JURNAL MATEMATIKA DAN KOMPUTER
Vol. 6. No. 2, 59 - 70, Agustus 2003, ISSN : 1410-8518

MASALAH RUTE TERPENDEK PADA JARINGAN JALAN MENGGUNAKAN LAMPU LALU-LINTAS

Studi Kasus: Rute Perjalanan Ngesrep – Simpang Lima

Eko Budi P dan Sunarsih

Jurusan Matematika Fakultas MIPA Universitas Diponegoro

Abstrak

Permasalahan rute terpendek pada jaringan jalan yang menggunakan lampu lalu-lintas bertujuan untuk menentukan rute yang menghubungkan titik asal *s* dan titik tujuan *j*, yang mempunyai waktu perjalanan total minimum. Lampu lalu-lintas pada jaringan jalan ini diasumsikan hanya terdiri dari dua fase yaitu merah dan hijau, dengan periode waktu siklus adalah konstan. Permasalahan ini dapat direpresentasikan kedalam graph berarah, dengan waktu perjalanan untuk tiap-tiap jalan adalah bobot arc, dan waktu tunggu pada persimpangan jalan merupakan bobot titik. Waktu perjalanan dari titik asal ke titik tujuan dipengaruhi oleh dua faktor yaitu waktu perjalanan untuk tiap jalan dan waktu tunggu pada persimpangan jalan, dengan lamanya waktu tunggu diatur oleh lampu lalu-lintas. Untuk menyelesaikan permasalahan rute terpendek ini digunakan algoritma Ford Moore Bellman yang telah dimodifikasi. Pada studi kasus: rute perjalanan Ngesrep – Simpang Lima, dengan menggunakan algoritma ini diperoleh waktu perjalanan minimum dari rute tersebut adalah 10 menit 59 detik, melalui rute Setya Budi → Teuku Umar → Sultan Agung → Diponegoro → Pahlawan → Simpang Lima, dengan beberapa asumsi yaitu: kecepatan kendaraan ketika melewati rute ini adalah konstan yaitu 40 km/jam, tidak terdapat kemacetan pada rute tersebut dan kendaraan hanya berhenti di persimpangan jalan karena lampu lalu-lintas.

Kata Kunci : rute terpendek, jalan, lampu lalu-lintas, graph berarah.

1. PENDAHULUAN

Jaringan jalan menggunakan lampu lalu-lintas adalah jaringan jalan yang mempunyai lampu lalu-lintas disetiap simpang jalan. Lampu lalu-lintas ini biasanya terdiri atas tiga warna lampu yaitu merah, kuning dan hijau. Tanda merah berarti berhenti, kuning dan hijau berarti berjalan. Tanda ini berubah secara teratur. Setiap pengulangan urutan tanda lampu secara keseluruhan disebut satu siklus sinyal dan lamanya disebut waktu siklus.

Selain menguntungkan karena dapat memperlancar lalu-lintas kendaraan, penggunaan lampu lalu-lintas juga mempunyai kerugian yaitu menambah waktu perjalanan karena menunggu pada persimpangan jalan. Lamanya seseorang menunggu pada persimpangan jalan didefinisikan sebagai waktu tunggu yaitu lamanya menunggu sebelum lampu hijau menyala.

Permasalahan rute terpendek pada jaringan jalan menggunakan lampu lalu-lintas dapat dimodelkan dalam bentuk jaringan yang berupa graph berarah $G(N, A)$. Tempat atau persimpangan jalan diwakili oleh suatu titik N sedangkan jalan yang dilalui direpresentasikan dalam bentuk garis berarah atau arc A.

2. TINJAUAN PUSTAKA

2.1 Jaringan jalan menggunakan lampu lalu-lintas

Jaringan jalan menggunakan lampu lalu-lintas adalah jaringan jalan yang mempunyai lampu lalu-lintas di persimpangan jalan. Jaringan ini dapat direpresentasikan kedalam graph berarah, dengan persimpangan jalan diwakili oleh *titik*, sedangkan jalan direpresentasikan dalam *arc*.

Definisi 2.1 Waktu perjalanan yang dinotasikan dengan $d_{ij}(t)$ adalah waktu yang diperlukan untuk melakukan perjalanan pada jalan yang dinotasikan dengan arc (i, j).

Definisi 2.2 Waktu tunggu yang dinotasikan dengan $w_i(t)$ adalah lamanya kendaraan menunggu pada persimpangan jalan yang dinotasikan dengan titik i, sebelum melanjutkan perjalanan.

Lamanya waktu tunggu kendaraan pada persimpangan jalan ditentukan oleh lampu lalu-lintas. Jika lampu lalu-lintas berwarna merah ketika kendaraan akan melewati persimpangan jalan, maka waktu tunggu adalah lamanya kendaraan menunggu pada persimpangan jalan sebelum lampu lalu-lintas berwarna hijau menyala. Sedangkan apabila lampu lalu-lintas berwarna hijau, maka waktu tunggu pada kondisi ini nol.

2.2.1 Lampu lalu-lintas pada Persimpangan Jalan

Lampu lalu-lintas terdiri dari tiga warna yaitu merah, kuning, dan hijau. Warna merah berarti semua kendaraan harus berhenti, kuning dan hijau berarti semua kendaraan berjalan. Lampu lalu-lintas diasumsikan hanya terdiri dari dua warna yaitu merah dan hijau, karena kuning disamakan dengan hijau yaitu jalan. Setiap pengulangan urutan tanda lampu lalu-lintas secara keseluruhan disebut satu siklus sinyal dan lamanya disebut waktu siklus.

Definisi 2.3 Misalkan $i \in N$ adalah suatu titik yang diatur oleh lampu lalu-lintas, $a = (h,i)$ adalah arc masuk pada titik i dan $b = (i, j)$ merupakan arc keluar dari titik i, maka pasangan (a, b) dinamakan pasangan fisibel (*fisibel*).

Definisi 2.4 Fase hijau (*green*) yang dinotasikan dengan $g(a,b)$ adalah lamanya lampu lalu-lintas menyala berwarna hijau tiap periode atau waktu siklus.

Definisi 2.5 Fase merah (*red*) yang dinotasikan dengan $r(a,b)$ adalah lamanya lampu lalu-lintas berwarna merah menyala tiap periode atau waktu siklus.

Definisi 2.6 Waktu horizon yang dinotasikan dengan $t(h)$ adalah waktu lampu lalu-lintas mulai menyala.

Definisi 2.7 Nilai fase yang dinotasikan dengan $s(a,b)$ adalah selisih antara mulainya waktu horizon $t(h)$ dengan fase hijau pertama sesudah $t(h)$.

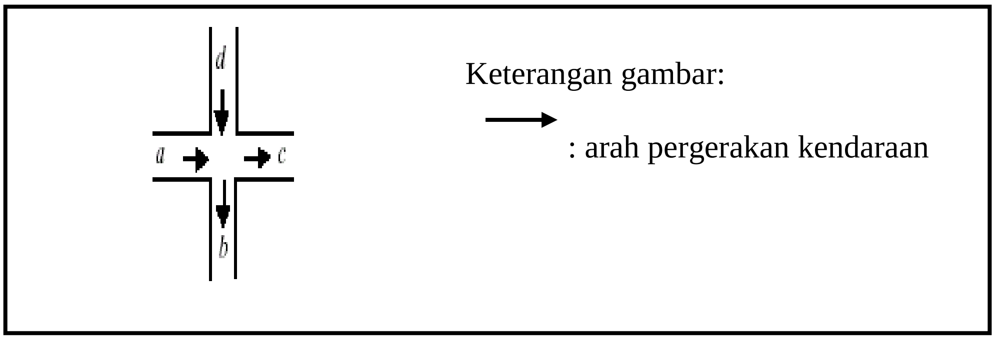
Jika dalam waktu horizon $t(h)$, (a, b) adalah fase merah maka $s(a,b) \le r(a,b)$; dan sebaliknya jika (a, b) dalam fase hijau maka $s(a,b) > r(a,b)$.

Pada kedua kondisi tersebut, fase hijau pertama mulai pada saat $t(h) + s(a,b)$.

Definisi 2.8 Waktu relatif dari arc *a* ke arc *b* yang didefinisikan dengan "triplet" $[g(a,b), r(a,b), s(a,b)]$ merupakan barisan fase hijau dan fase merah yang diulang.

Untuk ilustrasi perhatikan contoh berikut ini :

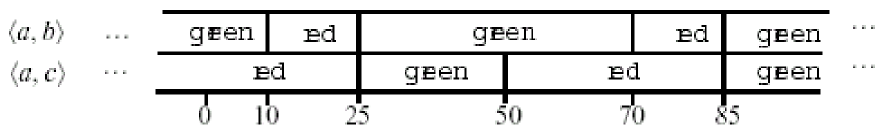
Sebuah persimpangan jalan terdiri atas 4 jalan, yaitu jalan a, b, c dan jalan d. Pada persimpangan tersebut terdapat 2 lampu lalu-lintas yang berada pada ujung jalan a dan d, dengan masing-masing lampu lalu-lintas mempunyai pengaturan yang berbeda-beda. Lampu lalu-lintas pada ujung jalan a mengatur pergerakan kendaraan dari jalan a menuju jalan b dan c. Begitu pula lampu lalu-lintas pada ujung jalan d, mengatur arus kendaraan dari jalan d menuju ke jalan b dan c. Fenomena tersebut dapat digambarkan kedalam gambar berikut ini :



Gambar 3.1 Graph persimpangan jalan

Jika diketahui durasi lampu lalu-lintas pada persimpangan tersebut ditunjukkan oleh tabel berikut ini:

Tabel 3.1 Durasi lampu lalu-lintas (dalam detik)



Dari tabel tersebut, dapat dijelaskan bahwa pada pasangan fisibel (*a*, *b*), fase hijau adalah 70 – 25 = 45, fase merah sebesar 85 – 70 = 15 detik, waktu horizon *t(h)* = 0, nilai fase yaitu selisih antara fase hijau pertama dengan waktu *horizon* 0 adalah 25, sehingga *s(a,b)* = 25 – 0 = 25.

Jadi "triplet" dari (*a*, *b*) adalah $[g(a,b), r(a,b), s(a,b)]$ = [45, 15, 25].

2.2.2 Algoritma Waktu Tunggu

Untuk memperoleh waktu tunggu dalam *titik* pada jaringan jalan dengan lampu lalu-lintas digunakan algoritma berikut ini:

Misal $Q = (\Pi_{sj}(t) - t_k) \bmod \pi(a,b)$ (3.1)

dimana $\pi(a,b) = g(a,b) + r(a,b)$ (3.2)

adalah periode lampu lalu-lintas, dengan $\Pi_{sj}(t)$ adalah waktu perjalanan dari *titik* *s* ke *j*, dan t_k merupakan waktu keberangkatan dari *titik s*. Algoritma ini dapat diterapkan pada kasus berikut ini:

Kasus 1 : Pada saat waktu *horizon* $t(h)$, lampu lalu-lintas berwarna merah, sehingga dari Definisi 3.7 diperoleh $s(a,b) \le r(a,b)$. Waktu tunggu kendaraan pada *titik j* jika terjadi kondisi tersebut dapat dicari dengan menggunakan rumus di bawah ini:

$$\mathrm{w}(a,b,t) = \begin{cases} s(a,b) - Q, & \text{jika } 0 \le Q < \mathrm{s(a,b)}; \\ 0, & \text{jika } s(a,b) \le Q < g(a,b) + s(a,b); \\ \pi(a,b) + s(a,b) - Q, & \text{jika } g(a,b) + s(a,) \le Q < \pi(\mathrm{a,b}). \end{cases} \quad (3.3)$$

Kasus 2 : Pada saat waktu *horizon* $t(h)$, lampu lalu-lintas berwarna hijau, sehingga menurut Definisi 3.7 didapat $s(a,b) > r(a,b)$. Untuk kasus ini, waktu tunggu kendaraan pada *titik j* dapat diperoleh dengan menggunakan rumus berikut ini:

$$w(a,b,t) = \begin{cases} 0, & \text{jika } 0 \le Q < \mathrm{g(a,b) + s(a,b)} - \pi(\mathrm{a,b}); \\ s(a,b) - Q, & \text{jika } g(a,b) + s(a,b) - \pi(a,b) \le Q < s(a,b); \\ 0, & \text{jika } s(a,b) \le Q < \pi(a,b) \end{cases} \quad (3.4)$$

Contoh :

Perhatikan pergerakan kendaraan yang terdapat pada persimpangan *j*, yang ditunjukkan oleh Gambar 3.1 . Pergerakan tersebut diatur oleh lampu lalu-lintas yang terdapat pada ujung jalan a, dengan durasi lampu diperlihatkan pada Tabel 3.1.

Misalkan diketahui waktu perjalanan kendaraan sebelum sampai pada persimpangan *j* yaitu waktu perjalanan dari *titik* awal *s* ke *titik j* ($\Pi_{sj}(t)$) adalah selama 1565 detik, dan waktu keberangkatan dari *titik s* yang dinotasikan dengan t_k adalah nol. Waktu tunggu kendaraan di persimpangan tersebut, jika kendaraan hendak melanjutkan perjalanan dari jalan a ke jalan b dapat diperoleh dengan menggunakan algoritma sebagai berikut:

1. Menghitung $\pi(a,b)$, dengan menggunakan persamaan (3.2), sehingga diperoleh : $\pi(a,b) = (g(a,b) + r(a,b) = 45 + 15 = 60.$
2. Mencari nilai *Q*, dengan menggunakan rumus pada persamaan (3.1), sehingga diperoleh :

   $Q =$ $(\Pi_{sj}(t) - t_k) \bmod \pi(a,b)$ $= 5$

   $Q = 5$
3. Diketahui Q = 5, $s(a,b)$ = 25, $\pi(a,b) = 60$, $g(a,b)$ = 45 dan $r(a,b) = 15$.

   Dari data tersebut di atas diperoleh suatu kondisi sebagai berikut :

   ✓ $s(a,b) > r(a,b)$

   ✓ $g(a,b) + s(a,b) - \pi(a,b) = 10$

   ✓ $Q < g(a,b) + s(a,b) - \pi(a,b)$

   ✓ $0 \le Q < g(a,b) + s(a,b) - \pi(a,b)$

   Dengan memperhatikan kondisi tersebut, dan dilakukan "cross check" dengan 2 kasus dari algoritma waktu tunggu pada *titik*, diketahui bahwa permasalahan waktu tunggu ini merupakan contoh dari kasus kedua.
4. Untuk mencari waktu tunggu pada *titik j*, lihat algoritma untuk mencari waktu tunggu pada kasus dua. Dengan menggunakan persamaan (3.4)

JURNAL MATEMATIKA DAN KOMPUTER
Vol. 6. No. 2, 59 - 70, Agustus 2003, ISSN : 1410-8518

diperoleh waktu tunggu kendaraan apabila kendaraan bergerak dari jalan a menuju ke b yaitu sebesar $w(a,b,\bar{t}) = 0$.

Artinya kendaraan tersebut tidak perlu menunggu pada persimpangan jalan *j*, dan bisa terus melanjutkan perjalanan.

2.2.3 Algoritma Ford Moore Bellman

Algoritma ini digunakan untuk mencari rute terpendek pada jaringan jalan. Algoritma Ford Moore Bellman ditemukan oleh Ford (1956), Moore (1957), dan Bellman (1958). Dasar dari algoritma ini adalah lintasan terpendek dari titik *s* ke titik *j* yang memuat paling banyak $\mathrm{k}+1$ garis berarah dapat diperoleh dari lintasan terpendek dari titik *s* ke titik *j* yang memuat paling banyak k garis berarah.

Dalam algoritma Ford Moore Bellman, lambang $L_k^{(k)}$ menyatakan bobot lintasan ($P_{sj}^{(k)}$) dari titik *s* ke titik *j* yang melalui paling banyak k buah garis berarah pada suatu graph $G(N,E,l)$.

Berikut ini diberikan teorema yang mendukung algoritma Ford Moore Bellman.

Teorema 3.1

Dalam suatu graph $G(N,E,l)$ yang memuat n titik dan $P_{sj}^{(k+1)}$ adalah lintasan terpendek dari *s* ke *j* yang memuat k+1 garis berarah, maka $L(P_{sj}^{(k+1)})$ dapat dicari dengan rumus:

$$L_j^{(k+1)} = L(P_{sj}^{(k+1)}) = \min[L_{si}^{(k)} + L_{ij}]$$

Bukti :

Diketahui $P_{sj}^{(k+1)}$ adalah lintasan terpendek dari *s* ke titik *j* yang memuat k+1 garis berarah dengan (*i, j*) adalah garis berarah yang terakhir. Ini berarti $P_{sj}^{(k+1)}$ dapat dianggap memiliki k buah garis berarah yang diikuti oleh garis berarah terakhir yaitu (*i, j*) sehingga

$$L_j^{(k+1)} = L(P_{sj}^{(k+1)}) = [L_{si}^{(k)} + L_{ij}]$$

Karena $P_{sj}^{(k+1)}$ adalah lintasan terpendek maka dipilih yang paling minimum sehingga $L_j^{(k+1)} = L(P_{sj}^{(k+1)}) = \min[L_{si}^{(k)} + L_{ij}]$

Teorema 3.2

Pada suatu graph $G(N, E, l)$ yang memuat n titik dan $P_{sj}^{(k)}$ adalah lintasan terpendek dari titik *s* ke titik *j* yang memuat paling banyak k buah garis berarah maka:

$$L_j^{(k+1)} = L(P_{sj}^{(k+1)}) = L_j^{(k)}$$

Bukti :

Dari Teorema 3.1 telah diperoleh rumus $L_j^{(k+1)} = \min[L_i^{(k)} + L_{ij}]$, karena diketahui bahwa $P_{sj}^{(k)}$ hanya memuat paling banyak k garis berarah, maka jika dianggap $P_{sj}^{(k)}$ memuat k+1 garis berarah, ini berarti lintasan terpendek $P_{sj}^{(k)}$ terdiri atas lintasan $P_{si}^{(k)}$ yang memuat k garis berarah dan diikuti garis (*i*, *j*) yang berbobot nol. Hal ini berarti bahwa kedua titik yaitu titik *i* dan *j* berhimpit, sehingga $P_{si}^{(k)} = P_{sj}^{(k)}$ atau dengan kata lain:

$$\begin{aligned} L_j^{(k+1)} &= \min[L_i^{(k)} + L_{ij}] \\ &= \min[L_i^{(k)} + 0] \\ &= \min[L_j^{(k)}] = L_j^{(k)} \end{aligned}$$

Langkah-langkah Algoritma Ford Moore Bellman

1. Langkah awal

   Diberikan :

   - $L_s = L_s^{(l)} = 0$, k = 1,2,...,n-1.

     Dengan $L_s = L_s^{(l)}$ adalah jarak dari *titik* awal *s* ke *titik s* itu sendiri.

   - $L_j^{(1)} = L_{(s,j)}^{(1)}$, *j* = 1,2,...,n-1.

     Jika *titik j adjacent* dari *titik s* maka:

     $L_j^{(1)} = L_{(s,j)}^{(1)}$, dengan $L_j^{(1)}$ merupakan jarak antara *titik* awal *s* ke *titik* tujuan *j*.

Sebaliknya jika *titik j* bukan *adjacent* dari *s* maka:

$L_{(s,j)}^{(1)} = \infty$.

Setelah semua $L_j^{(1)}$ diketahui, dilanjutkan ke langkah 2.

2. Langkah 2

   Menghitung $L_j^{(k+1)}$ dengan menggunakan rumus berikut ini :

   $L_j^{(k+1)}$ = min { $L_j^{(k)}, \min[L_i^{(k)} + L_{ij}]\}$

   untuk $k = 1$ dan $j = 1,2,..,n-1 \in N(G)$, dengan $i \neq j$.

3. Ulangi langkah 2, untuk $k = 2,3,...,n-1$.
4. Penghentian iterasi

   Jika diperoleh $L_j^{(k+1)} = L_j^{(k)}$ untuk semua $j = 1,2,3,...,n-1$, dengan syarat $k \leq n-1$, maka iterasi dihentikan. Jika belum kembali ke langkah 3.

   Sebaliknya, jika $L_j^{(k+1)} \neq L_j^{(k)}$, ketika *k = n-1* maka ini berarti jaringan memuat sirkuit negatif, dan iterasi dihentikan.

Sebelum algoritma ini digunakan untuk menyelesaikan permasalahan rute terpendek pada jaringan jalan yang menggunakan lampu lalu-lintas, algoritma Ford Moore Bellman ini terlebih dahulu dimodifikasi. Modifikasi dimaksudkan untuk mengganti bobot lintasan dari jarak menjadi waktu perjalanan *titik s* ke *j*.

Jika $\Pi_j(t)$ adalah waktu perjalanan tercepat atau minimum dari *titik* awal *s* ke *titik* tujuan *j* dengan waktu keberangkatan adalah t_k, maka $\Pi_j(t)$ dapat diperoleh dengan menggunakan fungsi berikut ini:

$\Pi_j(t) = \min_{j \in A(i)} \{\Pi_i(t) + D_{ij}(t)\}$

dengan $D_{ij}(t) = w_i(t) + d_{ij}(t)$

dimana $\Pi_i(t)$: waktu perjalanan dari *titik* asal *s* ke *titik i*.

$D_{ij}(t)$: waktu perjalanan total dari *titik i* ke *titik j*.

$d_{ij}(t)$: waktu perjalanan dalam *arc* (*i, j*).

$w_i(t)$: waktu tunggu pada *titik i*.

2.2.4 Model Matematika Waktu Perjalanan Minimum

Model matematika masalah rute terpendek pada jaringan jalan menggunakan lampu lalu-lintas adalah meminimalkan bobot lintasan yang menghubungkan *titik* awal dan *titik* tujuan, dengan bobot lintasan terdiri dari waktu perjalanan pada *arc* dan waktu tunggu pada *titik* yang dilalui.

Jika dituliskan kedalam persamaan matematika maka masalah waktu perjalanan minimum dalam jaringan jalan yang menggunakan lampu lalu-lintas dapat diformulasikan kedalam model matematis sebagai berikut:

$$\text{Min } \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}x_{ij}$$

Dengan kendala pada tiap-tiap *titik* sebagai berikut:

$$\sum_{keluar} x_{ij} - \sum_{masuk} x_{ij} = 1 \quad \text{untuk } \textit{node} \text{ sumber} \qquad (1)$$

$$\sum_{keluar} x_{ij} - \sum_{masuk} x_{ij} = 0 \quad \text{untuk } \textit{node} \text{ antara} \qquad (2)$$

$$\sum_{keluar} x_{ij} - \sum_{masuk} x_{ij} = -1 \quad \text{untuk } \textit{node} \text{ tujuan} \qquad (3)$$

dengan,

i : *titik* asal

j : *titik* tujuan

$w_{ij}(t)$: waktu perjalanan pada *arc* (i, j)

$w_{ii}(t)$: waktu tunggu pada *titik i*

3. METODOLOGI PENELITIAN

3.1 Tahap Pertama

Pada tahap pertama (formulasi masalah), kegiatan penelitian dilakukan dengan mengidentifikasi masalah rute terpendek pada jaringan jalan yang menghubungkan Ngesrep dan Simpang Lima.

3.2 Tahap Kedua

Pada tahap kedua yang meliputi pengumpulan serta analisa data yang diperoleh dapat dijelaskan sebagai berikut :

3.2.1 Pengambilan Data

Data-data yang diperlukan untuk penelitian ini diperoleh dengan 2 cara :

1. Data Primer

   Data primer diperoleh dengan cara survei dilapangan. Data-data primer yang dikumpulkan meliputi "setting" lampu lalu-lintas yaitu (berupa lamanya lampu lalu-lintas menyala berwarna merah, kuning, dan hijau) untuk setiap persimpangan jalan yang menghubungkan Ngesrep – Simpanglima.

2. Data Sekunder

   Data sekunder yang berupa lokasi penempatan lampu lalu-lintas diperoleh dari instansi terkait yaitu Dinas Perhubungan Kota Semarang, sedangkan data gambar atau peta jaringan jalan yang menghubungkan Ngesrep – Simpanglima dapat diperoleh dari peta Semarang.

3.2.2 Pengolahan dan Analisa Data

3.2.2.1 Pengolahan Data

Pada tahap ini dari data yang sudah terkumpul dimodifikasi yaitu merubah jarak menjadi waktu perjalanan dengan cara membagi jarak antar persimpangan dengan kecepatan rata-rata kendaraan yaitu 40 km/jam dan menyesuaikan durasi lampu lalu-lintas dari tiga fase (merah, hijau, dan kuning) menjadi dua fase (merah dan hijau), dengan mengasumsikan fase kuning adalah fase hijau.

3.2.2.2 Analisa Data

Setelah pengolahan data dilakukan, langkah selanjutnya, dari data tersebut dibuat sebuah graph berarah yang menggambarkan model jaringan jalan yang menghubungkan Ngesrep dan Simpanglima, dimana persimpangan jalan diwakili oleh titik sedangkan jalan direpresentasikan kedalam arc.

Kemudian langkah selanjutnya adalah mencari rute yang mempunyai waktu perjalanan minimum yang menghubungkan Ngesrep – Simpanglima, dengan menggunakan algoritma Ford Moore Bellman yang telah dimodifikasi.

4. HASIL PENELITIAN

Dari hasil pengolahan data diperoleh waktu perjalanan minimum yang dibutuhkan untuk bepergian dari Ngesrep ke Simpanglima adalah 659 detik atau 10 menit 59 detik. Rute yang mempunyai waktu perjalanan minimum tersebut adalah :
Setya Budi $\rightarrow$ Teuku Umar $\rightarrow$ Sultan Agung $\rightarrow$ Diponegoro $\rightarrow$ Pahlawan $\rightarrow$ Simpanglima.

5. KESIMPULAN

Algoritma Ford Moore Bellman digunakan untuk mencari lintasan terpendek dari titik s ke titik j yang memuat paling banyak $\mathrm{k}+1$ garis berarah. Lintasan ini dapat diperoleh dari lintasan terpendek dari titik s ke titik j yang memuat paling banyak k garis berarah. Dengan algoritma Ford Moore Bellman yang dimodifikasi dimaksudkan untuk mengganti bobot lintasan dari jarak menjadi waktu perjalanan titik s ke j.

DAFTAR PUSTAKA

1. Ahuja,R.K.,J.B. Orlin, S. Pallottino, dan M.G. Scutella. *Minimum time and minimum cost path problems in street networks with periodic traffic lights*. Journal In Transportation Science. 2000.

2. Whitelaw, T.A. *Introduction to abstract algebra*.Thirtd edition. New York: Blachie Academic and Profesional. 1995.