

Penentuan Rute Terpendek Tempat Wisata di Kota Tasikmalaya Dengan Algoritma *Floyd-warshall*

Muhamad Fikri Alhwarizmi - 13513009

Program Studi Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

mfikria@students.itb.ac.id

Abstrak—Tasikmalaya merupakan kota yang menyimpan potensi pariwisata yang besar. Hal ini dapat dilihat dari banyaknya tempat wisata yang menyuguhkan berbagai keindahan kepada para pariwisataawan. Untuk mendukung sebuah sistem pengelolaan tempat pariwisata, hal yang mesti diperhatikan adalah rute untuk mencapai tempat pariwisata tersebut dengan efisien dan optimum. Dalam jurnal ini akan dibahas bagaimana menentukan rute terpendek yang dapat menghubungkan tempat transportasi vital seperti terminal dan stasiun kepada tempat-tempat wisata. Pengaplikasian hal tersebut dilakukan dengan menggunakan algoritma *Floyd-warshall*.

Kata Kunci—rute, wisata, Tasikmalaya, *Floyd-washall*.

I. PENDAHULUAN

Kota Tasikmalaya adalah salah satu kota penuh dinamika yang mempunyai berbagai tempat wisata yang memiliki potensi besar untuk mendatangkan banyak pariwisataawan baik lokal maupun mancanegara. Dalam membangun tempat wisata, banyak hal yang perlu diperhatikan, diantaranya mengenai sektor transportasi. Permasalahan yang perlu diperhatikan dalam transportasi. Transportasi merupakan sarana vital yang menunjang keberjalanan suatu tempat wisata.

Kenyataannya salah satu faktor bahwa pariwisataawan yang kebingungan bahkan tidak mengetahui rute tujuan yang menghubungkan antar tempat transportasi vital seperti terminal dan stasiun dengan tempat wisata yang satu dengan yang lainnya hingga pada kasus ekstrimnya, para pendatang pun enggan untuk berkunjung ke tempat wisata tersebut karena rute yang terlalu jauh atau membingungkan. Ini menjadi salah satu perhatian karena akan berdampak buruk yang dapat menimbulkan kerugian bagi para penyedia tempat wisata dan masyarakat pada umumnya seperti sopir angkutan kota, ataupun pemakai angkutan kota dalam segi materi dan waktu.

Karena pentingnya perencanaan rute pada trayek dalam pemetaan tempat wisata, transportasi yang melewati jalur dengan area penting khususnya wisata belanja maka kebutuhan akan informasi mengenai trayek angkutan umum yang melewati titik-titik wisata di kota Tasikmalaya merupakan kebutuhan yang sangat penting.

Wisatawan yang datang berkunjung membutuhkan informasi yang baik tentang rute wisata untuk membantu merencanakan perjalanan wisatanya ke tempat tujuan yaitu obyek wisata yang dituju dan dari obyek wisata hingga kembali lagi ke tempat tinggal asal, penginapan, atau tempat vital lainnya. Selain itu wisatawan juga seharusnya dapat mencari rute terpendek menuju tempat-tempat wisata yang akan dikunjungi agar dapat melakukan efisiensi waktu, biaya, dan jarak.

Pencarian rute terpendek dapat dicari dengan menggunakan algoritma Floyd Warshall. Hasil pencarian rute wisata terpendek menggunakan yang dihasilkan dari algoritma *Floyd-Warshall* ini dapat diterapkan dan diimplementasikan kedepannya dalam membangun aplikasi untuk mengetahui rute wisata yang paling optimum.

II. DASAR TEORI

A. Teori Graf

Graf atau graph merupakan struktur data yang paling umum. Jika struktur linear memungkinkan pendefinisian keterhubungan sekuensial antara entitas data, struktur data tree memungkinkan pendefinisian keterhubungan hirarkis. Graf sering digunakan dalam berbagai disiplin ilmu, seperti pengelolaan intertautan jaringan, memetakan jalan raya antarkota, memodelkan objek tiga dimensi, hingga pemetaan DNA pada manusia.

1. Definisi Graf

Graf G didefinisikan sebagai pasangan himpunan (V,E) , ditulis dengan notasi $G=(V,E)$, yang dalam hal ini V adalah himpunan tidak-kosong dari simpul-simpul (*vertices* atau *node*) dan E adalah himpunan sisi (*edges* atau *arcs*) yang menghubungkan sepasang simpul.[1]

2. Terminologi Graf

2.1. Ketetanggaan (Adjacent)

Dua buah simpul pada graf G dikatakan bertetanggaan jika keduanya terhubung langsung pada suatu sisi. Misal terdapat simpul a dan simpul b

yang dihubungkan dengan suatu sisi, maka dapat dikatakan bahwa simpul a dan b saling bertetangga.

2.2. Besisian (Incidency)

Pada Graf $G=(V,E)$, untuk sembarang $e=(u,v)$, sisi e dikatakan bersisian dengan simpul u dan simpul v.

2.3. Derajat (Degree)

Pada Graf tidak-berarah, derajat suatu simpul adalah jumlah sisi yang bersisian dengan simpul tersebut. Misal, suatu graf G memiliki simpul a yang bersisian dengan tiga sisi, maka dapat dikatakan bahwa simpul a berderajat tiga. Pada Graf berarah, derajat simpul u, $d(u) = \text{din}(u) + \text{dout}(u)$. Dimana, $\text{din}(u) =$ jumlah busur yang menuju simpul u, $\text{dout}(u) =$ jumlah busur yang keluar dari simpul u.

2.4. Lintasan (Path)

Lintasan yang memiliki panjang n dari simpul awal v_0 ke simpul tujuan v_n di dalam graf G adalah barisan berselang-seling antara simpul dan sisi $e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n$ sehingga $e_1 = (v_0, v_1)$, $e_2 = (v_1, v_2), \dots, e_n = (v_{n-1}, v_n)$ adalah sisi-sisi dari graf G. Misal dalam graf $G=(V,E)$ terdapat sisi e_1 yang menghubungkan simpul i dengan j, sisi e_2 yang menghubungkan j dengan k, dan sisi e_3 yang menghubungkan k dengan l, maka dapat dikatakan bahwa dari i menuju l dapat melewati lintasan i-j-k-l.

2.5. Siklus (Cycle)

Siklus adalah lintasan yang berawal dan berakhir pada simpul yang sama. Misal terdapat suatu siklus a-b-c-d-a atau siklus x-y-z-x pada suatu graf.

2.6. Terhubung (Connected)

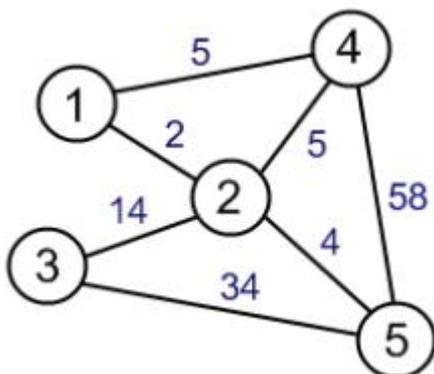
Graf berarah G dikatakan terhubung jika untuk sepasang simpul u dan v terdapat lintasan dari u ke v atau dari v ke u. Jika tidak, maka disebut graf takterhubung.

2.7. Upagraf (Subgraph)

Misal $G=(V,E)$ adalah sebuah graf. $G_1=(V_1,E_1)$ adalah upagraf dari G jika $V_1 \subseteq V$ dan $E_1 \subseteq E$. Suatu graf dapat tidak memiliki upagraf atau dapat memiliki lebih dari satu upagraf.

2.8. Graf Berbobot (Weighted Graph)

Graf berbobot adalah graf yang setiap sisinya diberi nilai sebagai suatu bobot.



Gambar 2.2.1. Contoh Graf berbobot dan tidak berarah.

B. Algoritma Floyd-Warshall

Algoritma Floyd-Warshall akan memproses data yang telah direpresentasikan dalam bentuk graf berarah dan berbobot (V,E) , yang berupa daftar simpul (*node/vertex* V) dan daftar sisi (*edge* E). Jumlah bobot yang terdapat pada sisi-sisi pada sebuah jalur dari satu simpul menuju simpul lainnya adalah bobot jalur tersebut [2]. Dalam mendesain pemetaan Graf G maka setiap ruas sisi diberi simpul awal dan simpul akhir, ini digunakan untuk memperoleh daftar simpul (*node/vertex*). Dalam studi kasus mencari rute terpendek, sisi (*edge* E) merupakan ruas jalan. Terlihat juga bahwa tiap ruas sisi memiliki bobot yang merepresentasikan *cost* menempuh dari simpul I ke simpul j. Sisi pada E diperbolehkan memiliki bobot negatif sehingga dengan menggunakan algoritma ini kita dapat memperhitungkan studi kasus yang membutuhkan bobot negatif. Algoritma ini pada dasarnya akan menghitung bobot terkecil dari seluruh sisi yang terdapat dalam graf yang menghubungkan sebuah pasangan simpul, dan kemudian melakukan akumulasi bobot yang telah diproses sampai ke simpul tujuan. Pada saat perhitungan *cost* optimal yang akan dilakukan adalah menentukan semua kemungkinan sisi yang akan dilalui lalu melakukan perhitungan semua kemungkinan rute tersebut dengan cara melihat tiap pasangan rute kemudian membandingkan pasangan rute tersebut untuk mendapatkan pasangan rute lain yang lebih optimum.

```
//Algoritma Floyd Warshall
function FloydWarshall(int[1..n,1..n]
graph)
{
    // Inisialisasi
    var int[1..n,1..n] jarak := graph
    var int[1..n,1..n] sebelum
    for i from 1 to n
        for j from 1 to n
            if jarak[i,j] < Tak-hingga
                sebelum[i,j] := i
    // Perulangan utama pada algoritma
    for k from 1 to n
        for i from 1 to n
            for j from 1 to n
                if jarak[i,j] >
                    jarak[i,k] + jarak[k,j]
                    jarak[i,j] =
                    jarak[i,k] + jarak[k,j]
```

```

        sebelum[i,j] =
    sebelum[k,j]
    return jarak
}

```

Berikut adalah rumus dari algoritma Floydwarshall

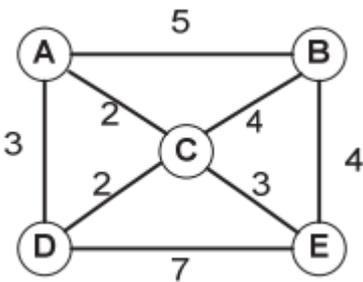
$$d_{ij}^{(k)} = \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}) \quad \text{jika } k \geq 1$$

Algoritma Floyd Warshall bekerja dengan cara membandingkan semua lintasan yang mungkin terjadi dalam suatu graf untuk setiap pasang simpul. Selain itu algoritma ini akan mengujian dari setiap kombinasi simpul yang diperoleh. Misalkan, W_0 merupakan matriks ketetanggaan awal graf berarah berbobot dan W^* adalah matriks ketetanggaan yang berbobot terpendek dengan W_{ij} sama dengan path terpendek dari titik v_i ke v_j . Kelebihan dari algoritma Floyd Warshall antara lain (Adams, 2012):

1. Algoritma Floyd Warshall dapat digunakan untuk mencari jarak terpendek (shortest path) dari setiap pasangan node
2. Algoritma Floyd Warshall menggunakan matriks bobot $n \times n$ sebagai masukan, dimana n merupakan jumlah simpul.
3. Algoritma Floyd Warshall dapat mentolerir sisi yang berbobot negatif.

Ide utama dari algoritma *Floyd-Warshall* adalah tentang bagaimana menentukan seluruh jarak terpendek untuk setiap pasang simpul dan kemudian melakukan perbandingan terhadap pasangan simpul tersebut untuk memperoleh rute yang paling optimum.

Berikut adalah contoh kasus pencarian rute terpendek dari titik A ke E yang dapat dilihat dari gambar dibawah ini.



Gambar 2.3.1. Contoh Graf berbobot.

Gambar 2.3.1 menunjukkan bahwa untuk menuju titik E dari titik A terdapat beberapa jalur yang dapat digunakan. Jalur yang dapat dilewati yaitu melalui titik B, C, atau D. Untuk memudahkan proses pengerjaan dapat dibuat tabel bobot dari setiap pasangan path yang ditunjukkan pada tabel 1.

Tabel 2.3.1. Bobot tiap pasangan path

	A	B	C	D	E
A	0	5	2	3	∞
B	5	0	4	∞	4
C	2	4	0	2	3
D	3	∞	2	0	7
E	∞	4	3	7	0

Perhitungan bobot tiap pasangan sisi tidak dapat langsung ditentukan karena masih ada nilai yang belum pasti seperti $A \rightarrow E$, $B \rightarrow D$ dikarenakan masih adanya beberapa jalur yang dapat digunakan. Oleh karena itu, perlu diambil bobot terkecil dari jalur yang dilewati yang kemudian hasilnya dapat dilihat dan ditentukan nilai optimalnya.

Tabel 2.3.2. Hasil akhir perhitungan bobot pasangan path

	A	B	C	D	E
A	0	5	2	3	5
B	5	0	4	6	4
C	2	4	0	2	3
D	3	6	2	0	7
E	5	4	3	7	0

Pada tabel 2.3.2. memperlihatkan hasil akhir dari perhitungan sebelumnya untuk mencari bobot tiap pasangan sisi sudah bernilai yang terkecil. Dari faka itu, dapat dilihat rute terpendek dari titik A ke titik E yaitu $A \rightarrow C \rightarrow E$.

III. ANALISIS DAN PEMBAHASAN

A. Graf Pemetaan Lokasi Wisata Tasikmalaya

Tahap awal dari pencarian rute terpendek yaitu mentransformasikan peta wisata Tasikmalaya seperti pada gambar 3.1.1. ke dalam diagram grafik yang hasilnya dapat dilihat pada gambar 3.1.2. yang ada pada halaman selanjutnya.



Gambar 3.1.1. Peta Wisata Kota Tasikmalaya
(sumber: <https://ucup33.files.wordpress.com/2010/04/peta-wisata-tasikmalaya.jpg>)

Dari gambar di atas terlihat banyaknya perimpangan dan ruas jalan yang menghubungkan antara suatu tempat dengan tempat yang lain. Dalam studi kasus di jurnal ini akan dilakukan penyederhanaan tempat yang akan dianalisis untuk dicari rute terpendeknya. Tempat-tempat yang akan dianalisis adalah sebagai berikut, yaitu Situ Gede, Kawasan Payung Tasik, Kawasan Batik, Terminal Padayungan, Komplek Olahraga Dadaha, Alun-alun, Stasiun Kereta Api, Terminal Pancasila, Tempat Rekreasi Taman Resik, dan Kawasan Kerajinan Mendong.

Langkah-langkah penelitian yang dilakukan secara garis besar terdiri dari pencarian jalur terpendek menuju beberapa obyek wisata populer di Tasikmalaya dengan rute yang optimum dan terpendek. Tahap selanjutnya yaitu mentransformasikan peta wisata Tasikmalaya ke dalam bentuk diagram grafik. Berdasarkan peta wisata dalam diagram grafik dicari rute wisata terpendek menggunakan konsep algoritma Floyd-Warshall. Hasil dari pencarian rute terpendek diterapkan dalam pembangunan sistem informasi untuk mencari rute terpendek. Sistem informasi yang dibangun dalam bentuk website sehingga hasilnya dapat dipahami dan digunakan masyarakat umum.

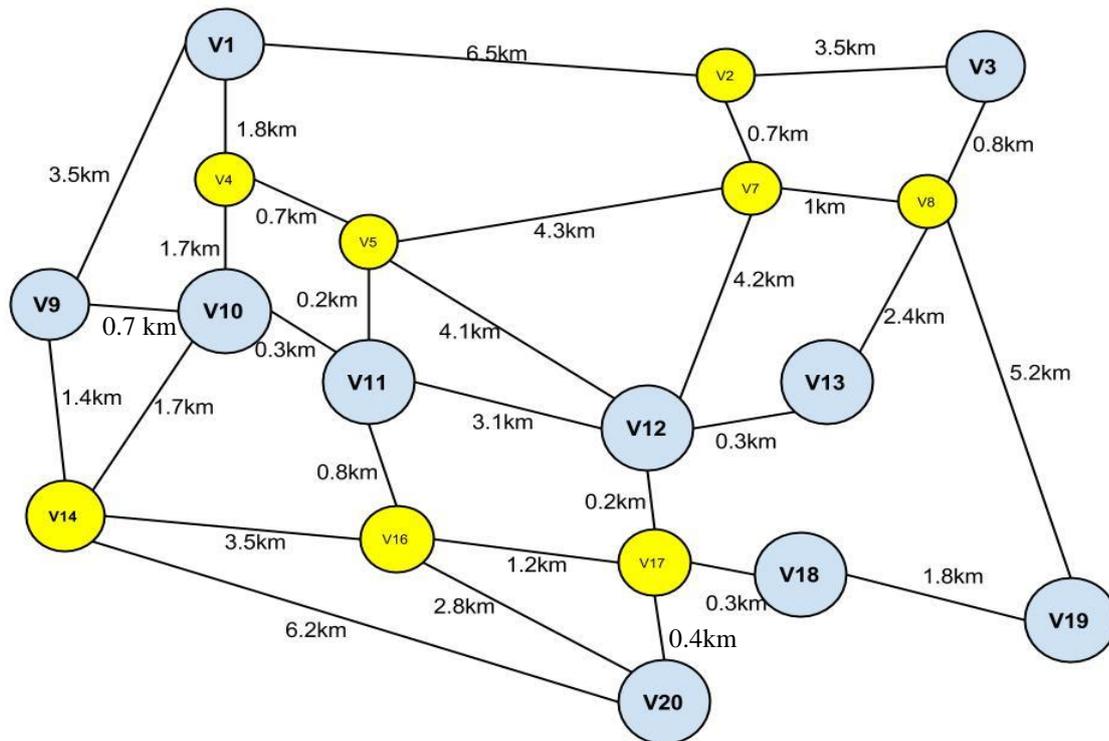
Selanjutnya, dari gambar diatas yang menunjukkan

pemetaan lokasi wisata di Kota Tasikmalaya akan dibuat penyederhanaan peta, dapat dilihat pada gambar 3.1.2.

Hasil pengolahan diagram grafik dengan algoritma Floyd Warshall ditunjukkan dalam tabel 1 sampai dengan tabel 3 sampai dengan tabel 8. Tiap tabel mewakili hasil perhitungan rute terpendek menuju tempat wisata di Tasikmalaya dari satu titik awal ke beberapa titik akhir atau titik tujuan.

Tabel 3.1.1. Deskripsi Simpul Pada Graf

Simpul	Deskripsi
V1	Situ Gede
V3	Kawasan Payung Tasik
V9	Kawasan Batik
V10	Terminal Padayungan
V11	Komplek Olahraga Dadaha
V12	Alun-alun
V13	Stasiun Kereta Api
V18	Terminal Pancasila
V19	Tempat Rekreasi Taman Resik
V20	Kawasan Kerajinan Mendong



Gambar 3.1.2 Graf Representasi Pemetaan Tempat Wisata

Data jarak pada gambar di atas diambil dari data kasar hasil pengukuran pada *google maps*. Oleh karena itu masih terdapat banyak galat dalam penghitungan rute terpendek dengan kondisi sebenarnya.

Algoritma Floyd Warshall untuk mencari path terpendek. Dimisalkan W_0 adalah matriks ketetanggaan awal graf berarah berbobot. W^* adalah matriks ketetanggaan berbobot terpendek dengan W_{ij} sama dengan path terpendek dari titik v_i ke v_j .

1) $W = W_0$

2) Untuk $k = 1$ hingga n , lakukan: Untuk $i = 1$ hingga n , lakukan: Untuk $j = 1$ hingga n lakukan:

3) Jika $W_{i,j} > W_{i,k} + W_{k,j}$ maka Tukar $W_{i,j}$ dengan $W_{i,k} + W_{k,j}$

4) $W^* = W$

Iterasi untuk $k = 1$ Pada setiap elemen matriks W dilakukan pengecekan apakah $W_{i,j} > W_{i,k} + W_{k,j}$. Jika ya, maka ganti $W_{i,j}$ dengan $W_{i,k} + W_{k,j}$.

Contoh : $W_{1,2} = 6,5$, sedangkan $W_{1,1} + W_{1,2} = \infty + 6,5 = \infty$ Karena $W_{1,2} \not> W_{1,1} + W_{1,2}$ maka bobot $W_{1,2}$ tidak diubah. $W_{2,4} = \infty$, sedangkan $W_{2,1} + W_{1,4} = 6,5 + 1,8 = 8,3$. Karena $W_{2,4} > W_{2,1} + W_{1,4}$, maka bobot $W_{2,4}$ diubah menjadi 8,3. Berarti, ada jalur dari V_2 ke V_4 melalui V_1 yang mempunyai bobot lebih kecil yaitu path $V_2 V_1 V_4$ dengan jumlah bobot 8,3. Kemudian dengan cara yang sama, harga $W_{i,j}$ dihitung untuk setiap

i dan j . Penghitungan iterasi dilakukan hingga iterasi $k = 20$. Untuk mengetahui jalur terpendek dari setiap titik tempat wisata maka perhatikan perubahan bobot dari setiap iterasi. Misalnya dari titik Stasiun Kereta Api menuju Kawasan Kerajinan Mendong, jarak terpendek dari (V_{12}) ke (V_{20}) sebesar 0,6 km. Hal ini berarti terdapat jalur-jalur sejauh 0,6 km untuk menuju Kawasan Kerajinan Mendong di (V_{20}). Lakukan pengecekan dari (V_{12}) ke (V_{20}) pada setiap k untuk mengetahui perubahan setiap bobotnya. (V_{12}) ke (V_{20}). Kemudian perhatikan graf untuk mengetahui (V_{12}) \rightarrow (V_{11}) \rightarrow (V_{16}) \rightarrow (V_{20}) yang berjarak terpendek dari (V_{12}) ke (V_{20}) sebesar 6,7 km. Lakukan pengecekan dari (V_{12}) ke (V_{20}) pada setiap k untuk mengetahui perubahan setiap bobotnya. (V_{12}) ke (V_{20}) memiliki bobot 0,6 km pada saat $k = n$ hal ini berarti terdapat jalur terpendek dari (V_{12}) ke (V_{20}) melalui suatu simpul yang terhubung. Setelah iterasi sampai $k=20$ dicapai maka diperoleh jalur terpendek dari (V_{12}) ke (V_{20}) yaitu (V_{12}) \rightarrow (V_{17}) \rightarrow (V_{20}) sejauh 0,6 km.

Hasil dari iterasi dengan algoritma Floyd-Warshall dapat dilihat pada Tabel 3.1.2. yang menunjukkan hasil pemrosesan peta dalam bentuk diagram grafik dengan algoritma Floyd Warshall berupa rute terpendek dari tempat vital transportasi ke objek-objek wisata di Tasikmalaya.

Tabel 3.1.2. Hasil Penghitungan Rute Terpendek Dengan Algoritma *Floyd-Warshall*

Terminal/Stasiun	Tempat Wisata	Jalur Terpendek	Jarak
Terminal Padayungan	Situ Gede	V10-V4-V1	2,5 km
	Kawasan Payung Tasik	V10-V11-V5-V7-V8-V3	6,6 km
	Kawasan Batik	V10-V9	0,7 km
	Komplek Olahraga Dadaha	V10-V11	0,3 km
	Alun-alun	V10-V11-V12	3,4 km
	Tempat Rekreasi Taman Resik	V10-V11-V16-V17-V18-V19	4,4 km
	Kawasan Kerajinan Mendong	V10-V11-V16-V20	3,9 km
Stasiun Kereta Api	Situ Gede	V13-V12-V17-V16-V11-V5-V4-V1	5,2 km
	Kawasan Payung Tasik	V13-V8-V3	3,2 km
	Kawasan Batik	V13-V12-V17-V16-V11-V10-V9	3,5 km
	Komplek Olahraga Dadaha	V13-V12-V17-V16-V11	2,5 km
	Alun-alun	V13-V12	0,3 km
	Tempat Rekreasi Taman Resik	V13-V12-V17-V18-V19	2,6 km
	Kawasan Kerajinan Mendong	V13-V12-V17-V20	0,9 km
Terminal Pancasila	Situ Gede	V18-V18-V17-V16-V11-V5-V4-V1	5,0 km
	Kawasan Payung Tasik	V18-V17-V12-V13-V8-V3	4,0 km
	Kawasan Batik	V18-V18-V17-V16-V11-V10-V9	3,3 km
	Komplek Olahraga Dadaha	V18-V18-V17-V16-V11	2,3 km
	Alun-alun	V18-V17-V12	0,5 km
	Tempat Rekreasi Taman Resik	V18-V19	1,8 km
	Kawasan Kerajinan Mendong	V18-V17-V20	0,7 km

IV. KESIMPULAN

Algoritma Floyd Warshall dapat diterapkan dalam pencarian rute terpendek yang menghubungkan tempat-tempat vital transportasi seperti terminal dan stasiun menuju tempat-tempat wisata. Hasilnya dapat diterapkan dalam pembangunan sistem informasi untuk rute wisata terpendek yang kedepannya dapat diimplementasikan ke dalam aplikasi berbasis web. Dengan mengetahui rute terpendek, hal ini dapat membantu wisatawan mendapatkan informasi tentang objek-objek wisata populer di Tasikmalaya beserta informasi jalur terpendek yang dapat dilalui untuk menuju ke objek wisata yang akan dituju.

V. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Allah SWT karena atas rahmat-Nya, penulis dapat menyelesaikan makalah ini. Terima kasih juga penulis ucapkan kepada kedua orangtua yang telah memberikan dukungan dan motivasi. Penulis juga berterima kasih kepada Bapak Rinaldi Munir, selaku dosen Strategi Algoritma semester genap tahun ajaran 2014/2015 atas bimbingan yang diberikan selama ini. Tak lupa, ucapan terima kasih penulis sampaikan kepada para teman-teman yang telah membantu keberjalanan dalam menyusun jurnal ini.

REFERENSI

- [1] Munir, Rinaldi. Strategi Algoritma. 2013. Bandung : Penerbit ITB.
- [2] Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L. (1990). Introduction to Algorithms (ed. first edition). MIT Press and McGraw-Hill. ISBN 0-262-03141-8.
- [3] Floyd, Robert W. (June 1962). "Algorithm 97: Shortest Path". Communications of the ACM 5 (6)
- [4] Warshall, Stephen (January 1962). "A theorem on Boolean matrices". Journal of the ACM 9 (1)

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 27 November 2015



Muhamad Fikri Alhwarizmi - 13513009