \leq 4,8 \\ \frac{8 - P}{3,2} ; 4,8 \leq P \leq 8 \end{cases}$$

$$\mu_{jlnPANJANG}(P) = \begin{cases} 0 ; P \leq 4,8 \\ \frac{P - 4,8}{3,2} ; 4,8 \leq P \leq 8 \\ 1 ; P \geq 8 \end{cases}$$

P = panjang jalan

Tahap kedua adalah pembentukan aturan fuzzy. Terdapat 9 aturan fuzzy, berikut adalah aturan fuzzy pada penelitian ini.

- (R1) Jika jalan pendek dan kepadatan jalan lengang maka output fuzzy lancar.
- (R2) Jika jalan pendek dan kepadatan jalan normal maka output fuzzy sedang.
- (R3) Jika jalan pendek dan kepadatan jalan padat maka output fuzzy sedikit padat.
- (R4) Jika jalan sedang dan kepadatan jalan lengang maka output fuzzy sedang.
- (R5) Jika jalan sedang dan kepadatan jalan normal maka output fuzzy sedikit padat.
- (R6) Jika jalan sedang dan kepadatan jalan padat maka output fuzzy padat merayap.
- (R7) Jika jalan panjang dan kepadatan jalan lengang maka output fuzzy sedikit padat.
- (R8) Jika jalan panjang dan kepadatan jalan normal maka output fuzzy padat merayap.

(R9) Jika jalan panjang dan kepadatan jalan padat maka output fuzzy sangat padat/macet.

Tahap selanjutnya adalah mencari *output fuzzy*. Untuk mendapatkan *output fuzzy* digunakan metode berbobot rata-rata. Berikut adalah rumus metode berbobot rata-rata.

$$z = \frac{(\alpha_{pred_1}(T.K_1) + \alpha_{pred_2}(T.K_2) + \dots + \alpha_{pred_n}(T.K_n))}{\alpha_{pred_1} + \alpha_{pred_2} + \dots + \alpha_{pred_n}}$$

$TK = \text{tingkat kemacetan}$

Nilai tingkat kemacetan:

Lancar = 0,1

Sedang = 0,25

Sedikit padat = 0,5

Padat merayap = 0,75

Macet = 1.

Setiap parameter sisi yang berupa panjang jalan dan kepadatan jalan dimasukkan kedalam fungsi keanggotaan masing-masing.

Berikut adalah input untuk sisi $v_1 v_2$ dengan nilai panjang jalan 9,6 km dan kepadatan 1195,01 smp/km.

Input hasil $v_1 v_2$

- Input jarak = 9,6
 $\mu_{jlnPENDEK}(P) = 0$
 $\mu_{jlnSEDANG}(P) = 0$
 $\mu_{jlnPANJANG}(P) = 1$
- Input kepadatan = 1195,01
 $\mu_{jlnLENGANG}(L) = 0$
 $\mu_{jlnNORMAL}(L) = 0$
 $\mu_{jlnPADAT}(L) = 1$

Selanjutnya mencari *alpha - predikat* dari setiap aturan *fuzzy*.

(R1) IF panjang jalan pendek AND kepadatan lengang THEN Tingkat kemacetan lancar

$$\alpha - \text{predikat}_1 = \min \{\mu_{jlnPENDEK}(9,6); \mu_{jlnLENGANG}(1195,01)\} \\ = \min\{0; 0\} = 0$$

$$TK_1 = 0,1$$

(R2) IF panjang jalan pendek AND kepadatan normal THEN Tingkat kemacetan sedang (0,25)

$$\alpha - \text{predikat}_2 = \min \{\mu_{jlnPENDEK}(9,6); \mu_{jlnNORMAL}(1195,01)\} \\ = \min\{0; 0\} = 0$$

$$TK_2 = 0,25$$

(R3) IF panjang jalan pendek AND kepadatan padat THEN Tingkat kemacetan sedikit padat (0,5)

$$\alpha - \text{predikat}_3 = \min \{\mu_{jlnPENDEK}(9,6); \mu_{jlnPADAT}(1195,01)\} \\ = \min\{0; 1\} = 0$$

$$TK_3 = 0,5$$

(R4) IF panjang jalan sedang AND kepadatan lengang THEN Tingkat kemacetan sedang (0,25)

$$\alpha - \text{predikat}_4 = \min \{\mu_{jlnSEDANG}(9,6); \mu_{jlnLENGANG}(1195,01)\} \\ = \min\{0; 0\} = 0$$

$$TK_4 = 0,25$$

(R5) IF panjang jalan sedang AND kepadatan normal THEN Tingkat kemacetan sedikit padat (0,5)

$$\alpha - \text{predikat}_5 = \min \{\mu_{jlnSEDANG}(9,6); \mu_{jlnNORMAL}(1195,01)\} \\ = \min\{0; 0\} = 0$$

$$TK_5 = 0,5$$

(R6) IF panjang jalan sedang AND kepadatan padat THEN Tingkat kemacetan padat merayap (0,75)

$$\alpha - \text{predikat}_6 = \min \{\mu_{jlnSEDANG}(9,6); \mu_{jlnPADAT}(1195,01)\} \\ = \min\{0; 1\} = 0$$

$$TK_6 = 0,75$$

(R7) IF panjang jalan panjang AND kepadatan lengang THEN Tingkat kemacetan sedikit padat (0,5)

$$\alpha - \text{predikat}_7 = \min \{\mu_{jlnPANJANG}(9,6); \mu_{jlnLENGANG}(1195,01)\} \\ = \min\{1; 0\} = 0$$

$$TK_7 = 0,5$$

(R8) IF panjang jalan panjang AND kepadatan normal THEN Tingkat kemacetan padat merayap (0,75)

$$\alpha - \text{predikat}_8 = \min \{\mu_{jlnPANJANG}(9,6); \mu_{jlnNORMAL}(1195,01)\} \\ = \min\{1; 0\} = 0$$

$$TK_8 = 0,75$$

[R9] IF panjang jalan panjang AND kepadatan padat THEN Tingkat kemacetan macet (0,1)

$$\alpha - \text{predikat}_9 = \min \{\mu_{jlnPANJANG}(9,6); \mu_{jlnPADAT}(1195,01)\} \\ = \min\{1; 1\} = 1$$

$$TK_9 = 1$$

Selanjutnya gunakan metode berbobot rata-rata untuk memperoleh nilai *output fuzzy* dari $v_1 v_2$. Karena $\alpha - \text{predikat}$ yang tidak nol hanya terdapat pada aturan (R9), maka nilai *output fuzzy* dari $v_1 v_2$ adalah:

$$z = \frac{1 \times (1)}{1} = 1$$

Dari hasil inferensi fuzzy nilai tingkat kemacetan $v_1 v_2$ adalah 1 atau dalam kategori macet. Lanjutkan untuk setiap sisi. Tabel 2 adalah hasil untuk semua sisi.

Tabel 2. Hasil Output Fuzzy

Sisi	Panjang Jalan (km)	Kepadatan Jalan (smp)	Output Fuzzy			
v_1v_2	9,6	1195,01	1	$v_{27}v_{26}$	4,8	1029,46
v_1v_3	3,8	403,22	0,296	$v_{28}v_{15}$	3,8	283,14
v_2v_1	9,6	1195,01	1	$v_{28}v_{26}$	2,8	218,96
v_2v_5	3,4	361,62	0,267			0,251
v_3v_1	3,8	403,22	0,296			0,171
v_3v_4	1,6	169,78	0,1			
v_4v_6	2,05	414,37	0,221			
v_4v_3	1,6	169,78	0,1			
v_5v_2	3,5	372,26	0,273			
v_5v_6	4,7	307,80	0,305			
v_5v_{11}	4,5	448,16	0,364			
v_6v_4	2,9	586,18	0,331			
v_6v_5	4,4	288,16	0,276			
v_6v_7	2,1	238,35	0,159			
v_6v_8	4,6	217,58	0,249			
v_6v_{11}	2,6	209,40	0,153			
v_7v_6	2,1	238,35	0,159			
v_7v_8	3,8	176,13	0,203			
v_7v_9	0,85	127,82	0,1			
v_7v_{11}	2,4	113,62	0,138			
v_7v_{17}	2,9	258,71	0,206			
v_8v_6	4,6	217,58	0,249			
v_8v_7	4,3	199,31	0,227			
v_8v_{19}	3,1	146,63	0,170			
v_9v_7	1,2	180,46	0,1			
v_9v_{10}	0,55	43,29	0,1			
v_9v_{16}	1,7	403,68	0,185			
$v_{10}v_9$	0,55	43,29	0,1			
$v_{10}v_{17}$	1,9	149,53	0,114			
$v_{11}v_5$	4,5	448,16	0,364			
$v_{11}v_6$	2,5	201,35	0,142			
$v_{11}v_7$	2,3	108,88	0,133			
$v_{11}v_{12}$	4,5	1082,88	0,727			
$v_{11}v_{16}$	2,1	108,91	0,123			
$v_{12}v_{11}$	4,1	986,62	0,649			
$v_{12}v_{13}$	3,8	1108,54	0,672			
$v_{12}v_{15}$	2,3	350,89	0,221			
$v_{13}v_{12}$	4,2	1225,22	0,703			
$v_{13}v_{14}$	0,17	1,56	0,1			
$v_{14}v_{13}$	0,17	1,56	0,1			
$v_{15}v_{12}$	2,4	366,14	0,231			
$v_{15}v_{27}$	1,6	349,52	0,153			
$v_{15}v_{28}$	3,8	283,14	0,251			
$v_{16}v_9$	1,8	427,43	0,203			
$v_{16}v_{27}$	1,2	156,16	0,1			
$v_{17}v_7$	2,9	258,71	0,206			
$v_{17}v_{10}$	1,9	149,53	0,114			
$v_{17}v_{18}$	2,2	161,66	0,128			
$v_{18}v_{17}$	2,6	152,44	0,147			
$v_{18}v_{19}$	3,3	234,63	0,205			
$v_{18}v_{21}$	3,8	1160,48	0,672			
$v_{18}v_{27}$	4,4	961,18	0,646			
$v_{19}v_8$	3,35	158,46	0,182			
$v_{19}v_{18}$	2,2	156,42	0,128			
$v_{19}v_{20}$	2,3	108,79	0,133			
$v_{20}v_{19}$	1,8	85,14	0,109			
$v_{20}v_{21}$	1,8	85,14	0,109			
$v_{21}v_{18}$	3,8	1160,48	0,672			
$v_{21}v_{20}$	1,8	85,14	0,109			
$v_{21}v_{22}$	3,8	1160,48	0,672			
$v_{21}v_{23}$	3,4	87,48	0,184			
$v_{22}v_{21}$	3,8	1160,48	0,672			
$v_{22}v_{23}$	2,4	61,75	0,138			
$v_{23}v_{21}$	4,2	108,07	0,222			
$v_{23}v_{22}$	2,4	61,75	0,138			
$v_{23}v_{24}$	1,1	28,30	0,1			
$v_{24}v_{23}$	1,2	30,88	0,1			
$v_{24}v_{25}$	5,2	133,80	0,281			
$v_{24}v_{26}$	5,6	570,08	0,530			
$v_{25}v_{24}$	5,2	133,80	0,281			
$v_{25}v_{26}$	2	428,94	0,221			
$v_{26}v_{24}$	5,3	539,54	0,496			
$v_{26}v_{25}$	2,1	450,39	0,234			
$v_{26}v_{27}$	4,8	1029,46	0,75			
$v_{26}v_{28}$	2,7	211,14	0,159			
$v_{27}v_{15}$	1,9	415,06	0,207			
$v_{27}v_{16}$	1,3	169,17	0,1			
$v_{27}v_{18}$	3,8	830,11	0,534			

Bobot yang telah diperoleh kemudian dimasukkan kedalam graf jaringan jalan pos pemandam kebakaran, sehingga diperoleh graf dengan bobot tingkat kemacetan (nilai *output fuzzy*). Selanjutnya dicari rute terbaik menggunakan algoritma Dijkstra.

Pencarian rute terbaik diperoleh dengan menggunakan algoritma Dijkstra. Langkah-langkah pada algoritma Dijkstra sebagai berikut.

Input : Graf bobot G dengan $s, t \in V(G)$.

Step 1 : Label titik dengan $\lambda(s) = 0$ dan untuk setiap titik v di G selain s , label titik v dengan $\lambda(v) = \infty$. (dalam praktik ∞ diganti dengan bilangan yang sangat besar). Tulis $T = V(G)$.

Pencarian rute terbaik menggunakan aplikasi yang dibangun dengan bahasa pemrograman PHP bertujuan untuk mempersingkat waktu pencarian rute. Berikut adalah langkah pencarian pada program.

- Pada menu utama pilih menu pencarian rute.

Step 2 : Misalkan $u \in T$ dengan $\lambda(u)$ minimum.

Step 3 : Jika $u = t$, berhenti, berarti panjang lintasan terpendek dari s ke t adalah $\lambda(t)$.

Step 4 : Untuk setiap sisi $e = uv, v \in T$; ganti label v dengan $\lambda(v) = \min\{\lambda(v), \lambda(u) + w(e)\}$.

Step 5 : Tulis $T = T - \{u\}$, dan kembali ke step 2. (Budayasa, 2007).

Berdasarkan hasil pencarian rute terbaik dengan algoritma Dijkstra diperoleh rute terbaik untuk masing-masing pos pemandam seperti pada Tabel 3.

- Pilih kasus rute yang akan dipilih, yaitu rute dengan bobot tingkat kemacetan.
- Klik pada menu mulai.
- Tentukan titik awal dan titik tujuan rute terbaik yang akan dicari.
- Klik tombol cari

Gambar 4 merupakan tampilan dari program

Tabel 3. Hasil rute terbaik

Lokasi	Pos Pemadam		
Gardu Induk Listrik Tugu	Pos Tugu (v3) v3,v4	Pos Madukoro (v7) v7,v6,v4	Pos Balaikota (v10) v10,v9,v7,v6,v4
TPA Jatibarang	v3,v4,v6,v2	v7,v6,v5,v2	v10,v9,v7,v6,v5,v2
Bandara A. Yani	v3,v4,v6	v7,v6	v10,v9,v7,v6
Gardu Induk Listrik Srondol	v3,v4,v6,v11,v16,v27,v15,v12,v13,v14	v7,v11,v16,v27,v15,v12,v13,v14	v10,v9,v16,v27,v15,v12,v13,v14
Depo Pertamina Pengapon	v3,v4,v6,v7,v17,v18	v7,v11,v18	v10,v17,v18
Lap. Penumpukan & Pembangkit Listrik	v3,v4,v6,v8,v19	v7,v8,v19	v10,v17,v18,v19
IPLT Tanggung Rejo	v3,v4,v6,v8,v19,v20	v7,v8,v19,v20	v10,v17,v18,v19,v20
Lokasi	Pos Pemadam		
Gardu Induk Listrik Tugu	Pos Banyumanik (v13) v13,v12,v11,v6,v4	Pos Plamongan (v25) v25,v24,v23,v21,v20,v19,v8,v6,v4	Pos Terboyo (v22) v22,v23,v21,v20,v19,v8,v6,v4
TPA Jatibarang	v13,v12,v11,v5,v2	v25,v24,v23,v21,v20,v19,v8,v6,v5,v2	v22,v23,v21,v20,v19,v8,v6,v5,v2
Bandara A. Yani	v13,v12,v11,v6	v25,v24,v23,v21,v20,v19,v8,v6	v22,v23,v21,v20,v19,v8,v6
Gardu Induk Listrik Srondol	v13,v14	v25,v26,v28,v15,v12,v13,v14	v22,v23,v24,v25,v26,v28,v15,v12,v13,v14
Depo Pertamina Pengapon	v13,v12,v15,v27,v18	v25,v24,v23,v21,v20,v19,v18	v22,v23,v21,v20,v19,v18
Lap. Penumpukan & Pembangkit Listrik	v13,v12,v15,v27,v18,v19	v25,v24,v23,v21,v20,v19	v22,v23,v21,v20,v19
IPLT Tanggung Rejo	v13,v12,v15,v27,v18,v19,v20	v25,v24,v23,v21,v20	v22,v23,v21,v20



Gambar 4. Tampilan Hasil Program

PENUTUP

Berdasarkan hasil penelitian dan pembahasan tentang penerapan algoritma Dijkstra dalam menentukan rute terbaik dari pos pemadam kebakaran di Semarang menuju ke daerah rawan kebakaran di Semarang, dapat ditarik kesimpulan sebagai berikut: (1) rute pos pemadam kebakaran di kota Semarang dapat direpresentasikan sebagai graf pada Gambar 1 (2) rute terbaik dapat dihasilkan dari beberapa parameter jalan menggunakan logika *fuzzy*, diantaranya panjang jalan dan kepadatan jalan (3) hasil pencarian rute terbaik dengan cara perhitungan manual maupun dengan program memiliki solusi sama.

DAFTAR PUSTAKA

- Budayasa, I. K. (2007). *Teori Graph dan Aplikasinya*. Surabaya: Unesa University Press.
- Chen, Y., Shen, S., & Yang, R. (2014). Path Optimization Study for Vehicles Evacuation Based on Dijkstra algorithm. *Elsevier*, (71): 159-165. doi: 10.1016/j.proeng.2014.04.023
- Faro, A., & Giordano, D. (2016). Algorithm to Find Shortest and Alternative Path in Free Flow and Congested Traffic Regimes. Italy: *Elsevier B.V. Journal of Electrical, Electronics and Computer Engineering*, C(73): 1-29. Doi:10.1016/j.trc.2016.09.009
- Fitriana, N. E. & Sugiharti, E. (2015). Implementasi Algoritma Genetika dengan Teknik Kendali Logika Fuzzy untuk Mengatasi Travelling Salesman Problem menggunakan Matlab. *UNNES Journal of Mathematics*, 4(2): 115-121.
- Ichsan, M. H., Yudaningtyas, E., & Muslim, M. A. (2012). Solusi Optimal Pencarian Jalur Tercepat dengan Algoritma Hybrid Fuzzy Dijkstra. *EECCIS*, 6(2) : 155-160.
- Mardlootillah, H. I., Suyitno, A., & Arini, F. Y. (2015). Simulasi Algoritma Dijkstra dalam Menangani Masalah Lintasan Terpendek pada Graf Menggunakan Visual Basic. *UNNES. Journal of Mathematics*, 4(2): 91-96. [<http://journal.unnes.ac.id/sju/index.php/ujm/article/view/9349>]
- Handaka, M. S. (2011). *Perbandingan Algoritma Dijkstra (Greedy), Bellman-Ford (BFS-DFS), dan Floyd-Warshall (Dynamic Programming) dalam Pengaplikasian Lintasan Terpendek pada Link-State Routing Protocol*. Bandung: Institut Teknologi Bandung.
- Munir, R. (2005). *Matematika Diskrit*. Bandung: Informatika.
- Munir, R. (2010). *Matematika Diskrit Revisi Keempat*. Bandung: Informatika.
- Patel, V., & Chitra, B. (2014). A Survey Paper of Bellman-Ford Algorithm and Dijkstra Algorithm for Finding Shortest Path in GIS Application. *International Journal of P2P Network Trends and Technology*, 5: 1-4. Tersedia di <http://www.ijpttjournal.org/volume-5/IJPTT-V5P401.pdf> [27 Maret 2016].
- Peraturan Pemerintah. (2006). *UU Nomor 34 Tahun 2006 tentang Jalan*. Indonesia: Undang-Undang Republik Indonesia.
- Peraturan Pemerintah. (2007). *UU Nomor 27 Tahun 2007 tentang Penanggulangan Bencana*. Indonesia: Undang-Undang Republik Indonesia.
- Peyer, S., Rautenbach, D., & Vygen, J. (2009). A Generalization of Dijkstra's Shortest Path Algorithm Applications to VLSI routing. Germany: Elsevier B.V. *Journal of Discrete Algorithm*, 4(7): 377-390. Doi:10.1016/j.jda.2007.08.003.
- Prasetyo, V. Z., Suyitno, A., & Mashuri. (2013). Penerapan Algoritma Dijkstra dan Prim pada Pendistribusian Air di PDAM kota Demak. Indonesia: *UNNES Journal of Mathematics*, 2(1): 72-78. [<https://journal.unnes.ac.id/sju/index.php/ujm/article/view/1714>]
- Purwanto, Y., Purwitasari, A., & Wibowo, A. W. (2005). Implementasi dan Analisis Algoritma Pencarian Rute Terpendek di Kota Surabaya. *Jurnal Penelitian dan Pengembangan TELEKOMUNIKASI*, 10 (2): 94-101.
- Taufiq, M. (2015). *Menentukan Rute Terpendek dengan Memanfaatkan Metode Heuristik Berbasis Algoritma A**. Semarang: Universitas Negeri Semarang.

Triato, A. (2013). *Analisis Prioritas Lokasi Pembangunan Infrastruktur Pos Pemadam Kebakaran Kabupaten Kendal*. Proposal Thesis. Semarang: Jurusan Manajemen dan Rekayasa Infrastruktur UNDIP.

Yulianto, A. W., Suyitno, H., & Mashuri. (2012). Aplikasi Fuzzy Linear Programming dalam Optimalisasi Produksi. *UNNES Journal of Mathematics*, 1 (1): 8-14.