

PENGGUNAAN ALGORITMA DIJKSTRA DALAM PERENCANAAN RUTE EVAKUASI BENCANA LONGSOR DI KOTA SEMARANG

R. A. Pramudya^a, Subiyanto^a

^a Universitas Negeri Semarang, Indonesia

Article Info:

Received: 3 September 2015
in revised form: 25 September 2015
Accepted: 1 Oktober 2015
Available Online: 31 Oktober 2015

Keywords:

Dijkstra algorithm, network analyst, avenue script, evacuation route

Corresponding Author:

Ryan Arya Pramudya
Universitas Negeri Semarang,
Semarang, Indonesia
Email:
ryanaryapramudya@gmail.com

Info Artikel:

Diterima: 3 September 2015
Hasil Revisi: 25 September 2015
Disetujui: 1 Oktober 2015
Publikasi On-Line: 31 Oktober 2015

Kata Kunci:

Algoritma Dijkstra, Network Analyst, Avenue Script, Rute Evakuasi

Kontak Penulis:

Ryan Arya Pramudya
Universitas Negeri Semarang,
Semarang, Indonesia
Email:
ryanaryapramudya@gmail.com

Abstract: Finding fastest route is necessary for evacuation process when a disaster happened. Route that generated slightly made evacuation process easily. The availability about route distance and line direction can be considered when route finding process takes place. Geographic information systems can be used with Dijkstra algorithms in the network analyst as a method to find a route that serves to reduce losses that occur when the landslide occurred. In combination with ArcView system where there is an avenue script that can be modified to build a system. By utilizing Dijkstra algorithms in the network analyst, the search becomes faster and can provide information on how long the evacuation route should be, which will help the government in the decision-making process of disaster mitigation. Validation of the system is done by comparing manual counting with Dijkstra algorithms and automatic counting by the network analyst. The difference between manual counting and network analyst as far as two meters, so that the system closer to the real and could be used for determining evacuation route.

Abstrak: Pencarian rute tercepat diperlukan saat melakukan proses evakuasi ketika terjadi suatu bencana. Rute yang dihasilkan dapat mempermudah dalam proses evakuasi. Ketersediaan informasi mengenai jarak rute dan panduan jalan dapat menjadi suatu hal yang perlu dipertimbangkan saat proses pencarian rute berlangsung. Sistem informasi geografis dapat digunakan dengan menggunakan algoritma dijkstra yang ada pada network analyst sebagai metode dalam mencari rute yang berfungsi untuk mengurangi kerugian yang terjadi saat bencana tanah longsor terjadi. Dalam Sistem dikombinasikan dengan ArcView dimana terdapat avenue script yang dapat dimodifikasi untuk membangun sebuah sistem. Dengan memanfaatkan algoritma dijkstra pada network analyst, pencarian rute menjadi lebih cepat dan dapat memberikan informasi mengenai seberapa panjang rute evakuasi sehingga akan membantu pemerintah dalam proses pengambilan keputusan mitigasi bencana. Validasi sistem dilakukan dengan melakukan penghitungan manual, hasil dari perhitungan itu akan dikomparasikan dengan hasil proses network analyst. Selisih antara penghitungan manual dan network analyst terpaut sejauh dua meter sehingga sistem mendekati riil dan dapat digunakan untuk acuan penentuan rute evakuasi bencana.

Copyright © 2015 GJGP-UNDIP

This open access article is distributed under a Creative Commons Attribution (CC-BY-NC-SA) 4.0 International license.

How to cite (APA 6th Style):

Pramudya, R. A., & Subiyanto. (2015). Penggunaan algoritma Dijkstra dalam perencanaan rute evakuasi bencana longsor di Kota Semarang. *Geoplanning: Journal of Geomatics and Planning*, 2(2), 93-102. doi:10.14710/geoplanning.2.2.93-102

1. PENDAHULUAN

Pemanfaatan perkembangan teknologi informasi dapat diaplikasikan sebagai salah satu cara untuk melakukan mitigasi bencana. Menurut Undang-Undang nomor 24 tahun 2007 tentang penanggulangan bencana pada BAB I pasal I menyebutkan bahwa mitigasi adalah serangkaian upaya untuk mengurangi risiko bencana, baik melalui pembangunan fisik maupun penyadaran dan peningkatan kemampuan

menghadapi ancaman bencana. Dalam lampiran peraturan Menteri Dalam Negeri nomor 33 tahun 2006 menyebutkan ada empat hal penting dalam mitigasi bencana, yaitu:

- Tersedia informasi dan peta kawasan rawan bencana untuk tiap jenis bencana;
- Sosialisasi untuk meningkatkan pemahaman dan kesadaran masyarakat dalam menghadapi bencana, karena bermukim di daerah rawan bencana;
- Mengetahui apa yang perlu dilakukan dan dihindari, serta mengetahui cara penyelamatan diri jika bencana timbul;
- Pengaturan dan penataan kawasan rawan bencana untuk mengurangi ancaman bencana. Salah satu bentuk peningkatan kemampuan dalam menghadapi ancaman bencana dengan memanfaatkan teknologi informasi yang tersedia.

Penggunaan teknologi geospasial dapat menunjang mitigasi bencana dalam bentuk penyediaan informasi dan peta kawasan rawan bencana. Pemanfaatan teknologi tersebut diaplikasikan dengan menggunakan SIG yang dapat diartikan sebagai sebuah sistem informasi yang berfungsi untuk memasukkan, menyimpan, memanggil kembali, mengolah, menganalisis dan menghasilkan data dengan orientasi geografis. SIG memegang kemampuan dalam mengintegrasikan peta dengan detail informasi *database* dan gambar serta mampu mengubah peta biasa menjadi peta pintar yang dapat mendukung dalam analisis kompleks (Hongchun, 2012). Penggunaan peta sangat bermanfaat dalam proses pencarian rute. Dalam peta jalan dapat berisi informasi *data attribut* yang dapat menunjang proses pencarian rute evakuasi. *Data attribut* tersebut dapat berisi informasi mengenai panjang jalan dan kode jalan. Informasi mengenai panjang jalan akan digunakan sebagai bobot dalam mencari rute terpendek dengan menggunakan *network analyst* yang berbasis *algoritma dijkstra*.

Algoritma Dijkstra digunakan sebagai acuan dasar dalam algoritma rute terpendek dengan sumber tunggal (Kai et al, 2014). Algoritma Dijkstra merupakan algoritma yang lebih efisien daripada algoritma Warshall (Siang, 2004). Node ketetangaan pada algoritma Dijkstra efisien untuk menghitung data yang besar serta menggunakan ruang yang sedikit (Fuhao dan Jiping, 2009). Algoritma dijkstra digunakan dalam *network analyst* sebagai metode untuk mencari rute terpendek. Sistem dibangun dengan memanfaatkan algoritma dijkstra yang terdapat pada *network analyst* dengan tujuan untuk memberikan informasi mengenai jalur yang dapat dipakai saat proses evakuasi terjadi. Validasi sistem dilakukan dengan membandingkan antara perhitungan algoritma dijkstra manual dan secara otomatis dengan *network analyst*.

2. DATA DAN METODE

Dalam membangun Sistem Informasi Geografis (SIG) jalur evakuasi, hal mendasar yang perlu dipersiapkan adalah ketersediaan data yang digunakan saat proses pencarian rute. Data yang telah dikumpulkan akan diproses menggunakan *network analyst*. Penghitungan manual dilakukan untuk membuktikan bahwa jarak yang dihasilkan adalah valid dengan mengambil titik lain selain titik bencana dan titik shelter. Penggunaan *avenue script* pada arcview akan dimodifikasi sehingga dapat digunakan sebagai dasar dalam pembangunan sistem. Perencanaan rute evakuasi merupakan komponen penting dalam mitigasi bencana yang mencari pengurangan kehilangan korban jiwa dan gangguan publik pada saat bencana alam atau serangan teroris (Shekar et al., 2012). Tujuan pembangunan perencanaan bencana adalah untuk memastikan pergerakan teraman dan tercepat untuk manusia agar terhindar dari ancaman atau bencana yang sedang terjadi (Stringfield, 1996 dalam Garcia-ojeda et al., 2013).

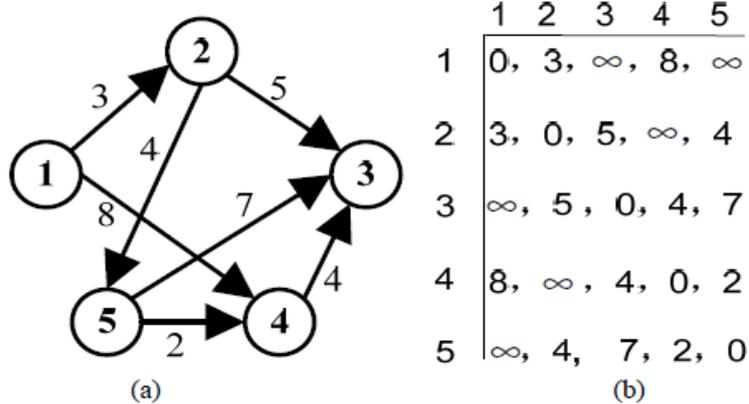
2.1 Algoritma Dijkstra

Algoritma Dijkstra adalah algoritma untuk menemukan jarak terpendek dari satu titik ke titik lain di graf berbobot, jarak yang ada diantara titik adalah bobot dari setiap tepi pada grafik itu. Algoritma Dijkstra dapat digunakan untuk menyelesaikan graph berarah, namun benar untuk menyelesaikan graph tak berarah (Munir, 2012). Salah satu komponen dari algoritma dijkstra adalah graf dan matriks ketetangaan. Graf merupakan pasangan himpunan $G = (V, E)$. Secara geometri graf digambarkan sebagai sekumpulan noktah(simpul) didalam bidang dwimatra yang dihubungkan dengan sekumpulan garis (sisi) (Munir, 2012). Dalam matriks ketetangaan terdapat komponen utama sebagai penyusunnya yaitu lintasan. Dua buah simpul dalam sebuah *graph* dinyatakan bertetangga apabila keduanya terhubung langsung dalam sebuah

sisi (Munir, 2012). Lintasan yang panjangnya n dari simpul awal v_0 kesimpul tujuan v_n didalam suatu graf adalah berselang-seling antar simpul dan sisi-sisi yang dapat dinotasikan sebagai;

$$e_1 = (v_0, v_1), e_2 = (v_1, v_2), \dots, e_n = (v_{n-1}, v_n).$$

Gambar 1. Graph berbobot dan matrix ketetangaan (Huang et al, 2013)



Untuk itu $G = (V, E)$. V merepresentasikan dari vertex. E merepresentasikan edge. $V = \{V_1, V_2, V_3, \dots, V_n\}$ termasuk pula n nodes didalamnya (**Gambar 1**). Matrix ketetangaan untuk *graph* berarah (Huang et al, 2013) $A = (a_{ij})_{n \times n}$ dijelaskan pada:

$$a_{ij} = \begin{cases} W_{ij}, & (V_i, V_j) \in E \\ \infty, & (V_i, V_j) \notin E \end{cases}, i, j = 1, 2, \dots, n \quad (1)$$

Dimana W_{ij} menyatakan bobot dari edge $\langle V_i, V_j \rangle$ sedangkan ∞ menyatakan bahwa tidak ada edge antara V_i dan V_j . L menyatakan gabungan titik akhir yang telah mendapatkan rute terpendek dari titik awal (Huang et al, 2013). $(V - L)$ menyatakan gabungan node yang belum dihitung dalam proses pencarian rute terpendek. Gunakan matrix ketetangaan W untuk menyimpan informasi jaringan. W_{ij} dinotasikan sebagai bobot dari /garis $\langle V_i, V_j \rangle$. Jika ternyata tidak ada keterkaitan antara V_i dan V_j , selanjutnya W_{ij} diubah menjadi ∞ . d_i menggambarkan sebagai bobot dari titik awal ke node V_i . Inisialisasikan titik awal dengan $d_s = 0$ dan $D_i = C_{si}$. $w(i, j)$ dimana mengandung bobot garis dari titik awal hingga titik akhir yang saling berhubungan dalam sebuah *graph*. Titik awal adalah i dan titik akhir adalah j .

$$V(G) - L = \{V_a, V_b, V_c, \dots, V_n\} - \{ \} = \{V_a, V_b, V_c, \dots, V_n\} \quad (2)$$

Untuk mengetahui node mana yang telah dipilih dalam proses penghitungan gunakan (2). Selama $V_n \notin L$ do:

- a) Pilih titik $V_k \in V - L$ dengan nilai $D(k)$ terkecil
- b) Untuk setiap $V_j \in V - L$, lakukan:

$$(D(j), (D(k) + W(k, j))) \quad (3)$$

$$\text{jika } (D(j) > (D(k) + W(k, j))) \quad (4)$$

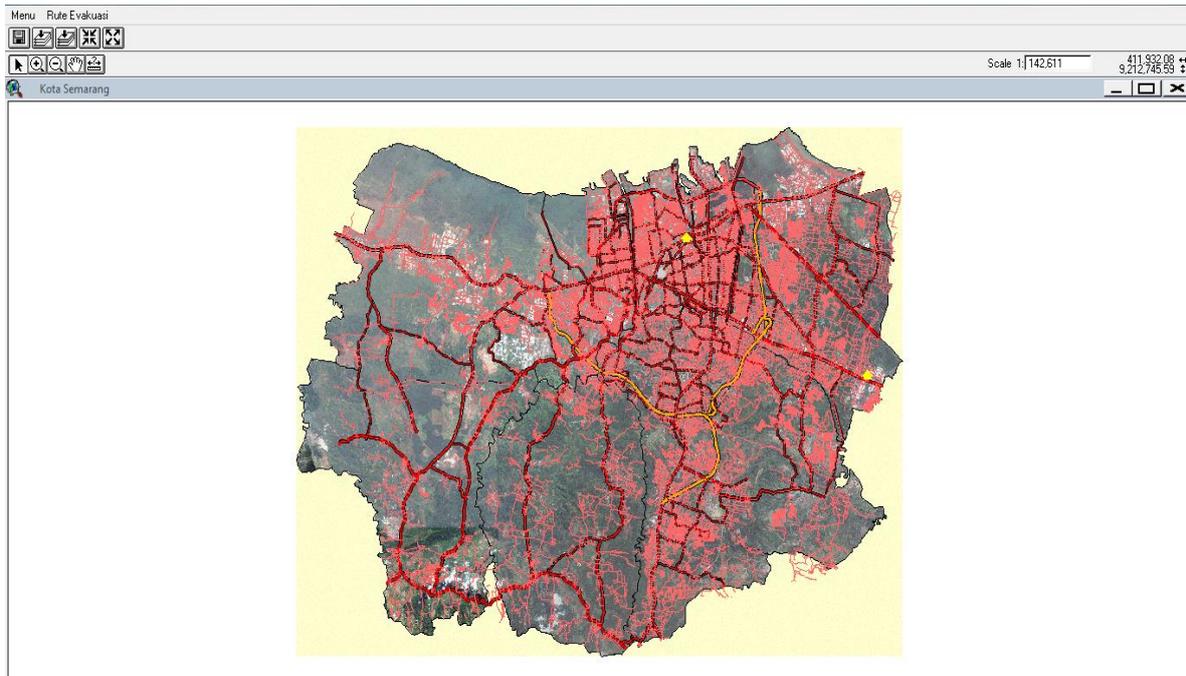
$$\text{pada iterasi selanjutnya ganti nilai bobot } D(j) \text{ dengan } (D(k) + W(k, j)) \quad (5)$$

2.2 Persiapan Data

a) Data Jalan

Data jalan merupakan faktor yang penting dalam proses pencarian rute. Data jalan (*vector shapefile*) yang berbentuk peta harus mengakomodasi kebutuhan mengenai panjang jalan, nama jalan. panjang jalan akan digunakan sebagai bobot saat pencarian rute dilakukan Sedangkan, nama jalan digunakan untuk menghasilkan panduan rute jalan (lihat **Gambar 2**).

Gambar 2. Peta Jalan Kota Semarang dalam SIG (Pemerintah Kota Semarang, 2010)

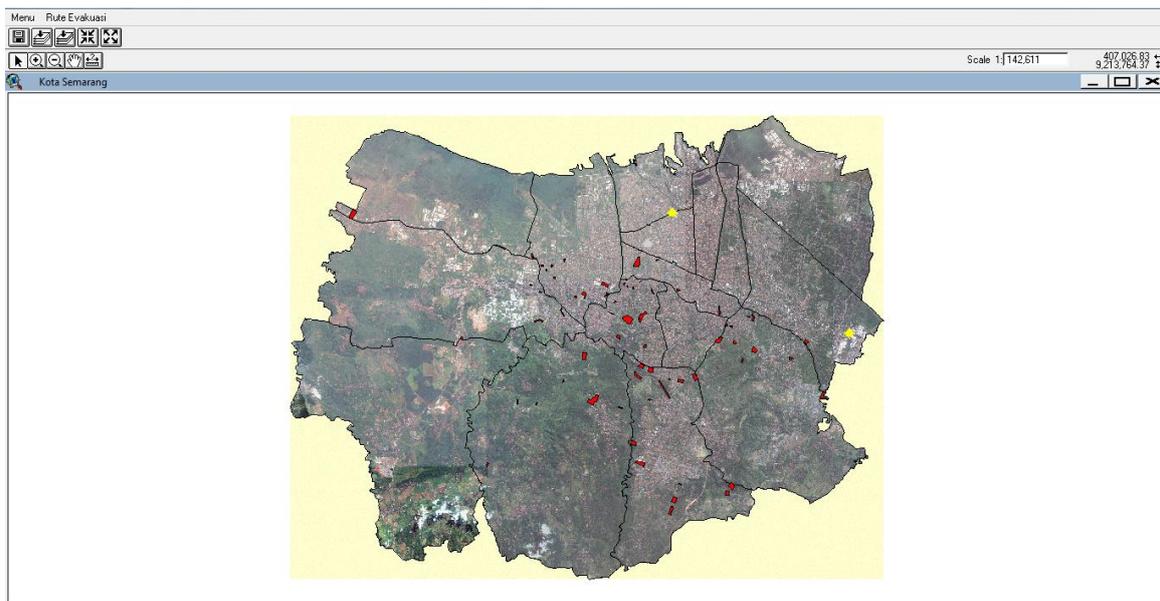


b) Data lokasi bencana

Berupa peta lokasi bencana yang berada pada Kelurahan yang ada di Kota Semarang. Peta ini awalnya berupa gambar yang kemudian didigitasi kedalam peta Kota Semarang. Peta yang akan didigitasi bersumber dari peta Kelurahan di Kota Semarang yang sebelumnya telah disusun oleh BPBD (Badan Penanggulangan Bencana Daerah) Kota Semarang. Pendigitasian tersebut bertujuan untuk menambahkan lokasi bencana tanah longsor yang ada di Kota Semarang. Dalam proses digitasi peta, kebutuhan data dasar berupa peta Kelurahan dan peta Kota Semarang.

Peta lokasi bencana dalam bentuk cetak yang berasal dari BPBD kota Semarang digunakan sebagai acuan dalam memasukan lokasi bencana khususnya referensi koordinat yang digunakan sebagai acuan. Alat yang digunakan dalam proses digitasi menggunakan *ArcView*. Proses digitasi dimulai dengan mengaktifkan peta Kota Semarang atau peta Kelurahan. Langkah selanjutnya yaitu mengaktifkan *editing tool polygon* yang terdapat ada *ArcView*. Mengacu pada data peta BPBD lokasi bencana longsor yang terdapat pada peta cetak dimasukkan kedalam peta Kota Semarang dan peta data lokasi bencana BPBD. Hasil digitasi peta dapat dilihat pada **Gambar 3**.

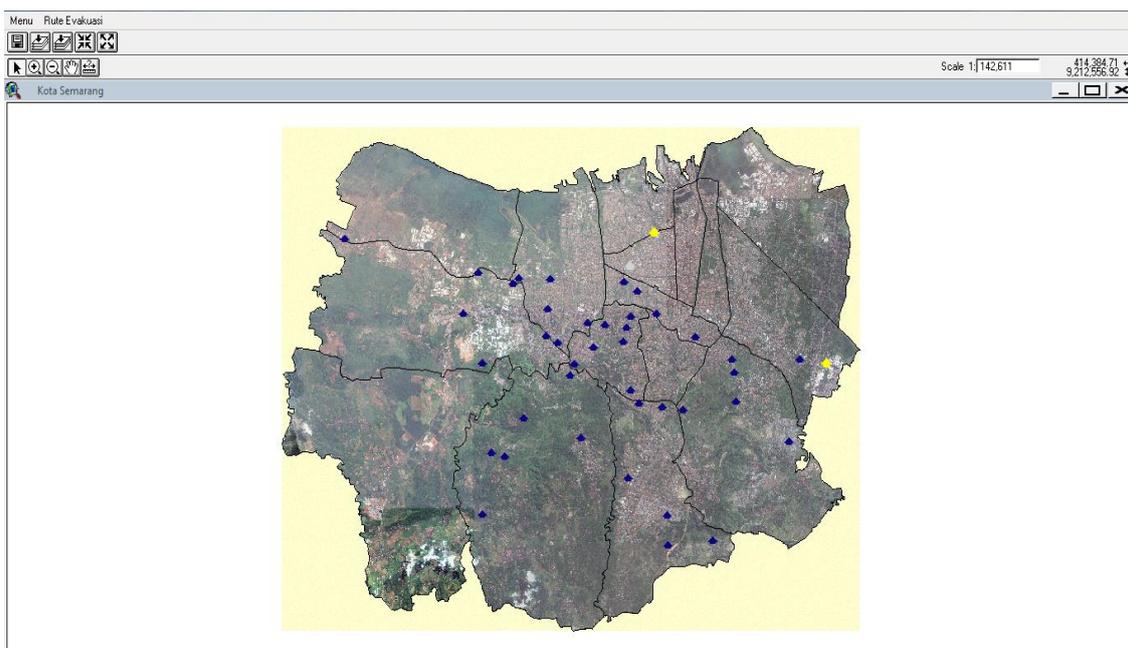
Gambar 3. Peta Bencana Tanah Longsor Kota Semarang dalam SIG
(Pemerintah Kota Semarang , 2015)



c) Data lokasi shelter.

Berisi informasi mengenai letak shelter bencana yang ada di Kota Semarang. Data mengenai lokasi shelter mengacu pada data peta BPBD Kota Semarang. Lokasi shelter bencana merupakan kantor kelurahan yang ada pada kota Semarang. Pemasukkan data dilakukan dengan cara digitasi yang sama dengan digitasi peta lokasi bencana, namun yang membedakan dalam digitasi peta lokasi bencana dan lokasi shelter adalah tipe data yaitu tipe peta *polygon* sedangkan peta lokasi shelter menggunakan tipe peta titik seperti pada **Gambar 4**.

Gambar 4. Peta Shelter Bencana Kota Semarang dalam SIG (Digitasi Peta, 2015)



2.3 Network Analyst

Pendekatan *network analyst* memiliki potensi yang kuat untuk menyelesaikan masalah transportasi dan masalah jalur (Akay et al, 2011). Node merepresentasikan simpangan jalan dan setiap *edge* pada *graph* berkaitan dengan segmen jalan diantara dua simpangan (Patel dan Baggar, 2014). Hampir semua tipe jaringan memiliki beberapa ciri yang mirip (Prahasta, 2004), diantaranya adalah:

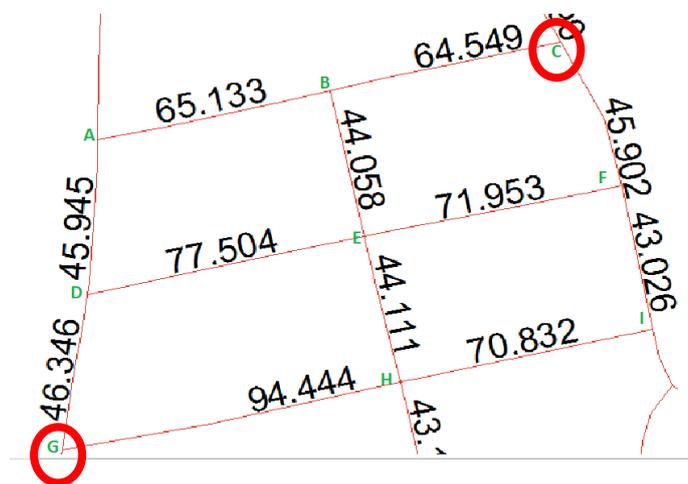
- Memiliki fenomena dimana terdapat objek atau *resource* yang bergerak di dalam jaringan yang bersangkutan.
- Memiliki fenomena dimana perpindahan dari lokasi awal ke lokasi tujuan suatu objek atau *resource* di dalam jaringan memerlukan keterhubungan (*connected path*) antara lokasi awal dan tujuan.

Dengan menggunakan *Network analyst* pengguna memiliki kemampuan untuk (Karadimas et al, 2007.)

- Menemukan rute perjalanan yang efisien.
- Menentukan jarak terdekat berdasarkan fasilitas atau kendaraan.
- Menghasilkan arah perjalanan.

Contoh titik yang akan digunakan saat pengujian validasi sistem, diawali dari titik C diakhiri pada titik G dapat dilihat pada **Gambar 5**.

Gambar 5. Informasi Panjang Titik Pengujian (Bappeda Kota Semarang, 2015)



2.4 Avenue Script

Avenue adalah bahasa computer yang digunakan untuk mengubah aplikasi ArcView. Dua jenis avenue script yang perlu dibuat adalah Antarmuka dan *system script*. Setiap kumpulan script didesain untuk mengontrol setiap proses kerja melalui setiap tombol antarmuka. Sedangkan *system script* adalah *script* yang digunakan untuk memanipulasi dengan sistem operasi ekstensi (Yue, 2003). Contoh potongan avenue script yang an digunakan saat pembuatan sistem adalah sebagai berikut:

```
theView = av.GetActiveDoc
for each t in theView.GetActiveThemes
  if (NetDef.CanMakeFromTheme(t)) then
    net = av.Run("Network.GetNetwork", (Dr. Wen Long Yue))
    if (net = nil) then
      exit ' (User has been informed of problems in Network.GetNetwork.)
    end
    defname = av.GetProject.MakeFileName("rute", "shp")
    resultFTab = ShortestPathWin.MakeNewResultFTab(net, defname)
    if (resultFTab.HasError) then
      MsgBox.Error("Error creating new FTab.", "Network.SPOpen")
    return(nil)
```

```

end
resultFTab.SetEditable(false)
resultTheme = FTheme.Make(resultFTab)
theLineSym = BasicPen.Make
theSymList = SymbolList.FromList({theLineSym})
theSymList.RandomColors
theLineSym.SetSize(2)
resultTheme.GetLegend.GetSymbols.Set(0,theLineSym)
resultTheme.SetVisible(True)
resultTheme.SetName(resultTheme.GetName.Substitute(".shp", ""))
theView.AddTheme(resultTheme)
win = ShortestPathWin.Make(theView, net, resultTheme, t)
win.Open
resultTheme.SetActive(true)
ext = NetworkWinSrc.Make(win)
resultTheme.SetExtension(ext)
av.GetProject.SetModified(True)
end
end
    
```

3. HASIL DAN PEMBAHASAN

3.1 Hasil Analisis Penghitungan manual

Untuk mengetahui bobot dan hubungan dalam graph yang digunakan sebagai validasi sistem dapat dilihat pada **Tabel 1**. Dalam tabel tersebut, apabila salah satu titik tidak berkaitan dengan titik lainnya akan dinotasikan dengan ∞ .

Tabel 1. Tabel $w(i,j)$ (Hasil Analisis, 2015)

W_{ij}	V_a	V_b	V_c	V_d	V_e	V_f	V_g	V_h	V_i
V_a	0	65.13	∞	45.94	∞	∞	∞	∞	∞
V_b	65.13	0	64.54	∞	44.05	∞	∞	∞	∞
V_c	∞	64.54	0	∞	∞	45.90	∞	∞	∞
V_d	45.94	∞	∞	0	77.50	∞	46.34	∞	∞
V_e	∞	44.05	∞	77.50	0	71.95	∞	44.11	∞
V_f	∞	∞	45.90	∞	71.95	0	∞	∞	43.02
V_g	∞	∞	∞	46.34	∞	∞	0	94.44	∞
V_h	∞	∞	∞	∞	44.11	∞	94.44	0	70.83
V_i	∞	∞	∞	∞	∞	43.02	∞	70.83	0

Setelah diketahui bobot dari masing-masing garis, selanjutnya tentukan nilai masing-masing garis dari titik awal yaitu titik C hingga titik G yang dinotasikan sebagai $D(k)$ pada **Tabel 2**.

Tabel 2. Tabel $D(k)$ (Hasil Analisis, 2015)

$D(a) = W(c,a) = \infty$	$D(f) = W(c,f) = 45.902$
$D(b) = W(c,b) = 64.54$	$D(g) = W(c,g) = \infty$
$D(d) = W(c,d) = \infty$	$D(h) = W(c,h) = \infty$
$D(e) = W(c,e) = \infty$	$D(i) = W(c,i) = \infty$

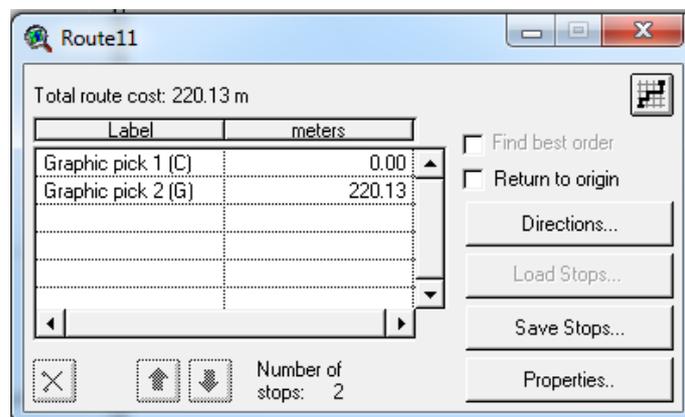
Dari **Tabel 3** dapat diketahui bahwa bobot $D(k)$ terendah adalah $D(f)$ dengan bobot 45.902 sehingga dapat disimpulkan bahwa $V(k) = V_f$.

Tabel 3. Hasil Analisis Perhitungan Manual (Hasil Analisis, 2015)

Indes k D(k) minimum	L	V-L	D(a)	D(b)	D(d)	D(e)	D(f)	D(g)	D(h)	D(i)
-	0	Va,Vb, Vd,Ve, Vf,Vg, Vh,Vi	W (c,a) =∞	W(c,b) =64.549	W (c,d) =∞	W (c,e) =∞	W(c,f) =45.902	W (c,g) =∞	W (c,h)= ∞	W (c,i) =∞
D(K)f	Vf	Va,Vb, Vd,Ve, Vg, Vh,Vi	∞	64.549	∞	117.85 7	45.902 tetap	∞	~	88.928
D(K)b	Vf,Vb	Va,Vd, Ve,Vg, Vh,Vi	129.682	64.549 Tetap	∞	108.60 7	45.902 tetap	∞	∞	88.928
D(K)i	Vf,Vb, Vi	Va,Vd, Ve,Vg, Vh	129.682	64.549 Tetap	∞	108.60 7	45.902 tetap	∞	159.76	88.928 tetap
D(K)e	Vf,Vb, Vi,Ve	Va,Vd, Vg,Vh	129.682	64.549 Tetap	186.111	108.60 7 tetap	45.902 tetap	∞	152.718	88.928 tetap
D(K)a	Vf,Vb, Vi,Ve, Va	Vd,Vg, Vh	129.682 Tetap	64.549 Tetap	175.573	108.60 7 tetap	45.902 tetap	∞	152.718	88.928 tetap
D(K)h	Vf,Vb, Vi,Ve, Va,Vh	Vd,Vg	129.682 Tetap	64.549 Tetap	175.573	108.60 7 tetap	45.902 tetap	247.16 2	152.718 tetap	88.928 tetap
D(K)d	Vf,Vb, Vi,Ve, Va,Vh, Vd	Vg	129.682 Tetap	64.549 Tetap	175.573 Tetap	108.60 7 tetap	45.902 tetap	221.91 8	152.718 tetap	88.928 tetap
D(K)g	Vf,Vb, Vi,Ve, Va,Vh, Vd,Vg		129.682 Tetap	64.549 Tetap	175.573 tetap	108.60 7 tetap	45.902 tetap	221.91 8	152.718 tetap	88.928 tetap

Hasil analisis dengan penghitungan manual menunjukkan hasil 221.918 yang kemudian dibulatkan menjadi 222. Selanjutnya hasil analisis dengan algoritma Dijkstra secara manual akan dibandingkan dengan hasil *running* dari kostumasi *network analyst* pada *Arcview GIS* (**Gambar 6**).

Gambar 6. Hasil Pehitungan dengan *Network analyst GIS* (Hasil Analisis, 2015)



Hasil penghitungan jarak dengan *Network analyst* menunjukkan hasil 220,13 yang dibulatkan menjadi 220. Penghitungan manual dilakukan dengan mengambil titik acak yang terdapat pada peta jalan Kota Semarang. Proses pencarian rute dengan menggunakan data jalan Kota Semarang dilakukan dengan mengambil titik acak untuk melakukan penghitungan manual, titik awal dimulai dari titik C dan diakhiri pada titik G (Lihat **Tabel 4**).

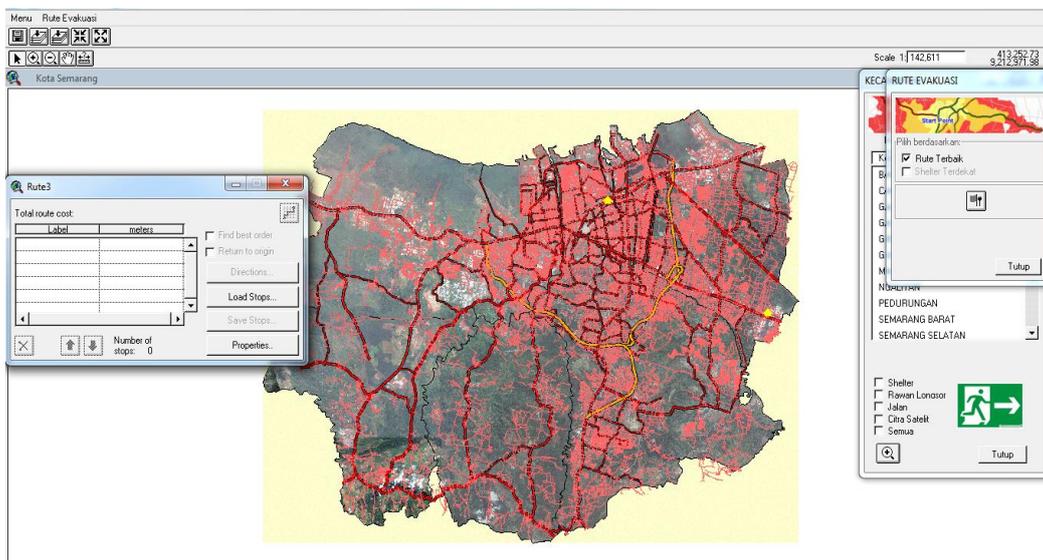
Tabel 4. Perbandingan Jarak (Hasil Analisis, 2015)

Perhitungan manual	Perhitungan <i>Network analyst</i>
221,918 dibulatkan menjadi 222	220,13 dibulatkan menjadi 220.

3.2 Hasil Pembangunan system dengan *avenue script*

Penggunaan *avenue script* yang ada pada Arcview dapat diterapkan pada pembangunan system, sehingga ekstensi *network analyst* dapat digunakan pada **gambar 7**. Hasil dari penggunaan *avenue script* adalah sistem informasi geografis yang dapat digunakan untuk mencai rute tercepat. Dalam sistem ini pengguna dapat mengakses fitur *network analyst* untuk mencari rute tercepat. Rute yang telah dihasilkan dapat dicetak sebagai penunjuk jalan saat proses evakuasi berlangsung. Dalam sistem ini tersedia titik lokasi bencana dan titik lokasi shelter yang dapat digunakan sebagai acuan dalam memilih titik awal dan akhir pencarian rute. Perubahan kondisi jalan yang sewaktu-waktu menjadi salah satu hambatan, oleh karena itu, pengguna diberi kebebasan dalam menentukan titik awal dan titik akhir dalam pencarian rute. Pemilihan tersebut dapat dilakukan tanpa mengacu pada titik lokasi bencana dan lokasi shelter.

Gambar 7. Kostumasi Sistem dengan Avenue Script untuk *Network analyst* (Hasil Pembangunan Sistem, 2015)



4. KESIMPULAN

Penggunaan algoritma dijkstra dalam ekstensi *network analyst* dapat digunakan untuk mencari rute terpendek, perbandingan hasil hitung algoritma dijkstra secara manual dengan hasil *network analyst* hanya terpaut 2 meter, sehingga *network analyst* cukup akurat jika digunakan pada kondisi nyata. *Avenue script* yang dapat membantu modifikasi sistem sangat membantu pengguna untuk membangun sistem yang sesuai dengan kebutuhan, khususnya dalam hal mitigasi bencana. Dengan adanya sistem ini proses pencarian rute evakuasi bencana longsor menjadi lebih cepat daripada menggunakan penghitungan manual. Sistem ini dapat diterapkan selama kebutuhan data tercukupi dan diperbarui apabila terjadi hal yang berkaitan dengan kondisi jalan, lokasi bencana, dan lokasi *shelter* untuk evakuasi.

5. DAFTAR PUSTAKA

- Akay, A. E., Wing, M. G., Sivrikaya, F., & Sakar, D. (2011). *A GIS-based decision support system for determining the shortest and safest route to forest fires: a case study in Mediterranean Region of Turkey*. Springer Science Business Media B.V.
- Fuhao, Z., & Jiping, L. (2009). *An algorithm of shortest path based on Dijkstra for huge data*. 6th International Conference on Fuzzy Systems and Knowledge Discovery.

- Gracia-ojeda, J.C. et al. (2013). Building-evacuation-route planning via time-expanded process-network synthesis. *Fire Safety Journal*, 61, 338–347.
- Hongchun, H., Yinghua, S. (2012). Research on the Construction and Improvement to the Emergency Response Mechanism in Public Emergencies. *Information Engineering and Applications*, 154, 458-459.
- Huang, Y., Yi, Q., & Shi, M. (2013) *An Improved Dijkstra Shortest Path Algorithm*. Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE).
- Kai, N., Yao-ting, Z., Yue-peng, M. (2014). Shortest Path Analysis Based on Dijkstra's Algorithm in Emergency Response System. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, 12(5), 376-482.
- Karadimas, V. et al. (2007). *Municipal Waste Collection of Large Items Optimized with ARC GIS Network analyst*. Proceedings 21st European Conference on Modelling and Simulation.
- Kementerian Dalam Negeri. (2006). *Pedoman Umum Mitigasi Bencana*.
- Munir, R. (2012). *Matematika Diskrit*. Bandung: Infomatika.
- Patel, V., & Baggar, C. (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 (IJPTT)*, 5.
- Prahasta, E. (2004). *Sistem Informasi Geografi: Tools dan Plug-Ins*. Bandung: Infomatika.
- Shekar, S. et al. (2012). Experiences With Evacuation Route Planning Algorithms. *International Journal of Geographical Information Science*, 26, 2253–2265.
- Siang, J. J. (2004). *Matematika Diskrit dan Aplikasinya pada Ilmu Komputer*. Yogyakarta: Andi.
- Yue, W. L., & Monthira. (2003). The Development Of Arcview Extension Tool For Road Project Cost And Benefit Analysis; Case Study Of Bangkok, Thailand. *Journal of the Eastern Asia Society for Transportation Studies*, 5, 2972-2987.