

Rancang Bangun Aplikasi untuk Menentukan Jalur Terpendek Menggunakan Algoritma Floyd di Lokasi Wisata Purbalingga

(The Design of an Application to Determine the Shortest Path of Purbalingga Tourist at Traction Using Floyd Algorithm)

Irfan Ardiansyah¹⁾ dan Dimara Kusuma Hakim²⁾

¹⁾²⁾ Jurusan Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Purwokerto,
Jl. Raya Dukuhwaluh Po. Box 202 Purwokerto, Jawa Tengah, 532182,

iyansinosuke@yahoo.com

dimarahin@yahoo.com

Abstract - Purbalingga tourism are scattered in various locations complicate for local or foreign tourist in determining the path to go to tourist sites. By determining the starting point of shortest path from the town square of Purbalingga, the shortest path can be searched using Floyd algorithm. The workings of this algorithm are to find the possible paths exist by aggregating and comparing the points with positive weights on the path which has constructed and determined. The result of this research was obtained the shortest path to go to the various tourist sites in Purbalingga.

Keywords: Purbalingga tourism, Floyd Algorithm, Shortest Path.

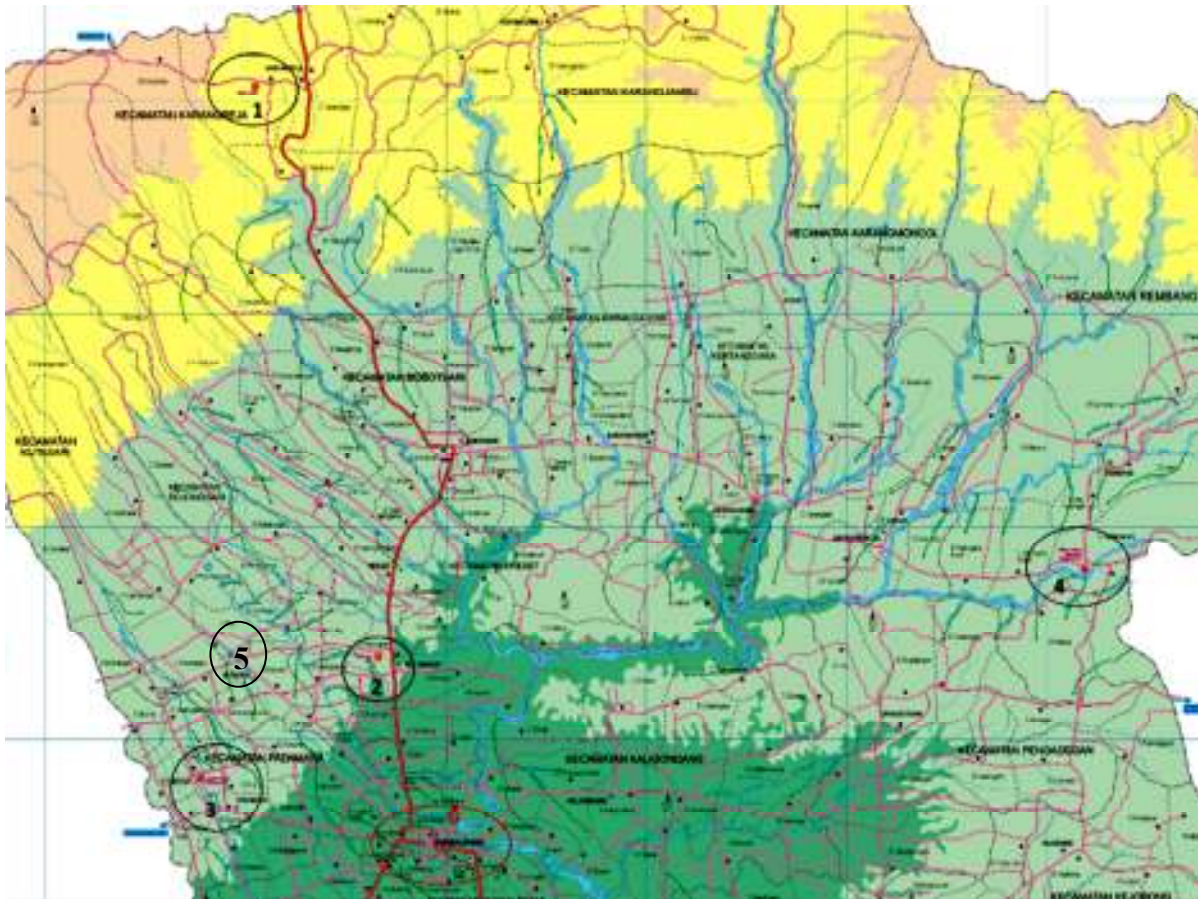
I. PENDAHULUAN

Lokasi wisata di Purbalingga merupakan bagian aset dari daerah, namun dengan banyaknya lokasi wisata dan dengan lokasinya yang tersebar di beberapa daerah di Purbalingga menyulitkan para wisatawan untuk menuju salah satu atau semua dari lokasi wisata di Purbalingga. Disamping lokasinya yang tersebar

sehingga banyak juga jalan yang harus dilalui oleh wisatawan.

Perkembangan jaman dan teknologi yang semakin maju menuntut manusia untuk berfikir semakin kritis dalam menemukan solusi-solusi yang terbaik. Misalnya dalam hal penentuan jalur terpendek agar mencapai suatu tujuan dengan lebih cepat saat melakukan perjalanan. Setiap pengguna jalan selalumenginginkan efisiensi waktu dan jalur perjalanan yang lebih dekat jarak tempuhnya dalam perjalanan menuju suatu lokasi tujuan tertentu. Kemacetan jalanan utama juga menjadi salah satu yang terkadang sering dihadapi oleh pengguna jalan. Untuk menuju lokasi wisata di Purbalingga, semakin jauh jarak yang ditempuh maka semakin lama waktu yang dibutuhkan untuk mencapai tujuan, namun sebaliknya jika semakin dekat maka waktu yang ditempuh semakin cepat dan lebih menghemat waktu.

Lokasi wisata di Purbalingga merupakan aset dari daerah sebagai pemasukan tambahan dari daerah tersebut melalui pemasukan dari wisatawan baik wisatawan lokal ataupun manca negara, dengan berbagai lokasi (Gambar 1).



Gambar 1. Lokasi wisata pada Peta Purbalingga.

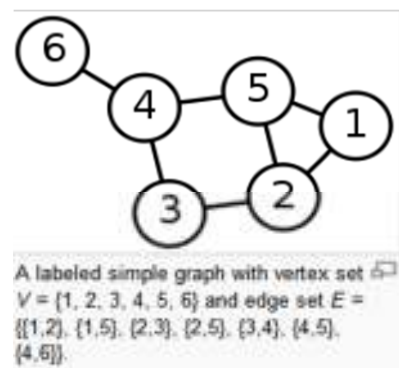
Keterangan:

- Alun-alun Purbalingga berlingkaran merah (0).
- Wisata air (Owabong) di Desa Bojong sari (2).
- Taman Reptil desa Karangbanjar (5).
- Aquarium Raksasa di Desa Purbayasa (3).
- Monumen jendral sudirman di Desa Bantargebang (4).
- Gua Lawa di Desa Siwarak (1).

Algoritma *Floyd* atau Algoritma *Floyd-Warshall* adalah salah satu cabang dari ilmu matematika yang salah satu fungsinya adalah untuk menyelesaikan masalah lintasan terpendek. Dalam Algoritma *Floyd* terdapat fungsi $(G=V,E)$ dengan G = graf yang merupakan kumpulan simpul (*nodes*) yang dihubungkan satu sama lain melalui sisi/busur (*edges*). Suatu Graf G terdiri dari dua himpunan yaitu himpunan V dan himpunan E .

- Verteks (simpul) : V = himpunan simpul yang terbatas dan tidak kosong.
- Edge (sisi/busur): E = himpunan busur yang menghubungkan sepasang simpul.

Notasi graf: $G(V,E)$ artinya graf G memiliki V simpul dan E busur (Gambar 2).

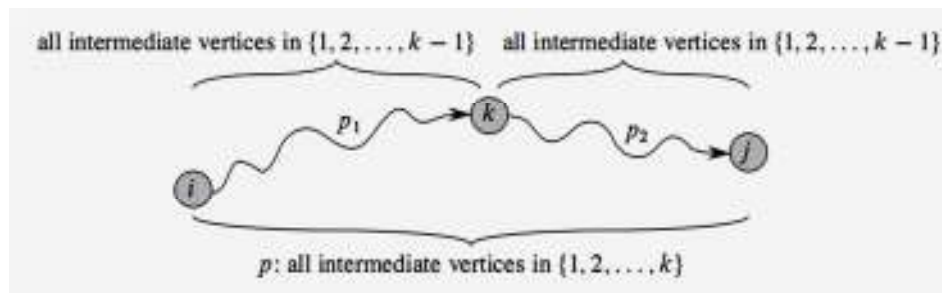


Gambar 2. *Graph set* $G=(V,E)$.

Algoritma *Floyd* membandingkan semua kemungkinan lintasan pada graf untuk setiap sisi dari semua simpul. Hal tersebut bisa terjadi karena adanya perkiraan pengambilan keputusan (pemilihan jalur terpendek) pada setiap tahap antara dua simpul, hingga perkiraan tersebut diketahui sebagai nilai optimal. Misalkan terdapat suatu graf G dengan simpul-simpul V yang masing-masing bernomor 1 s.d. N (sebanyak N buah). Misalkan pula terdapat suatu fungsi *shortest path*(i, j, k) yang mengembalikan kemungkinan jalur terpendek dari i ke j dengan hanya memanfaatkan

simpul 1 s.d. k sebagai titik perantara. Tujuan akhir penggunaan fungsi ini adalah untuk mencari jalur terpendek dari setiap simpul i ke simpul j dengan perantara simpul 1 s.d. $k+1$. Algoritma *Floyd* yang

menerapkan pemrograman dinamis lebih menjamin keberhasilan penemuan solusi optimum untuk kasus penentuan lintasan terpendek (*single pair shortest path*) seperti tampak pada Gambar 3.



Gambar 3. Contoh Diagram jalur terpendek.

Jalur p adalah jalur terpendek dari simpul i ke simpul j , dan k adalah nilai tertinggi antara titik p . Jalur $p1$ adalah jalan atau simpul i ke simpul k semua memiliki simpul antar himpunan $\{1, 2, \dots, k-1\}$. Hal yang sama berlaku untuk jalur $p2$ dari k ke simpul j [2].

Kode sumber dasar Algoritma Floyd tersaji pada Gambar 4 berikut.

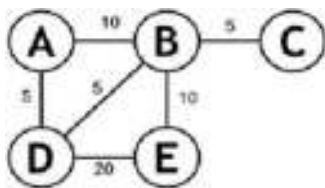
```

procedure FloydWarshall ()
  for k := 1 to n
    for i := 1 to n
      for j := 1 to n
        path[i][j] = min ( path[i][j], path[i][k]+path[k][j] );

```

Gambar 4. Kode sumber dasar Algoritma Floyd

Sebagai contoh kasus penggunaan Algoritma *Floyd*, jika dipunyai diagram seperti pada Gambar 5, selanjutnya dibuat matrik untuk perhitungan iterasi dalam pencarian jalur terpendek (Tabel 1). Hasil iterasi tersaji pada Tabel 2.



Gambar 5. Diagram Pencarian jalur terpendek.

Tabel 1. Matrik pencarian jalur terpendek.

	A	B	C	D	E
A	0	10	~	5	~
B	10	0	5	5	10
C	~	5	0	~	~
D	5	5	~	0	20
E	~	10	~	20	0

Tabel 2. Matrik hasil iterasi pencarian jalur terpendek.

	A	B	C	D	E
A	0	10	15	5	20
B	10	0	5	5	10
C	15	5	0	10	15
D	5	5	10	0	15
E	20	10	15	15	0

Berdasarkan tabel tersebut, terdapat beberapa jalur (*path*) antara A dan E yaitu:

Path 1 : A -> B -> E 20

Path 2 : A -> D -> E 25

Path 3 : A -> B -> D -> E 35

Path 4 : A -> D -> B -> E 20

Pada path 1 dan path 4 memiliki jumlah bobot yang sama namun pada algoritma ini diperhatikan agar hasil akhir adalah se-optimum mungkin. Maka jalur terpendek yang akan dimunculkan adalah path 1 yaitu Path 1 : A -> B -> E = 20.

Berdasarkan uraian tersebut, maka Algoritma Floyd bisa digunakan untuk menyelesaikan masalah atau memberikan solusi lintasan terpendek dari tiap titik ke semua titik lain terutama dalam pencarian lintasan terpendek untuk menuju lokasi wisata di Purbalingga. Untuk mewujudkan suatu aplikasi menggunakan Algoritma Floyd maka diperlukan konversi dari perhitungan menjadi bahasa pemrograman yang dapat diolah oleh sistem komputer. Dalam penelitian ini pembangunan program computer menggunakan bahasa pemrograman Java.

Java adalah sebuah bahasa pemrograman berbasis *Object Oriented Programming* (OOP) yang sederhana dan tidak tergantung pada *platform* Sistem Operasi. Java didesain sedemikian rupa sehingga ukurannya kecil, sederhana, dan *portable* (dapat dipindah-pindahkan di antara bermacam sistem operasi). Program yang dihasilkan dengan bahasa Java dapat berupa

applet (aplikasi kecil yang jalan di atas web browser) maupun berupa aplikasi mandiri yang dijalankan dengan program Java Interpreter. Bahasa JAVA yang diperkenalkan pada tahun 1995 oleh Sun Microsystems Corp, Inc. Disamping itu dengan memakai bahasa pemrograman java dapat memasukkan berbagai macam program aplikasi animasi, multimedia, dan *data base* ke dalam situs web [4]. Selain Java, dalam penyimpanan data jalur dan jarak dilakukan menggunakan database. Dengan database ini, data yang digunakan atau diperuntukkan terhadap banyak 'user' dimana masing-masing 'user' (baik menggunakan teknik pemrosesan yang bersifat *batch* atau *on-line*) akan menggunakan data tersebut sesuai dengan tugas dan fungsinya, dan 'user' lain dapat juga menggunakan data tersebut dalam waktu yang bersamaan [3]. Karena system yang dibangun berbasis web, maka digunakan juga perangkat lunak *Adobe Dreamweaver* yang merupakan perangkat lunak yang ditujukan untuk membuat suatu situs web. *Dreamweaver CS4* merupakan versi terbaru yang memiliki performa lebih baik dan memiliki tampilan yang memudahkan anda untuk membuat dan mengelola halaman web [1]. Dalam kaitannya pembangunan peta jalur, digunakan konsep GIS (*Geography Informatics Sistem*) yang merupakan ilmu yang mempelajari tentang pemetaan. *Tools* yang digunakan adalah ArcGis 9.3 untuk mengolah dan membentuk peta jalan menuju lokasi wisata di Purbalingga [6].

Dengan penelitian ini diharapkan memudahkan wisatawan atau masyarakat dalam pencarian jalur terpendek yang akan ditempuh untuk menuju lokasi wisata di Purbalingga.

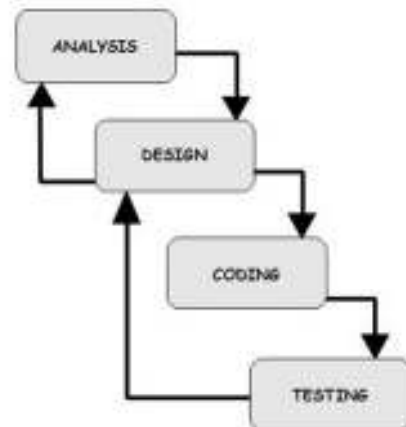
II. METODE

Penelitian ini dilakukan di Kabupaten Purbalingga Jawa Tengah. Sedangkan alat yang digunakan adalah sebuah PC (*Personal Unit*) dengan sistem operasi *Windows 7 Ultimate* yang di dukung dengan *tools Adobe Dreamweaver CS4* sebagai pembuat aplikasi dan *SQL Server 2005* sebagai *data base*. Sedangkan perangkat pengumpulan data berupa GPS untuk menghitung titik-titik jalan dan jarak yang akan dilalui dari titik awal (alun-alun) menuju berbagai lokasi wisata di Purbalingga.

Data yang digunakan adalah data primer dengan melakukan pengukuran langsung jarak dari titik awal (alun-alun) ke berbagai titik jalan menuju lokasi wisata Purbalingga yang tersebar di berbagai Kecamatan di Kabupaten Purbalingga menggunakan GPS.

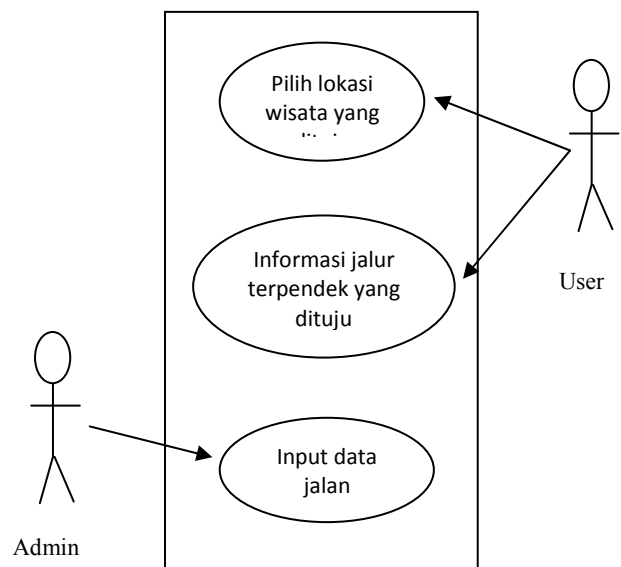
Langkah operasional penelitian ini Metode Pengembangan sistem menggunakan model *Waterfall* karena metode ini sering digunakan oleh penganalisa

system. Secara garis besar metode *waterfall* mempunyai langkah-langkah sebagai berikut : Analisa, Design, Code dan Testing, Penerapan dan Pemeliharaan seperti tersaji pada Gambar 6.



Gambar 6. Diagram Metode *Waterfall* [5]

Dalam perancangan aplikasinya digunakan *use case diagram* (Gambar 7).



Gambar 7. *Use Case Diagram* pada aplikasi algoritma Floyd

Penyelesaian Algoritma Floyd dilakukan dengan langkah-langkah sebagai berikut:

- Mengkonstruksi jalur yang akan dilalui.
- Memberi bobot sisi yang keluar dari titik baru tersebut dengan 0 (titik awal).
- Mencari lintasan terpendek dari titik baru ke semua titik lain. Lintasan terpendek tersebut digunakan untuk mengubah bobot semua sisi pada *digraph* baru agar bobot semua sisi bernilai positif.
- Mencari lintasan terpendek dari tiap titik ke semua titik lain dan mengubah hasilnya dengan menggunakan lintasan terpendek dari titik baru ke semua titik lain.

- Hasil dari perhitungan ini berupa matriks. Dari matriks ini dapat diketahui panjang lintasan terpendek dari tiap titik ke semua titik lain. Untuk menghitung lintasan terpendek dari titik baru ke semua titik lain yang berguna untuk mengubah semua bobot menjadi positif digunakan Algoritma *Floyd*.

III. HASIL DAN PEMBAHASAN

A. Kebutuhan Data jalan

Obyek wisata besar yang ada di Kabupaten Purbalingga adalah Owabong, Akuarium Raksasa, Taman Reptil, Monumen Jendral Sudirman, dan Goa Lawa. Data jalan digunakan untuk mengetahui tujuan dan jalur yang akan dilalui untuk sampai ke obyek wisata tersebut. Satuan yang digunakan dalam jarak jalur ini adalah meter. Sebagai berikut data jalan yang digunakan dalam pencarian jalur terpendek pada lokasi wisata di Purbalingga (Tabel 3 – 7).

Tabel 3. Data jalan jalur Owabong

No	Titik	Nama Jalan	Jarak
1	A	Alun-Alun	0
2	A-B	Jl. Jambu Karang	250
3	A-C	Jl. Let. Achmad Nur	300
4	B-D	Jl. Mayjend Panjaitan	500
5	C-D	Jl. Wirasaba	100
6	D-E	Jl. Aw Sumarmo	600
7	D-F	Jl. Banjaransari	1100
8	E-F	Jl. Sudani	210
9	F-G	Jl. Karanglewas	1900
10	G-H	Jl. Kutasari	1100
11	I-J	Jl. Karangbanjar	1100
12	J-L	Jl. Karangbanjar	1100
13	E-K	Jl. Raya Bojongsari	3700
14	K-L	Jl. Raya Owabong	800

Tabel 4. Data jalan jalur Aquarium Raksasa

No	Titik	Nama Jalan	Jarak
1	A	Alun-Alun	0
2	A-B	Jl. Jambu Karang	250
3	A-C	Jl. Let. Achmad Nur	300
4	B-D	Jl. Mayjend Panjaitan	500
5	C-D	Jl. Wirasaba	100
6	D-E	Jl. Koprul Tanwir	1100
7	E-F	Jl. Sudani	1100
8	E-G	Jl. Gemuruh	3300
9	G-I	Jl. Prigi	2300
10	F-H	Jl. Padamara	3400
11	H-I	Jl. Purbasari	800

Tabel 5. Data jalan jalur Taman Reptil.

No	Titik	Nama Jalan	Jarak
1	A	Alun-Alun	0
2	A-B	Jl. Jambu Karang	250
3	A-C	Jl. Let. Achmad Nur	300
4	C-D	Jl. Wirasaba	100
5	B-D	Jl. Mayjend Panjaitan	500
6	D-E	Jl. Banjaransari	1100
7	E-F	Jl. Karang Lewas	1900
8	F-G	Jl. Kutasari	1000
9	G-H	Jl. Kutasari	200
10	F-H	Jl. Kutasari	1800

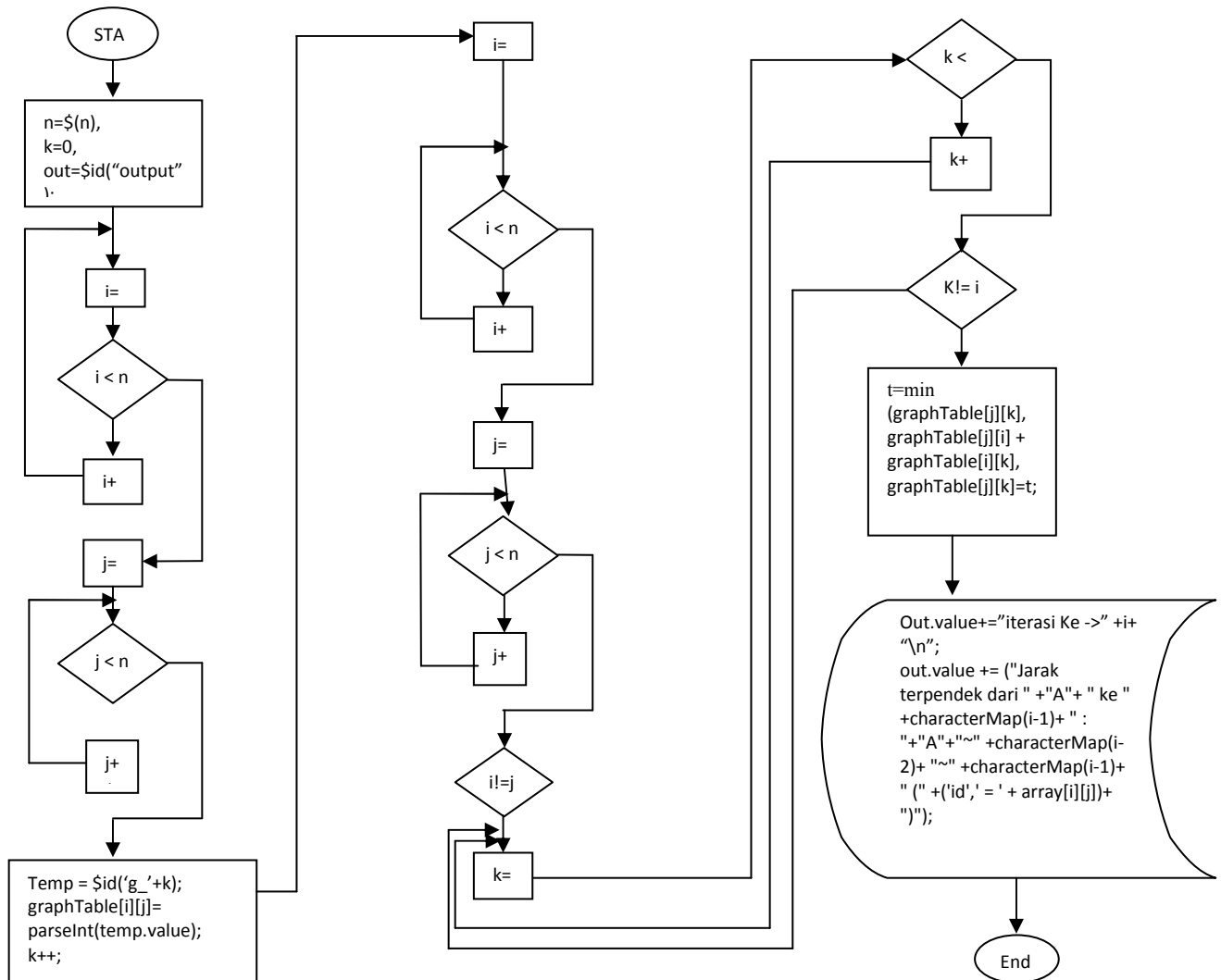
Tabel 6. Data jalan jalur Monumen Jendral Sudirman.

No	Titik	Nama Jalan	Jarak
1	A	Alun-alun	0
2	A-B	Jl. Kapten Pieretendean	200
3	A-C	Jl. Onje	200
4	B-D	Jl. Letkol Isdiman	1200
5	C-E	Jl. Dipokusumo	300
6	A-F	Jl. Jendral Soedirman	1500
7	D-F	Jl. Letjen Supratman	300
8	E-F	Jl. Pujawiyoto	200
9	F-G	Jl. Penaruban	2000
10	G-H	Jl. Kaligondang	5400
11	H-I	Jl. Sinduraja	1200
12	I-J	Jl. Pengadegan	9600
13	J-K	Jl. Wlahar	4700
14	K-L	Jl. Bantarbarang	600

Tabel 7. Data jalan jalur Goa Lawa

No	Titik	Nama Jalan	Jarak
1	A	Alun-Alun	0
2	A-B	Jl. Jambu Karang	250
3	A-C	Jl. Let. Achmad Nur	300
4	B-D	Jl. Mayjend Panjaitan	500
5	C-D	Jl. Wirasaba	100
6	D-E	Jl. A.W. Sumarmo	1200
7	E-F	Jl. Bojongsari	4100
8	F-G	Jl. Mangunegara	3400
9	G-H	Jl. Serayu	7700
10	G-I	Jl. Kampung Baru Gandasuli	2600
11	I-J	Jl. Tlagayasa	2900
12	J-K	Jl. Karangreja	8500
13	H-L	Jl. Serayu	14300
14	K-L	Jl. Goalawa	2500

B. Flowchart Algoritma Floyd (Gambar 8).

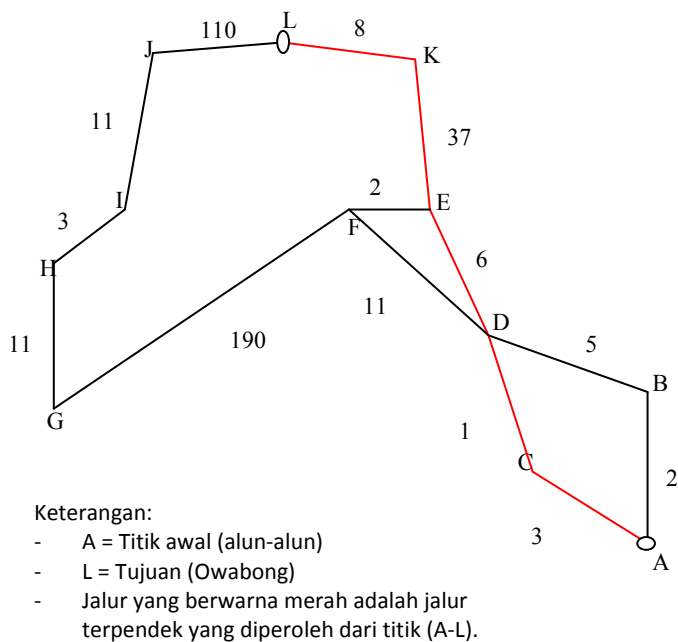


Gambar 8. Flowchart Algoritma Floyd

C. Langkah pencarian jalur terpendek pada masing-masing lokasi wisata

Contoh aplikasi ini diterapkan pada obyek wisata Owabong (Gambar 9). Untuk mulai pencarian jalur terpendek menggunakan Algoritma Floyd yang dimulai dari titik awal atau 0 adalah alun-alun dengan titik tujuan Owabong telah diketahui A-L. Algoritma Floyd melakukan pencarian jalur terpendek dengan cara

menjumlahkan dan membandingkan kemungkinan semua jalur yang ada. Pencarian ini dapat dilakukan dari sembarang titik, artinya tidak harus dari titik A. Karena algoritma ini menggunakan cara iterasi (proses perulangan dengan cara menjumlahkan dan membandingkan nilai bobot antar titik yang positif atau terkecil yang dapat dilakukan dari sembarang titik tertentu yang dimasukkan dalam program *matrix*).



Gambar 9. Rute jalan Owabong.

Matrix dari alun-alun menuju Owabong disajikan dalam Tabel 8 berikut.

Tabel 8. Matrix dari alun-alun menuju Owabong

	A	B	C	D	E	F	G	H	I	J	K	L
A	0	250	300	-	-	-	-	-	-	-	-	-
B	250	0	-	500	-	-	-	-	-	-	-	-
C	300	-	0	100	-	-	-	-	-	-	-	-
D	-	500	100	0	1100	700	-	-	-	-	-	-
E	-	-	-	1100	0	210	-	-	-	-	3700	-
F	-	-	-	700	210	0	1900	-	-	-	-	-
G	-	-	-	-	-	1900	0	1100	-	-	-	-
H	-	-	-	-	-	-	1100	0	300	-	-	-
I	-	-	-	-	-	-	-	300	0	1100	-	-
J	-	-	-	-	-	-	-	-	1100	0	-	1100
K	-	-	-	-	3700	-	-	-	-	-	0	800
L	-	-	-	-	-	-	-	-	-	1100	800	0

Setelah Matrik utama diketahui maka langkah selanjutnya adalah pencarian jalur terpendek satu per satu menggunakan sistem iterasi (Tabel 9 – 20).

Iterasi 0

$$BC = AB + AC$$

$$BC = 250 + 300$$

$$BC = 550$$

Tabel 9. Matrik Hasil iterasi 0

	A	B	C	D	E	F	G	H	I	J	K	L
A	0	250	300	-	-	-	-	-	-	-	-	-
B	250	0	550	500	-	-	-	-	-	-	-	-
C	300	550	0	100	-	-	-	-	-	-	-	-
D	-	500	100	0	600	1100	-	-	-	-	-	-
E	-	-	-	600	0	210	-	-	-	-	3700	-
F	-	-	-	1100	210	0	1900	-	-	-	-	-
G	-	-	-	-	-	1900	0	1100	-	-	-	-
H	-	-	-	-	-	-	1100	0	300	-	-	-
I	-	-	-	-	-	-	-	300	0	1100	-	-
J	-	-	-	-	-	-	-	-	1100	0	-	1100
K	-	-	-	-	3700	-	-	-	-	-	0	800
L	-	-	-	-	-	-	-	-	-	1100	800	0

Iterasi 1

$$AD = AB + BD$$

$$AD = 250 + 500$$

$$AD = 750$$

Tabel 10. Matrik hasil iterasi 1

	A	B	C	D	E	F	G	H	I	J	K	L
A	0	250	300	750	-	-	-	-	-	-	-	-
B	250	0	550	500	-	-	-	-	-	-	-	-
C	300	550	0	100	-	-	-	-	-	-	-	-
D	750	500	100	0	600	1100	-	-	-	-	-	-
E	-	-	-	600	0	210	-	-	-	-	3700	-
F	-	-	-	1100	210	0	1900	-	-	-	-	-
G	-	-	-	-	-	1900	0	1100	-	-	-	-
H	-	-	-	-	-	-	1100	0	300	-	-	-
I	-	-	-	-	-	-	-	300	0	1100	-	-
J	-	-	-	-	-	-	-	-	1100	0	-	1100
K	-	-	-	-	3700	-	-	-	-	-	0	800
L	-	-	-	-	-	-	-	-	-	1100	800	0

Iterasi 2

$$AD = AC + CD$$

$$AD = 300 + 100$$

$$AD = 400$$

Tabel 11. Matrik hasil iterasi 2

	A	B	C	D	E	F	G	H	I	J	K	L
A	0	250	300	400	-	-	-	-	-	-	-	-
B	250	0	550	500	-	-	-	-	-	-	-	-
C	300	550	0	100	-	-	-	-	-	-	-	-
D	400	500	100	0	600	1100	-	-	-	-	-	-
E	-	-	-	600	0	210	-	-	-	-	3700	-
F	-	-	-	1100	210	0	1900	-	-	-	-	-
G	-	-	-	-	-	1900	0	1100	-	-	-	-
H	-	-	-	-	-	-	1100	0	300	-	-	-
I	-	-	-	-	-	-	-	300	0	1100	-	-
J	-	-	-	-	-	-	-	-	1100	0	-	1100
K	-	-	-	-	3700	-	-	-	-	-	0	800
L	-	-	-	-	-	-	-	-	-	1100	800	0

Iterasi 3
 $AE=AD+DE$ $AF=AD+DF$ $BF=BD+DF$ $BE=BD+DE$
 $AE=400+600$ $AF=400+810$ $BF=500+810$ $BE=500+600$
 $AE=1000$ $AF=1210$ $BF=1310$ $BE=1100$

$CE=CD+DE$ $CF=CD+DF$
 $CE=100+600$ $CF=100+810$
 $CE=700$ $CF=910$

Tabel 12.Matrik hasil iterasi 3

	A	B	C	D	E	F	G	H	I	J	K	L
A	0	250	300	400	1000	1500	-	-	-	-	-	-
B	250	0	550	500	1100	1600	-	-	-	-	-	-
C	300	550	0	100	700	1200	-	-	-	-	-	-
D	400	500	100	0	600	1100	-	-	-	-	-	-
E	1000	1100	700	600	0	210	-	-	-	-	3700	-
F	1500	1600	1200	1100	210	0	1900	-	-	-	-	-
G	-	-	-	-	-	1900	0	1100	-	-	-	-
H	-	-	-	-	-	-	1100	0	300	-	-	-
I	-	-	-	-	-	-	-	300	0	1100	-	-
J	-	-	-	-	-	-	-	-	1100	0	-	1100
K	-	-	-	-	3700	-	-	-	-	-	0	800
L	-	-	-	-	-	-	-	-	-	1100	800	0

Iterasi 4
 $AK=AE+EK$ $BK=BE+EK$ $CK=CE+EK$
 $AK=1000+3700$ $BK=1100+3700$ $CK=700+3700$
 $AK=4700$ $BK=4800$ $CK=4400$

$DK=DE+EK$ $FK=EF+EK$
 $DK=600+3700$ $FK=210+3700$
 $DK=4300$ $FK=3910$

Tabel 13. Matrik hasil iterasi 4

	A	B	C	D	E	F	G	H	I	J	K	L
A	0	250	300	400	1000	1210	-	-	-	-	4700	-
B	250	0	550	500	1100	1310	-	-	-	-	4800	-
C	300	550	0	100	700	910	-	-	-	-	4400	-
D	400	500	100	0	600	810	-	-	-	-	4300	-
E	1000	1100	700	600	0	210	-	-	-	-	3700	-
F	1210	1310	910	810	210	0	1900	-	-	-	3910	-
G	-	-	-	-	-	1900	0	1100	-	-	-	-
H	-	-	-	-	-	-	1100	0	300	-	-	-
I	-	-	-	-	-	-	-	300	0	1100	-	-
J	-	-	-	-	-	-	-	-	1100	0	-	1100
K	4700	4800	4400	4300	3700	3910	-	-	-	-	0	800
L	-	-	-	-	-	-	-	-	-	1100	800	0

Iterasi 5
 $AG=AF+FG$ $BG=BF+FG$ $CG=CF+FG$
 $AG=1210+1900$ $BG=1310+1900$ $CG=910+1900$
 $AG=3110$ $BG=3210$ $CG=2810$
 $DG=DF+FG$ $EG=EF+FG$ $GK=FK+FG$
 $DG=810+1900$ $EG=210+1900$ $GK=3910+1900$
 $DG=2710$ $EG=2110$ $GK=5810$

Tabel 14. Matrik hasil iterasi 5

	A	B	C	D	E	F	G	H	I	J	K	L
A	0	250	300	400	1000	1210	3110	-	-	-	4700	-
B	250	0	550	500	1100	1310	3210	-	-	-	4800	-
C	300	550	0	100	700	910	2810	-	-	-	4400	-
D	400	500	100	0	600	810	2710	-	-	-	4300	-
E	1000	1100	700	600	0	210	2110	-	-	-	3700	-
F	1210	1310	910	810	210	0	1900	-	-	-	3910	-
G	3110	3210	2810	2710	2110	1900	0	1100	-	-	5810	-
H	-	-	-	-	-	-	1100	0	300	-	-	-
I	-	-	-	-	-	-	-	300	0	1100	-	-
J	-	-	-	-	-	-	-	-	1100	0	-	1100
K	4700	4800	4400	4300	3700	3910	5810	-	-	-	0	800
L	-	-	-	-	-	-	-	-	-	1100	800	0

Iterasi 6
 $AH=AG+GH$ $BH=BG+GH$ $CH=CG+GH$
 $AH=3110+1100$ $BH=3210+1100$ $CH=2810+1100$
 $AH=4210$ $BH=4310$ $CH=3910$

$KH=GK+GH$ $DH=DG+GH$ $EH=EG+GH$
 $KH=5810+1100$ $DH=2710+1100$ $EH=2110+1100$
 $KH=6910$ $DH=3810$ $EH=3210$
 $FH=FG+GH$
 $FH=1900+1100$
 $FH=3000$

Tabel 15. Matrik hasil iterasi 6

	A	B	C	D	E	F	G	H	I	J	K	L
A	0	250	300	400	1000	1210	3110	4210	-	-	4700	-
B	250	0	550	500	1100	1310	3210	4310	-	-	4800	-
C	300	550	0	100	700	910	2810	3910	-	-	4400	-
D	400	500	100	0	600	810	2710	3810	-	-	4300	-
E	1000	1100	700	600	0	210	2110	3210	-	-	3700	-
F	1210	1310	910	810	210	0	1900	3000	-	-	3910	-
G	3110	3210	2810	2710	2110	1900	0	1100	-	-	5810	-
H	4210	4310	3910	3810	3210	3000	1100	0	300	-	6910	-
I	-	-	-	-	-	-	-	300	0	1100	-	-
J	-	-	-	-	-	-	-	-	1100	0	-	1100
K	4700	4800	4400	4300	3700	3910	5810	6910	-	-	0	800
L	-	-	-	-	-	-	-	-	-	1100	800	0

Iterasi 7
 $AI=AH+HI$ $BI=BH+HI$ $CI=CH+HI$
 $AI=4210+300$ $BI=4310+300$ $CI=3910+300$
 $AI=4510$ $BI=4610$ $CI=4210$

$GI=GH+HI$ $KI=KH+GI$ $DI=DH+HI$
 $GI=1100+300$ $KI=6910+300$ $DI=3810+300$
 $GI=1400$ $KI=7210$ $DI=4110$

EI=EH+HI
EI=3210+300
EI=3510

FI=FH+HI
FI=3000+300
FI=3300

Tabel 16. Matrik hasil iterasi 7

	A	B	C	D	E	F	G	H	I	J	K	L
A	0	250	300	400	1000	1200	300	4200	4500	-	4700	-
B	250	0	550	500	1100	1300	3200	4300	4600	-	4800	-
C	300	550	0	100	700	900	2800	3900	4200	-	4400	-
D	400	500	100	0	600	800	2700	3800	4000	-	4300	-
E	1000	1100	700	600	0	200	2100	3200	3500	-	3700	-
F	1200	1300	900	800	200	0	1900	3000	3300	-	3900	-
G	300	3200	2800	2700	200	1900	0	1100	1400	-	5800	-
H	4200	4300	3900	3800	3200	3000	1100	0	300	-	6900	-
I	4500	4600	4200	4000	3500	3300	1400	300	0	1100	7200	-
J	-	-	-	-	-	-	-	-	1100	0	-	1100
K	4700	4800	4400	4300	3700	3900	5800	6900	7200	-	0	800
L	-	-	-	-	-	-	-	-	-	1100	800	0

Iterasi 8

AJ=AI+IJ
AJ=4510+1100
AJ=5610

BJ=BI+IJ
BJ=4610+1100
BJ=5710

CJ=CI+IJ
CJ=4210+1100
CJ=5310

KJ=KI+IJ
KJ=7210+1100
KJ=8310

DJ=DI+IJ
DJ=4110+1100
DJ=5210

FJ=FI+IJ
FJ=3300+1100
FJ=4400

GJ=GI+IJ
GJ=1400+1100
GJ=2500

Tabel 17. Matrik Hasil Iterasi 8

	A	B	C	D	E	F	G	H	I	J	K	L
A	0	250	300	400	1000	1200	300	4200	4500	5600	4700	-
B	250	0	550	500	1100	1300	3200	4300	4600	5700	4800	-
C	300	550	0	100	700	900	2800	3900	4200	5300	4400	-
D	400	500	100	0	600	800	2700	3800	4000	5200	4300	-
E	1000	1100	700	600	0	200	2100	3200	3500	4600	3700	-
F	1200	1300	900	800	200	0	1900	3000	3300	4400	3900	-
G	300	3200	2800	2700	200	1900	0	1100	1400	2500	5800	-
H	4200	4300	3900	3800	3200	3000	1100	0	300	1400	6900	-
I	4500	4600	4200	4000	3500	3300	1400	300	0	1100	7200	-
J	5600	5700	5300	5200	4600	4400	2500	1400	1100	0	8300	1100
K	4700	4800	4400	4300	3700	3900	5800	6900	7200	8300	0	800
L	-	-	-	-	-	-	-	-	-	1100	800	0

Iterasi 9

AL=AJ+JL
AL=5610+1100
AL=6710

BL=BJ+JL
BL=5710+1100
BL=6810

CL=CJ+JL
CL=5310+1100
CL=6410

GL=GJ+JL
HL=HJ+JL
IL=IJ+JL

GL=2500+1100
GL=3600

HL=1400+1100
HL=2500

IL=1100+1100
IL=2200

DL=DJ+JL
DL=5210+1100
DL=6310

EL=EJ+JL
EL=4610+1100
EL=5710

FL=FJ+JL
FL=4400+1100
FL=5500

Tabel 18. Matrik hasil iterasi 9

	A	B	C	D	E	F	G	H	I	J	K	L
A	0	250	300	400	1000	1200	300	4200	4500	5600	4700	6700
B	250	0	550	500	1100	1300	3200	4300	4600	5700	4800	6800
C	300	550	0	100	700	900	2800	3900	4200	5300	4400	6400
D	400	500	100	0	600	800	2700	3800	4000	5200	4300	6300
E	1000	1100	700	600	0	200	2100	3200	3500	4600	3700	5700
F	1200	1300	900	800	200	0	1900	3000	3300	4400	3900	5500
G	300	3200	2800	2700	200	1900	0	1100	1400	2500	5800	3600
H	4200	4300	3900	3800	3200	3000	1100	0	300	1400	6900	2500
I	4500	4600	4200	4000	3500	3300	1400	300	0	1100	7200	2200
J	5600	5700	5300	5200	4600	4400	2500	1400	1100	0	8300	1100
K	4700	4800	4400	4300	3700	3900	5800	6900	7200	8300	0	800
L	6700	6800	6400	6300	5700	5500	3600	2500	2200	1100	800	0

Iterasi 10

AL=AK+KL
AL=4700+800
AL=5500

BL=BK+KL
BL=4800+800
BL=5600

CL=CK+KL
CL=4400+800
CL=5200

GL=GJ+JL
GL=2500+1100
GL=3600

HL=HJ+JL
HL=1400+1100
HL=2500

IL=IJ+JL
IL=1100+1100
IL=2200

DL=DK+KL
DL=4300+800
DL=5100

EL=EK+KL
EL=3700+800
EL=4500

FL=FK+KL
FL=3910+800
FL=4710

Tabel 19. Matrik hasil iterasi 10

	A	B	C	D	E	F	G	H	I	J	K	L
A	0	250	300	400	1000	1200	300	4200	4500	5600	4700	5500
B	250	0	550	500	1100	1300	3200	4300	4600	5700	4800	5600
C	300	550	0	100	700	900	2800	3900	4200	5300	4400	5200
D	400	500	100	0	600	800	2700	3800	4000	5200	4300	5100
E	1000	1100	700	600	0	200	2100	3200	3500	4600	3700	4500
F	1200	1300	900	800	200	0	1900	3000	3300	4400	3900	4700
G	300	3200	2800	2700	200	1900	0	1100	1400	2500	5800	3600
H	4200	4300	3900	3800	3200	3000	1100	0	300	1400	6900	2500
I	4500	4600	4200	4000	3500	3300	1400	300	0	1100	7200	2200
J	5600	5700	5300	5200	4600	4400	2500	1400	1100	0	8300	1100
K	4700	4800	4400	4300	3700	3900	5800	6900	7200	8300	0	800
L	5500	5600	5200	5100	4500	4700	3600	2500	2200	1100	800	0

Iterasi 11

AL=AK+KL
AL=4700+800
AL=5500

Tabel 20. Matrik hasil iterasi 11

	A	B	C	D	E	F	G	H	I	J	K	L
A	0	250	300	400	1000	1200	300	4200	4500	5600	4700	5500
B	250	0	550	500	1100	1300	3200	4300	4600	5700	4800	5600
C	300	550	0	100	700	900	2800	3900	4200	5300	4400	5200
D	400	500	100	0	600	800	2700	3800	4100	5200	4300	5100
E	1000	1100	700	600	0	200	200	3200	3500	4600	3700	4700
F	1200	1300	900	800	200	0	1900	3000	3300	4400	3900	4700
G	300	3200	2800	2700	200	1900	0	1100	1400	2500	4400	3600
H	4200	4300	3900	3800	3200	3000	1100	0	300	1400	3300	2500
I	4500	4600	4200	4100	3500	3300	1400	300	0	1100	3000	2200
J	5600	5700	5300	5200	4600	4400	2500	1400	1100	0	1900	1100
K	4700	4800	4400	4300	3700	3900	4400	3300	3000	1900	0	800
L	5500	5600	5200	5100	4500	4700	3600	2500	2200	1100	800	0

Pada kasus ini terdapat 2 kemungkinan jalur:

- Path 1 (A $\Rightarrow$ C $\Rightarrow$ D $\Rightarrow$ F $\Rightarrow$ G $\Rightarrow$ H $\Rightarrow$ I $\Rightarrow$ J $\Rightarrow$ L) = 6710.
- Path 2 (A $\Rightarrow$ C $\Rightarrow$ D $\Rightarrow$ E $\Rightarrow$ K $\Rightarrow$ L) = 5500.

Berdasarkan hasil Iterasi 9 diperoleh jumlah jarak 6710, berarti rute sudah ditemukan dengan jumlah jarak 6710 dengan rute (A $\Rightarrow$ C $\Rightarrow$ D $\Rightarrow$ F $\Rightarrow$ G $\Rightarrow$ H $\Rightarrow$ I $\Rightarrow$ J $\Rightarrow$ L). Karena ada kemungkinan rute lain maka dilakukan iterasi lagi hingga menemukan rute yang lebih optimal yaitu pada hasil iterasi ke 11 dengan hasil jumlah jarak 5500, sedangkan rute terpendek dari alun-alun ke Owabong (A-L) yang dapat ditempuh adalah:

- (A $\Rightarrow$ C $\Rightarrow$ D $\Rightarrow$ E $\Rightarrow$ K $\Rightarrow$ L) Alun-alun $\rightarrow$ Jl. Let. Achmad Nur $\rightarrow$ Jl. Wirasaba $\rightarrow$ Jl. Aw Sumarmo $\rightarrow$ Jl. Raya Bojongsari $\rightarrow$ Jl. Raya Owabong $\rightarrow$ Owabong dengan jumlah panjang jarak yang ditempuh 5500 meter.
- Pada taman reptil terdapat jalur terpendek yaitu: (A $\Rightarrow$ C $\Rightarrow$ D $\Rightarrow$ E $\Rightarrow$ F $\Rightarrow$ G $\Rightarrow$ H) Alun-alun $\rightarrow$ Jl. Let. Achmad Nur $\rightarrow$ Jl. Wirasaba $\rightarrow$ Jl. Banjaransari $\rightarrow$ Jl. Karanglewas $\rightarrow$ Jl. Raya Kutasari $\rightarrow$ Jl. Raya Kutasari $\rightarrow$ Taman Reptil dengan jumlah panjang jarak yang ditempuh 4600 meter.
 - Pada kasus Aquarium Raksasa diperoleh jalur terpendek (A $\Rightarrow$ C $\Rightarrow$ D $\Rightarrow$ E $\Rightarrow$ F $\Rightarrow$ H $\Rightarrow$ I) Alun-alun $\Rightarrow$ Jl. Let. Achmad Nur $\Rightarrow$ Jl. Wirasaba $\Rightarrow$ Jl. Koprul Tanwir $\Rightarrow$ Jl. Sudani $\Rightarrow$ Jl. Raya Padamara $\Rightarrow$ Jl. Raya Purbasari $\Rightarrow$ Aquarium Raksasa, dengan jumlah panjang jarak yang ditempuh 6800 meter.
 - Pada kasus Monumen Jendral Soedirman diperoleh jalur terpendek (A $\Rightarrow$ C $\Rightarrow$ E $\Rightarrow$ F $\Rightarrow$ G $\Rightarrow$ H $\Rightarrow$ I $\Rightarrow$ J $\Rightarrow$ K $\Rightarrow$ L) = 24200, Alun-alun $\Rightarrow$ Jl. Onje $\Rightarrow$ Jl. Dipokusumo $\Rightarrow$ Jl. Pujowiyoto $\Rightarrow$ Jl. Penaruban $\Rightarrow$ Jl. Kaligondang $\Rightarrow$ Jl. Sinduraja $\Rightarrow$ Jl. Pengadegan $\Rightarrow$ Jl. Wlahar $\Rightarrow$ Jl. Bantarbarang $\Rightarrow$ Monumen Jendral Soedirman. Dengan jumlah jarak 24200 meter.
 - Pada kasus Goa Lawa diperoleh jalur terpendek (A $\Rightarrow$ C $\Rightarrow$ D $\Rightarrow$ E $\Rightarrow$ F $\Rightarrow$ G $\Rightarrow$ I $\Rightarrow$ J $\Rightarrow$ K $\Rightarrow$ L) = 25600, Alun-alun $\Rightarrow$ Jl. Let. Achmad Nur $\Rightarrow$ Jl. Wirasaba $\Rightarrow$ Jl.

A.W. Sumarmo $\Rightarrow$ Jl. Bojongsari $\Rightarrow$ Jl. Mangunegara $\Rightarrow$ Jl. Kampung Baru Gandasuli $\Rightarrow$ Jl. Kampung Baru Gandasuli $\Rightarrow$ Jl. Tlagayasa $\Rightarrow$ Jl. Karangreja $\Rightarrow$ Jl. Goalawa . dengan jumlah jarak 25600 meter.

D. Perancangan Aplikasi.

1. *Form* pencarian jalur terpendek. Pada *form* pencarian jalur terpendek ini user bisa mendapatkan informasi jalur terpendek yang telah tersedia, dan tanpa login. *Form* ini disajikan pada Gambar 5.

Gambar 5. *Form* Pencarian Jalur Terpendek.

2. *Form* informasi jalur terpendek. *Form* ini menampilkan informasi jalur terpendek setelah user memilih tujuan yang diinginkan pada *form* pencarian jalur terpendek. *Form* ini disajikan pada Gambar 6.

Gambar 6. *Form* Informasi Jalur Terpendek.

IV. PENUTUP

A. Simpulan

Simpulan dari penelitian ini adalah dihasilkan aplikasi pencarian jalur terpendek menggunakan Algoritma Floyd yang dapat digunakan di kabupaten Purbalingga. Aplikasi ini diharapkan dapat memudahkan wisatawan atau masyarakat dalam menuju lokasi wisata

di Purbalingga dengan mengetahui jalur mana saja yang dilalui, dengan adanya informasi jalur terpendek dan peta jalur terpendek sehingga aplikasi ini lebih informatif.

B. Saran

Pada penelitian ini Algoritma *Floyd* digunakan untuk menentukan jalur terpendek dengan titik awal di alun-alun dan tujuan berbagai lokasi wisata di Purbalingga. Algoritma ini dapat digunakan untuk pencarian jalur terpendek pada objek yang sama namun menggunakan sembarang titik awal dengan tujuan berbagai lokasi wisata yang sama. Oleh karena itu dalam penelitian selanjutnya dapat dikembangkan kembali pencarian jalur terpendek menggunakan algoritma *Floyd* untuk titik awal dari sembarang titik dan objek-objek selain lokasi wisata di Purbalingga.

DAFTAR PUSTAKA

- [1] Andi., 2004, *Dasar Pemrograman Web Dinamis Dengan JSP*, Andi Offset, Yogyakarta.
- [2] Cormen, T. H., Leiserson, C. E., Rivest, R. L., 2001, *Introduction to Algorithms, Second Edition*, MIT Press., Cambridge.
- [3] Ladjamuddin, B., 2004, *Konsep Sistem Basis Data Dan Implementasinya*, Graha Ilmu, Yogyakarta.
- [4] Nugroho, A. 2008, *Algoritma Dan Struktur Data Dalam Bahasa Java*, Andi, Yogyakarta.
- [5] Pressman, R.S., 2001, *Software Engginering*, Mc Graw-Hill Higher Education., New York.
- [6] Riyanto., 2010, *Sistem Informasi Geografi Berbasis Mobile*, http://id.wikipedia.org/wiki/Algoritma_Floyd-Warshall, diakses tanggal 11 Januari 2012.