



**IMPLEMENTASI ALGORITMA *DIJKSTRA* UNTUK
MENENTUKAN RUTE TERPENDEK LOKASI
OBJEK WISATA DI KABUPATEN PATI
BERBASIS *WEB***

Skripsi

diajukan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana
Pendidikan Program Studi Pendidikan Teknik Informatika dan Komputer

Oleh
Raafiud Fauzi Nuuryagandhi NIM.5302411025
UNNES
UNIVERSITAS NEGERI SEMARANG

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS NEGERI SEMARANG**

2016

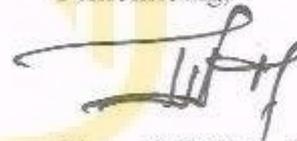
PERSETUJUAN PEMBIMBING

Nama : Raafiud Fauzi Nuuryagandhi
NIM : 5302411025
Program Studi : S-1 Pendidikan Teknik Informatika dan Komputer
Judul Skripsi : IMPLEMENTASI ALGORITMA *DIJKSTRA* UNTUK
MENENTUKAN RUTE TERPENDEK LOKASI OBJEK
WISATA DI KABUPATEN PATI BERBASIS WEB

Skripsi ini telah disetujui oleh pembimbing untuk diajukan ke sidang panitia ujian skripsi Program Studi S-1 Pendidikan Teknik Informatika dan Komputer FT. UNNES

Semarang, 8 September 2016

Pembimbing,



Riana Defi Mahadji Putri, S.T., M.T.

NIP. 197609182005012001

UNNES
UNIVERSITAS NEGERI SEMARANG

PENGESAHAN

Skripsi dengan Judul “Implementasi Algoritma *Dijkstra* Untuk Menentukan Rute Terpendek Lokasi Objek Wisata Di Kabupaten Pati Berbasis *Web*” telah dipertahankan di depan sidang Panitia Ujian Skripsi Fakultas Teknik UNNES pada tanggal 29 bulan September tahun 2016.

Oleh :

Nama : Raafiud Fauzi Nuuryagandhi
NIM : 5302411025
Program Studi : Pendidikan Teknik Informatika dan Komputer

Panitia :

Ketua Panitia



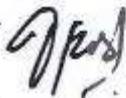
Dr.-Ing. Dhidik Prastiyanto, S.T., M.T.
NIP.197805312005011002

Sekretaris



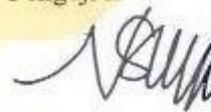
Ir. Ulfah Mediaty Arief, M.T.
NIP.1966005051998022001

Penguji I



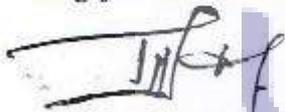
Dr.-Ing. Dhidik Prastiyanto, S.T., M.T.
NIP.197805312005011002

Penguji II



Drs. Said Sunardiyo, M.T.
NIP.196505121991031003

Penguji III/Pembimbing



Riana Defi Mahadji Putri, S.T., M.T.
NIP.197609182005012001

UNNES

UNIVERSITAS NEGERI SEMARANG

Mengetahui,
Dean Fakultas Teknik,



Dr. Nur Qudus, M.T.

NIP.196911301994031001

PERNYATAAN KEASLIAN

Dengan ini saya menyatakan bahwa :

1. Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (sarjana, magister, dan/atau doktor), baik di Universitas Negeri Semarang (UNNES) maupun di perguruan tinggi lain.
2. Karya tulis ini adalah murni gagasan, rumusan dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan Pembimbing dan masukkan Tim Penguji.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya ini, serta sanksi lainnya sesuai dengan norma yang berlaku di perguruan tinggi ini.

Semarang, 8 September 2016
yang membuat pernyataan,

Raafid Fauzi Nuuryagandhi
NIM. 5302411025

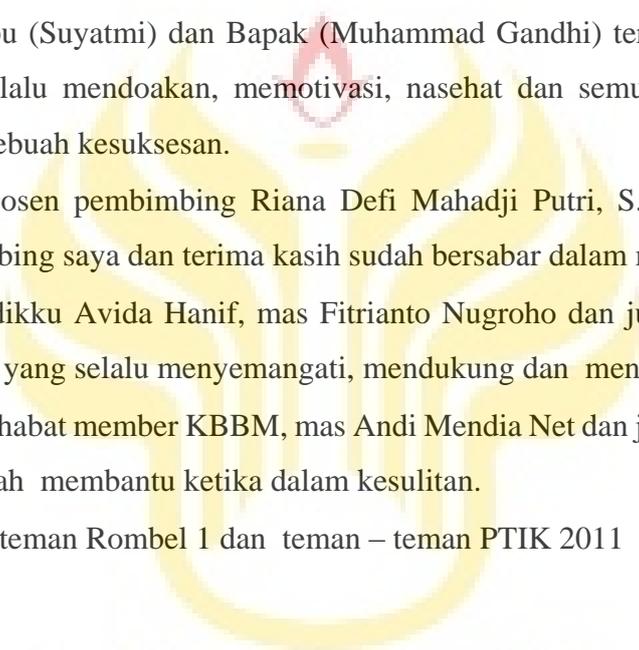
MOTO DAN PERSEMBAHAN

Moto :

Hai orang – orang yang beriman, jadikanlah sabar dan shalat sebagai penolongmu, sesungguhnya Allah beserta orang – orang yang sabar. (QS. Al-Baqarah : 153)

Persembahan :

- Untuk Ibu (Suyatmi) dan Bapak (Muhammad Gandhi) tercinta, terima kasih sudah selalu mendoakan, memotivasi, nasehat dan semua dukungan untuk meraih sebuah kesuksesan.
- Untuk Dosen pembimbing Riana Defi Mahadji Putri, S.T., M.T. berkenan membimbing saya dan terima kasih sudah bersabar dalam membimbing saya.
- Untuk adikku Avida Hanif, mas Fitrianto Nugroho dan juga dik Nurjannatu Na'imah yang selalu menyemangati, mendukung dan mendoakan saya.
- Untuk sahabat member KBBM, mas Andi Mendia Net dan juga teman – teman yang telah membantu ketika dalam kesulitan.
- Teman - teman Rombel 1 dan teman – teman PTIK 2011



UNNES
UNIVERSITAS NEGERI SEMARANG

ABSTRAK

Nuuryagandhi, Raafiud Fauzi. 2016. “Implementasi Algoritma *Dijkstra* Untuk Menentukan Rute Terpendek Lokasi Objek Wisata Di Kabupaten Pati Berbasis *Web*”. Skripsi. Pendidikan Teknik Informatika dan Komputer. Jurusan Teknik Elektro. Fakultas Teknik. Universitas Negeri Semarang.
Pembimbing : Riana Defi Mahadji Putri, S.T., M.T.

Kata Kunci : *Dijkstra*, rute terpendek, objek wisata, *node*, *graf*.

Pariwisata merupakan salah satu yang dapat menjadi ciri khas suatu daerah itu sendiri. Sesuai letak geografis Kabupaten Pati memiliki pelbagai macam kekayaan wisata yang menarik. Guna menunjang pendapatan daerah sektor pariwisata perlu dibutuhkan pemanfaatan teknologi agar para wisatawan mudah untuk mengakses informasi objek wisata, salah satunya adalah pembuatan sistem informasi pencarian rute menuju lokasi objek wisata di Kabupaten Pati.

Algoritma *Dijkstra* merupakan salah satu algoritma untuk menentukan rute terpendek dari lokasi objek wisata satu menuju objek wisata lainnya. Perhitungan algoritma *Dijkstra* diawali dengan menentukan *node-node* yang akan dibentuk menjadi sebuah *graf*. *Node* merupakan objek wisata yang ada di Kabupaten Pati. Algoritma *Dijkstra* menentukan bobot terkecil dari *node* awal menuju *node* akhir dan pada setiap perhitungan dari *node* awal menuju *node* selanjutnya akan diperbarui jika menemukan jarak terpendek. Pengujian *black box* digunakan untuk menguji fungsionalitas menu menu pada sistem, sedangkan untuk uji sistem perhitungan rute terpendek digunakan perbandingan dari hasil perhitungan *google map* dengan hasil perhitungan sistem menggunakan algoritma *Dijkstra*.

Hasil pengujian dari segi fungsionalitas menu sistem pada pengujian *black box* fungsi menu-menu pada sistem bekerja dengan baik. Sedangkan uji sistem algoritma *Dijkstra* dengan *google map* diperoleh hasil perbandingan selisih jarak kurang dari 1% yaitu sebesar 0,43%, sehingga sistem baik untuk digunakan. Dari segi pemanfaatannya sistem algoritma *Dijkstra* dalam menentukan rute terpedek lokasi objek wisata di Kabupaten Pati ini dapat menjadi media untuk promosi dan sebagai informasi rute perjalanan wisatawan menuju objek wisata di Kabupaten Pati.

KATA PENGANTAR

Alhamdulillah rabbil'alamin berkat ridho Allah SWT, semangat dan kerja keras, serta dukungan dari orang tua, kakak, adik dan teman-teman akhirnya penulis dapat menyelesaikan skripsi yang berjudul “Implementasi Algoritma *Dijkstra* Untuk Menentukan Rute Terpendek Lokasi Objek Wisata Di Kabupaten Pati Berbasis *Web*”, ini dapat diselesaikan dan di ajukan untuk memenuhi syarat akhir guna persyaratan memperoleh gelar Sarjana Pendidikan pada Universitas Negeri Semarang. Untuk itu pada kesempatan ini peneliti ingin mengucapkan terima kasih kepada Yth :

1. Dr. Nur Qudus, M.T. selaku Dekan Fakultas Teknik Universitas Negeri Semarang.
2. Dr.-Ing. Dhidik Prastiyanto, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Semarang.
3. Ir. Ulfah Mediaty Arief, M.T. selaku Kaprodi Pendidikan Teknik Informatika dan Komputer Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Semarang.
4. Riana Defi Mahadji Putri, S.T., M.T. selaku dosen pembimbing yang telah meluangkan waktu, tenaga dan pikiran dalam memberikan bimbingan.
5. Kepala Disbudparpora Kabupaten Pati beserta staff yang telah memberikan kesempatan kepada penulis untuk melakukan penelitian.
6. Bapak, Ibu, dan keluarga yang memberikan kasih sayang dan doa sehingga skripsi ini dapat diselesaikan dengan baik.
7. Teman-teman Teknik Elektro khususnya Prodi PTIK yang telah memberikan dukungan dalam penyusunan skripsi.
8. Pihak-pihak terkait yang telah membantu terlaksana dan tersusunnya skripsi ini.

Akhir kata, peneliti berharap semoga penelitian ini dapat bermanfaat bagi peneliti sendiri dan pembaca. Amin.

Semarang, 8 September 2016

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
PERSETUJUAN PEMBIMBING	ii
PENGESAHAN	iii
PERNYATAN KEASLIAN	iv
MOTO DAN PERSEMBAHAN	v
ABSTRAK	vi
KATA PENGANTAR	vii
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian	4
1.5. Manfaat Tugas Akhir	4
1.6. Sistematika Penulisan	5
BAB II KAJIAN PUSTAKA	8
2.1. Kajian Pustaka	8
2.2. Landasan Teori	10
2.2.1. Pengertian Pariwisata	10
2.2.2. <i>Website</i>	11
2.2.3. <i>Web Browser</i>	11
2.2.4. PHP	12
2.2.5. MySQL	14
2.2.6. Algoritma <i>Dijkstra</i>	15
2.2.6.1. Pengertian Algoritma <i>Dijkstra</i>	15

2.2.6.2. Metode Pencarian Jalur terpendek Algoritma Dijkstra	17
2.2.7. Google Maps	21
2.2.8. Xampp	22
BAB III METODE PENELITIAN	24
3.1. Waktu dan Tempat Pelaksanaan	24
3.2. Desain Penelitian	24
3.2.1. Metode Penelitian	
3.2.1.1. Rekayasa Sistem	25
3.2.1.2. Analisis Kebutuhan	26
3.2.1.3. Desain	28
3.2.1.3.1. Permodelan Proses	28
3.2.1.3.2. Desain Antarmuka	31
3.2.1.4. Coding	34
3.2.1.5. Pengujian	34
3.2.1.6. Pemeliharaan	35
3.3. Alat dan Bahan Penelitian	36
3.3.1. Alat Penelitian	36
3.3.2. Bahan Penelitian	36
3.4. Teknik Pengumpulan Data	37
3.4.1. Studi Lapangan	37
3.4.2. Studi Kepustakaan	37
3.5. Teknik Analisis Data	38
BAB IV HASIL DAN PEMBAHASAN	40
4.1. Deskripsi Data	40
4.1.1. Hasil Antarmuka Sistem Informasi	40
4.1.2. Tampilan Peta Objek Wisata di Kabupaten Pati	45
4.1.3. Hasil Penentuan <i>Node</i> dan <i>Graf</i> Objek Wisata	46
4.1.4. Perhitungan Algoritma <i>Dijkstra</i>	51
4.2. Analisis Data	56
4.2.1. Pengujian Algoritma <i>Dijkstra</i> pada <i>web</i> pencari rute	56
4.2.2. Hasil Pengujian <i>Black Box</i>	64

4.3. Pembahasan	66
BAB V PENUTUP	69
5.1. Kesimpulan	69
5.2. Saran	70
DAFTAR PUSTAKA	71
Lampiran	73



DAFTAR TABEL

	Halaman
Tabel 2.1. Perbedaan antara Skripsi yang dilakukan penulis dengan Penelitian terkait	9
Tabel 3.1. Penjelasan Pengujian Sistem	35
Tabel 4.1. Koordinat Objek Wisata Kabupaten Pati	47
Tabel 4.2. Bobot Jarak dan waktu antar <i>node – node</i> objek wisata	48
Tabel 4.3. Perbandingan sistem dengan <i>google map</i> dalam menentukan objek wisata terdekat dari <i>user</i>	57
Tabel 4.4. Hasil Pengujian Algoritma <i>Dijkstra</i> pada <i>web</i> pencarian rute oleh <i>user</i>	58
Tabel 4.5. Perhitungan manual antar objek wisata menggunakan <i>google map</i> .	59
Tabel 4.6. Selisih perhitungan jarak rute menurut <i>google map</i> dan menurut algoritma <i>Dijkstra</i>	60
Tabel 4.7. Hasil Pengujian <i>Black box</i>	64



DAFTAR GAMBAR

	Halaman
Gambar 2.1 Contoh keterhubungan antar titik dalam algoritma <i>Dijkstra</i>	17
Gambar 2.2 Contoh kasus <i>Dijkstra</i> - Langkah 1	18
Gambar 2.3 Contoh kasus <i>Dijkstra</i> - Langkah 2	19
Gambar 2.4 Contoh kasus <i>Dijkstra</i> - Langkah 3	19
Gambar 2.5 Contoh kasus <i>Dijkstra</i> - Langkah 4	20
Gambar 2.6 Contoh kasus <i>Dijkstra</i> - Langkah 5	20
Gambar 3.1 Metode <i>Waterfall</i>	25
Gambar 3.2 <i>Flowchart User</i>	26
Gambar 3.3 <i>Context Diagram</i>	28
Gambar 3.4 DFD Level 1	29
Gambar 3.5 DFD proses pertama	29
Gambar 3.6 DFD Level 2	31
Gambar 3.7 Halaman Utama atau <i>Home</i>	32
Gambar 3.8 Halaman Menu Berita	32
Gambar 3.9 Halaman Menu Objek Wisata	33
Gambar 3.10 Halaman Menu Lokasi Wisata	33
Gambar 3.11 Halaman Menu <i>Login</i>	34
Gambar 4.1 Hasil tampilan antarmuka menu <i>Home</i>	40
Gambar 4.2 Hasil tampilan antarmuka menu <i>Berita</i>	41
Gambar 4.3 Hasil tampilan antarmuka menu <i>Objek Wisata</i>	42
Gambar 4.4 Hasil tampilan antarmuka menu <i>Lokasi Wisata</i>	43
Gambar 4.5 Hasil tampilan antarmuka menu <i>Login</i>	44
Gambar 4.6 Tampilan Peta Objek Wisata di Kabupaten Pati	45
Gambar 4.7 <i>Graf</i> Objek Wisata di Kabupaten Pati	46
Gambar 4.8 Jaringan Objek Wisata di Kabupaten Pati	51
Gambar 4.9 <i>Node</i> awal berbobot 0	52
Gambar 4.10 <i>Node</i> bobot dari 2 ke 3, 4, 14	53
Gambar 4.11 <i>Node</i> ke 4 dari 3	54
Gambar 4.12 <i>Node</i> dari 4 ke 8, 6, 10, 13	55

Gambar 4.13 Jalur rute terpendek dari 1 ke 6	56
Gambar 4.14 Rute dari Air Terjun Santi - JWF (<i>Juwana Water Fantasi</i>)	61
Gambar 4.15 Rute dari Gunung Rowo Indah - Gua Wareh	62
Gambar 4.16 Rute dari Tadah Hujan - Air Terjun Grinjingan Sewu dan Air Terjun Sepletus	63



LAMPIRAN

Halaman

Lampiran 1	Surat Usulan Topik Skripsi	74
Lampiran 2	Surat Usulan Dosen Pembimbing Skripsi	73
Lampiran 3	Surat Keputusan Dosen Pembimbing Skripsi	76
Lampiran 4	Surat Permohonan Penelitian LITBANG Kabupaten Pati	77
Lampiran 5	Surat Rekomendasi LITBANG Kabupaten Pati	78
Lampiran 6	Surat Selesai Penelitian di DISBUDPARPORA Kabupaten Pati ..	79
Lampiran 7	<i>Source Code</i>	80
Lampiran 8	Gambar Objek Wisata Kabupaten Pati	92
Lampiran 9	Gambar Peta Objek Wisata di Kabupaten Pati pada Sistem Pencarian Rute Terpendek	99
Lampiran 10	Instruen Observasi	100
Lampiran 11	Instrumen Wawamcara	101



BAB I

PENDAHULUAN

1.1. Latar Belakang

Pariwisata merupakan salah satu yang dapat menjadi ciri khas suatu daerah itu sendiri. Menurut Undang Undang No 10 tahun 2009 tentang Kepariwisata, pariwisata adalah berbagai macam kegiatan wisata dan didukung berbagai fasilitas serta layanan yang disediakan masyarakat, pengusaha, Pemerintah dan Pemerintah daerah. Dengan adanya pariwisata mampu menambah pendapatan ekonomi daerah. Dari banyaknya objek wisata yang sering dikunjungi oleh para wisatawan asing dan juga domestik dapat menjadi sumber pendapatan daerah tersebut. Guna menunjang pendapatan daerah sektor pariwisata perlu dibutuhkan pemanfaatan teknologi agar para wisatawan mudah untuk mengakses informasi objek wisata.

Secara geografis Kabupaten Pati terletak di 644'56,80" LS 11102'06,96" BT dengan luas wilayah keseluruhan 1.419,07 km yang terbagi menjadi 21 Kecamatan dan 405 Desa yang berbatasan langsung dengan Kabupaten Jepara dan Laut Jawa di sebelah utara, Kabupaten Grobogan dan Kabupaten Blora di sebelah selatan, Kabupaten Kudus dan Kabupaten Jepara di sebelah barat, serta Kabupaten Rembang dan Laut Jawa di sebelah timur. Sesuai letak geografis yang ada Kabupaten Pati memiliki pelbagai macam kekayaan wisata yang menarik.

Tempat – tempat objek wisata yang ada di Kabupaten Pati tersebar diberbagai daerah Kabupaten Pati. Untuk menuju tempat objek wisata, ada beberapa rute yang bisa ditempuh. Wisatawan pastinya menginginkan rute yang

paling efisien untuk menuju objek wisata tujuan sehingga dapat menghemat waktu dan biaya. Tentunya masih banyak para wisatawan tidak mengetahui rute untuk mengakses tempat wisata di Kabupaten Pati.

Kesulitan menentukan jarak terpendek timbul karena terdapat banyak jalur yang ada pada tiap daerah karena pada kenyataannya dari daerah A ke daerah B tidak hanya memiliki satu jalur saja, banyak sekali jalur yang dapat dilalui sehingga terbentuk suatu jaringan. Untuk membantu dalam menentukan jarak terpendek dapat digunakan peta konvensional dan memilih mana jalur yang dianggap terpendek dari daerah asal ke daerah tujuan. Namun hal ini dirasa kurang maksimal dan memperlambat waktu karena harus memilih sendiri dari banyak jalur yang ada dan melakukan perhitungan sendiri mana kira-kira jarak terpendek dari daerah asal menuju daerah tujuan yang dikehendaki (Ardiani, 2011).

Proses penghitungan rute terpendek adalah proses mencari jarak terpendek atau biaya terkecil suatu rute dari *node* awal ke *node* tujuan dalam sebuah jaringan. Pada proses penghitungan rute terpendek terdapat dua macam proses yaitu proses pemberian label dan proses pemeriksaan *node*. Metode pemberian label adalah metode untuk memberikan identifikasi pada setiap *node* dalam jaringan (Purwananto, dkk, 2005:95). Dalam menentukan lintasan rute terpendek dapat menggunakan Algoritma. Beberapa algoritma yang dapat digunakan untuk menyelesaikan penentuan rute terpendek adalah *Dijkstra's Algorithm*, *Bellman-Ford's Algorithm*, *A* Search Algorithm*, dan *Floyd-Warshall Algorithm* (Sanan, dkk., 2013).

Salah satu cara untuk dapat menentukan jalur terpendek adalah dengan menginterpretasikan peta kedalam suatu *graf*. Dalam *graf*, terdapat metode yang dapat digunakan untuk menentukan jarak terpendek. Salah satu metode yang digunakan untuk pencarian jalur terpendek adalah Algoritma *Dijkstra*. Algoritma ini digunakan dalam *graf* berarah dimana setiap titik dihubungkan oleh sisi yang memiliki bobot. Dengan memperhitungkan bobot pada setiap sisi, algoritma ini dapat digunakan untuk menentukan jalur terpendek dari suatu titik ke titik akhir tujuan (Puspika, dkk., 2012). Algoritma *Dijkstra* lebih intensif dalam komputasi untuk pencarian jalur optimum dalam suatu jaringan seperti internet, dan waktu rata-rata eksekusi algoritma *Dijkstra* lebih kecil disbanding algoritma *Ant Colony*, maka algoritma *Dijkstra* banyak digunakan dalam pencarian jalur optimum pada jaringan internet dibanding algoritma lain (Gusmão, dkk., 2013:125).

Berdasarkan latar belakang yang telah diuraikan diatas, maka pokok permasalahan dalam penelitian ini adalah dibutuhkannya sebuah sistem informasi guna memudahkan prawisatawan dapat mengetahui lokasi dan informasi objek wisata yang ada di Kabupaten Pati sekaligus mengetahui rute terpendek menuju lokasi objek wisata menggunakan Algoritma *Dijkstra* berbasis *Web*.

1.2. Rumusan Masalah

Berdasarkan pokok permasalahan diatas, penulis menggali rumusan masalah dari penelitian ini adalah sebagai berikut :

1. Bagaimana membuat sistem informasi yang dapat memudahkan wisatawan dalam mencari informasi objek wisata di Kabupaten Pati?

2. Bagaimana wisatawan menentukan rute terpendek dari objek wisata satu ke objek wisata lainnya?
3. Bagaimana menerapkan Algoritma *Dijkstra* dalam menentukan rute terpendek objek wisata di Kabupaten Pati?
4. Bagaimana wisatawan mengetahui objek wisata terdekat di Kabupaten Pati dari posisi dimana mereka berada?

1.3. Batasan Masalah

Dalam penyusunan skripsi ini, permasalahan yang akan dibahas dibatasi pada :

- a. Objek wisata merupakan titik (*node*) pada *graf* yang ada pada pencarian rute terpendek.
- b. Perancangan sistem dilakukan dengan menggunakan bahasa pemrograman PHP dan MySQL.
- c. Hanya akan menampilkan satu rute terpendek.
- d. Pencarian rute terpendek menggunakan Algoritma *Dijkstra*.
- e. Ruang lingkup pengambilan data dan survei dilakukan di DINBUDPARPORA Kabupaten Pati.

1.4. Tujuan Penelitian

Sesuai dengan rumusan masalah yang dikemukakan diatas, maka tujuan penelitian ini adalah antara lain :

1. Membuat sitem informasi untuk memudahkan wisatawan mengetahui objek wisata dan rute terpendek dari tujuan objek wisata awal ke objek wisata selanjutnya di Kabupaten Pati.
2. Memudahkan wisatawan untuk mengetahui objek wisata terdekat dari posisi dimana wisatawan berada.
3. Menerapkan Algoritma *Dijkstra* dalam menentukan pencarian rute terpendek objek wisata di Kabupaten Pati

1.5. Manfaat Penelitian

Adapun manfaat yang diharapkan dari pelaksanaan penelitian ini adalah sebagai berikut :

a. Bagi Penulis

1. Melengkapi tugas dan syarat kelulusan Strata-1 Prodi Pendidikan Teknik Informatika dan Komputer Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Semarang.
2. Dapat menambah wawasan dan pengetahuan tentang masalah pencarian rute terpendek.
3. Dapat mengetahui cara kerja pencarian rute terdekat menggunakan metode Algoritma *Dijkstra*.

b. Bagi Universitas Negeri Semarang

1. Sebagai bahan referensi untuk nantinya dapat dijadikan sebagai acuan pada penelitian yang selanjutnya.
2. Dapat dijadikan tolak ukur keberhasilan akademik dalam mendidik dan memberikan ilmu pengetahuan.

c. Bagi Para Pengguna

1. Mempermudah wisatawan untuk mencari lokasi tempat pariwisata terdekat di Kabupaten Pati.
2. Mempersingkat waktu dan menghemat pengeluaran wisatawan.

1.6 Sistematika Penulisan

Secara garis besar skripsi ini dibagi menjadi tiga bagian, yaitu bagian awal skripsi, bagian pokok skripsi dan bagian akhir skripsi.

Bagian awal skripsi meliputi Halaman Sampul, Halaman Judul, Halaman Pengesahan, Motto dan Persembahan, Kata Pengantar, Abstrak, Daftar Isi, Daftar Gambar, Daftar Tabel dan Daftar Lampiran.

Bagian pokok skripsi secara garis besar terdiri dari lima bab, yaitu:

BAB I PENDAHULUAN

Di dalam bab ini dikemukakan latar belakang, permasalahan, pembatasan masalah, tujuan, manfaat, dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini berisi tentang beberapa dasar teori yang dijadikan acuan dalam perancangan dan pembuatan sistem, penelitian dahulu serta kerangka berfikir.

BAB III METODE PENELITIAN

Di dalam bab ini dikemukakan metode penelitian yang berisi langkah-langkah yang harus ditempuh untuk menyelesaikan permasalahan, pengambilan data, pemodelan jaringan dan

penyelesaian model matematis, pengembangan sistem serta perancangan pengujian.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

Di dalam bab ini dikemukakan tentang hasil dari perancangan dan pembuatan sistem. Pada bab ini juga berisi tentang hasil pengujian sistem dimana dalam penelitian ini menggunakan metode pengujian *black box*.

BAB V PENUTUP

Di dalam bab ini dikemukakan simpulan dari pembahasan hasil penelitian dan saran.

Bagian akhir skripsi meliputi Daftar Pustaka dan Lampiran-lampiran yang mendukung penulisan skripsi.

BAB II KAJIAN PUSTAKA

2.1. Kajian Pustaka

Penelitian dalam bidang pencarian rute terpendek sudah banyak dikembangkan sehingga menghasilkan banyak produk yang baru dan bermanfaat bagi pengguna produk tersebut. Salah satunya dalam penelitian, yaitu Shaga Bogas Priatmoko dalam jurnalnya yang berjudul “Algoritma *Dijkstra* Untuk Pencarian Jalur Terdekat Dan Rekomendasi Objek Pariwisata Di Pulau Bali” (Priatmoko, 2014) dan penelitian Diana Okta Pugas dkk yang berjudul “Pencarian Rute Terpendek Menggunakan Algoritma *Dijkstra* Dan Astar (A*) Pada SIG berbasis *Web* untuk Pemetaan Pariwisata Kota Sawahlunto” (Pugas dkk., 2011), yang menghasilkan aplikasi untuk membantu memudahkan wisatawan dalam mencari lokasi pariwisata. Untuk perbandingan penelitian skripsi yang dilakukan penulis terhadap penelitian terkait dapat dilihat pada tabel 2.1.

Setiap wisatawan menginginkan kemudahan dalam mencari lokasi objek wisata. Masing – masing wisatawan juga memiliki kriteria dalam mencari objek wisata. Untuk itu perlu dibangun sistem informasi pencarian rute terdekat yang akan membantu para wisatawan dalam pencarian rute dan lokasi objek wisata. Dengan metode tersebut, diharapkan akan mempermudah pencarian rute dan lokasi objek wisata. Sehingga wisatawan akan mampu menentukan objek wisata secara cepat, tepat dan efisien berdasarkan jarak dan waktu tempuh menuju objek wisata yang diinginkan.

Tabel 2.1. Perbedaan antara Skripsi yang dilakukan penulis dengan Penelitian terkait.

Klasifikasi	Penelitian Skripsi (Raafiud, 2016)	Jurnal Penelitian (Priatmoko, 2014)	Jurnal Penelitian (Pugas, dkk, 2011)
Algoritma	<i>Algoritma Dijkstra</i>	<i>Algoritma Dijkstra</i>	<i>Algoritma Dijkstra dan Algoritma Astar (A*)</i>
Aplikasi	Berbasis <i>Website</i>	Berbasis <i>Web Desktop</i>	Berbasis <i>Website</i>
Website	Penggunaan <i>website</i> pada browser komputer atau laptop dan browser internet Android.	Penggunaan <i>website</i> pada browser komputer atau laptop.	Penggunaan <i>website</i> pada browser komputer atau laptop.
Map yang digunakan	<i>Google map</i>	<i>Google map</i>	Map <i>Server SIG</i>
Aplikasi Pada Pencarian Rute terdekat	<ul style="list-style-type: none"> - Pemilihan posisi awal dan posisi akhir objek wisata secara manual maupun secara otomatis. - Terdapat <i>derection</i> rute. - Tampilan pencarian rute, map, dan <i>derection</i> rute dalam satu <i>page</i> halaman. - Dapat menentukan posisi objek wisata terdekat. 	<ul style="list-style-type: none"> - Pemilihan posisi awal dan posisi akhir objek wisata secara manual. - <i>Derection</i> rute tidak ada. - Tampilan pencarian rute, map, dan <i>derection</i> rute dalam satu <i>page</i> halaman. - Tidak dapat menentukan posisi objek wisata terdekat. 	<ul style="list-style-type: none"> - Pemilihan posisi awal dan posisi akhir objek wisata secara manual. - Terdapat <i>derection</i> rute. - Tampilan pencarian rute, map, dan <i>derection</i> rute terpisas dalam <i>page</i> halaman. - Tidak dapat menentukan posisi objek wisata terdekat.
Bahasa dan software yang digunakan	PHP, Xampp, MySQL, <i>Google map</i> ,	PHP, MySQL, <i>Google map</i>	PHP, MS4W (Map <i>Server</i>), PostGIS, PostgreSOL
Target atau <i>user</i> yang dituju	Umum, wisatawan lokal maupun luar daerah	Biro jasa Pariwisata di Bali	Umum, wisatawan lokal maupun luar daerah

Kelebihan dan Kekurangan	<ul style="list-style-type: none"> - Hanya menggunakan satu Algoritma (<i>Algoritma Dijkstra</i>). - Dapat memilih lokasi awal dan akhir objek wisata secara otomatis - Dapat menentukan objek wisata terdekat. - Terdapat <i>derection</i> rute. - Pemilihan lokasi objek wisata, Map dan <i>derection</i> rute dalam satu <i>page</i> halaman. - Dapat digunakan pada <i>mobile browser</i>. 	<ul style="list-style-type: none"> - Hanya menggunakan satu Algoritma (<i>Algoritma Dijkstra</i>). - Tidak dapat memilih lokasi secara otomatis. - Tidak dapat menentukan objek wisata terdekat. - Tidak terdapat <i>derection</i> rute. - Pemilihan lokasi onjek wisata dan Map dalam satu <i>page</i> halaman. - <i>Website</i> belum bisa digunakan pada <i>mobile browser</i>. 	<ul style="list-style-type: none"> - Dapat milih algoritma <i>Dijkstra</i> atau A* yang dapat digunakan - Tidak dapat memilih lokasi secara otomatis. - Tidak dapat menentukan objek wisata terdekat. - Terdapat <i>derection</i> rute. - Antara pemilihan lokasi rute dengan hasil <i>derection</i> rute dan map terpisah dalam satu <i>page</i> halaman. - <i>Website</i> belum bisa digunakan pada <i>mobile browser</i>.
--------------------------	--	--	--

2.2. Landasan Teori

2.2.1. Pengertian Pariwisata

Pariwisata atau turisme adalah suatu perjalanan yang dilakukan untuk rekreasi atau liburan, dan juga persiapan yang dilakukan untuk aktivitas ini. Seorang wisatawan atau turis adalah seseorang yang melakukan perjalanan paling tidak sejauh 80 km (50 mil) dari rumahnya dengan tujuan rekreasi, merupakan definisi oleh Organisasi Pariwisata Dunia. Menurut Undang Undang No. 10/2009 tentang Kepariwisataan, yang dimaksud dengan pariwisata adalah berbagai macam kegiatan wisata dan didukung berbagai fasilitas serta layanan yang disediakan masyarakat, pengusaha, Pemerintah dan Pemerintah Daerah.

2.2.2. Website

Website atau biasa dibilang situs merupakan kumpulan dari berbagai macam data yang bisa berupa gambar, suara, ataupun video yang terhubung dan saling terkait satu sama yang lainnya. *Website* awalnya merupakan suatu layanan sajian informasi yang menggunakan konsep *hyperlink*, yang memudahkan *user* atau pengguna dalam memperoleh informasi. Berdasarkan sifatnya *website* dibedakan menjadi dua jenis, yaitu *website* statis dan *website* dinamis. *Website* statis merupakan *website* yang isi informasinya bersifat tetap, jarang update dan isi informasinya searah dari pemiliknya. Contoh dari *website* statis adalah *website company profile* dan *website* profil organisasi. Sedangkan *website* dinamis isi webnya selalu berubah - berubah, *update* dan interaksinya bersifat interaktif dari pemilik *website* maupun pengguna atau *user website*. Contoh dari *website* dinamis antara lain adalah *facebook*, *olx*, *detik*, *wikipedia*. Faktor yang menjadikan sebuah *website* menjadi dinamis adalah *Content Management System* atau CMS yang memiliki akses ke administrator secara langsung sehingga dapat memutakhirkan isinya dengan sangat mudah.

2.2.3. Web Browser

Web Browser adalah perangkat lunak yang berguna untuk mengakses informasi *web* ataupun untuk melakukan transaksi via *web*. Awalnya *web browser* berorientasi pada teks dan belum dapat menampilkan gambar. Akan tetapi sekarang ini dengan dikembangkannya *web browser* tidak hanya mampu menampilkan gambar dan teks saja, tetapi juga dapat memutar file multimedia berupa foto, video,

maupun audio. Beberapa contoh *web browser* yang ada saat ini seperti : *Google Chrome, Internet Explorer (IE), Mozilla Firefox, Opera, Netscape, dan Safari.*

Web Browser pertama bernama *World Wide Web* yang diciptakan Tim Berners-Lee, kemudian diubah menjadi *Nexus*. Pada tahun 1993, Marc Andreessen melakukan inovasi *web browser* dengan membuat *Mosaic* yang sekarang di kenal dengan *Netscape*. Hal tersebut membuat sistem internet menjadi mudah untuk digunakan dan mudah untuk di akses oleh banyak orang. Pada tahun 1994 *Netscape* ini menjadi *browser* terpopuler di dunia karena menguasai 90% dari seluruh pengguna *web browser* dunia.

Pada tahun 1995, *Microsoft* menciptakan *Internet Explorer* dan pada tahun 2002 mendominasi di pasar *web browser* dunia dengan lebih dari 95% pengguna. Kemudian pada tahun 1998 *Netscape* meluncurkan kembali yang menjadi *Mozilla Firefox*. Lalu pada tahun 2003 *Apple* mengeluarkan *web browser* yaitu *Safari*, dan *Google Chrome* menjadi pendatang baru di pasar *web browser* yang dirilis pada September 2008.

2.2.4. PHP

PHP merupakan bahasa *server-side scripting* yang menyatu dengan HTML (*Hypertext Markup Language*) untuk membuat halaman *web* yang dinamik. Artinya semua sintaks yang diberikan akan sepenuhnya dijalankan pada *server* sedangkan yang dikirimkan ke *browser* hanya hasilnya saja. PHP dapat mengirim HTTP *header*, dapat mengeset *cookies*, mengatur *authentication* dan *redirect users* (Nugroho, 2004).

PHP dikembangkan dan pertama kali dibuat pada tahun 1995 oleh Rasmus Lerdorf. Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs Personal). Sesuai dengan namanya, PHP digunakan untuk membuat *website* pribadi. Sebelum dikenal dengan PHP pada waktu itu bernama FI (*From Interpretet*) yang berwujud sekumpulan skrip yang digunakan untuk mengolah data formulir.

Menurut Rouf dkk dalam skripsi Darul Ashar (2014:64) PHP memiliki keunggulan sebagai berikut :

1. PHP adalah bahasa pemrograman *web* yang mudah dipahami, karena banyak referensi yang dapat digunakan dengan mudah.
2. PHP dapat diakses dengan kecepatan yang tinggi dan pembuatannya relatif mudah.
3. PHP bersifat kompatibel dengan *web server* yang sudah ada dan dapat juga berjalan dengan baik walaupun sistem operasi yang dipakai berbeda baik pada *server* ataupun klien.
4. PHP dapat diperoleh secara gratis.
5. PHP juga dapat bekerja pada *Microsoft Personal Web Server, Apache, IIS, Xitami*, dan sebagainya.
6. Bahasa PHP dapat diletakkan atau disisipkan pada tag HTML (bersifat *embedded*).
7. PHP termasuk *server-side programming*.

2.2.5. MySQL

MySQL merupakan *software* manajemen *databasae* (*Database Management System* - DBMS) yang *Open Source* (gratis) yang sangat populer dikalangan pemrograman *web*, sehingga dapat digunakan untuk membuat aplikasi *web* yang mampu menampung *database* sebagai sumber dan data pengolahan datanya. Hal ini dikarenakan MySQL dapat digunakan secara cepat yang mampu menangani banyak *query* dengan kecepatan tinggi, dan dapat mencukupi untuk kebutuhan *database* perusahaan skala menengah kecil. MySQL mampu menampung *database* dengan sekala besar, dengan jumlah *record* lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris dengan batas indeks yang dapat ditampung mencapai 32 indek pada setiap tabelnya.

MySQL dikembangkan oleh David Axmark, Allan Larsson dan Michael "Monty" Widenius yang memiliki sebuah perusahaan pengembang *software* dan konsultan *database* di Swedia pada tahun 1994 bernama TcX Data KonsultanAB. Tujuan awal dikembangkan MySQL adalah untuk mengembangkan aplikasi berbasis *web* pada *client*.

Menurut Achmad Sholicin (2010:8) fitur – fitur MySQL antara lain :

- *Relational Database System*. Seperti halnya *software database* lain yang ada di pasaran, MySQL termasuk RDBMS.
- *Arsitektur Client-Server*. MySQL memiliki arsitektur *client-server* dimana *server database* MySQL terinstal di *server*. *Client* MySQL dapat berada di komputer yang sama dengan *server*, dan dapat juga di komputer lain yang berkomunikasi dengan *server* melalui jaringan bahkan internet.

- Mengenal perintah SQL standar. SQL (*Structured Query Language*) merupakan suatu bahasa standar yang berlaku di hampir semua *software database*. MySQL mendukung SQL versi SQL:2003.
- Mendukung *Sub Select*. Mulai versi 4.1 MySQL telah mendukung *select* dalam *select (sub select)*.
- Mendukung *Views*. MySQL mendukung *views* sejak versi 5.0
- Mendukung *Stored Prosedured (SP)*. MySQL mendukung SP sejak versi 5.0
- Mendukung *Triggers*. MySQL mendukung *trigger* pada versi 5.0 namun masih terbatas. Pengembang MySQL berjanji akan meningkatkan kemampuan *trigger* pada versi 5.1.
- Mendukung *replication*.
- Mendukung *foreign key*.

2.2.6. Algoritma Dijkstra

2.2.6.1. Pengertian Algoritma Dijkstra

Algoritma *Dijkstra* merupakan salah satu algoritma yang efektif dalam memberikan lintasan terpendek dari suatu lokasi ke lokasi yang lainnya. Prinsip dari algoritma *Dijkstra* adalah mencari titik lokasi dengan pencarian dua lintasan yang paling pendek. Pada setiap iterasi, jarak titik yang diketahui (dari titik awal) diperbaharui bila ternyata didapat titik yang baru yang memberikan jarak terpendek. Syarat algoritma ini adalah bobot sisinya yang harus non-negatif. (Satyananda, 2012:46).

Algoritma *Dijkstra* ditemukan oleh *Edsger Wybe Dijkstra* pada tahun 1959.

Algoritma *Dijkstra* termasuk dalam algoritma prinsip *greedy*, dimana pada setiap bobot minimum menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih.

Algoritma *Dijkstra* diterapkan untuk mencari rute terpendek pada *graf* berarah. Algoritma *Dijkstra* dapat mencari rute terpendek dalam sejumlah langkah.

Dibawah ini adalah *pseudocode* dari Algoritma *Dijkstra* :

procedure Dijkstra (input *m*: matriks, *a*: simpul awal) {Mencari lintasan terpendek dari simpul awal *a* ke semua simpul lainnya. Masukan: matriks (*m*) dari *graf* berbobot *G* dan simpul awal *a*. Keluaran: lintasan terpendek dari *a* ke semua simpul lainnya }

Deklarasi

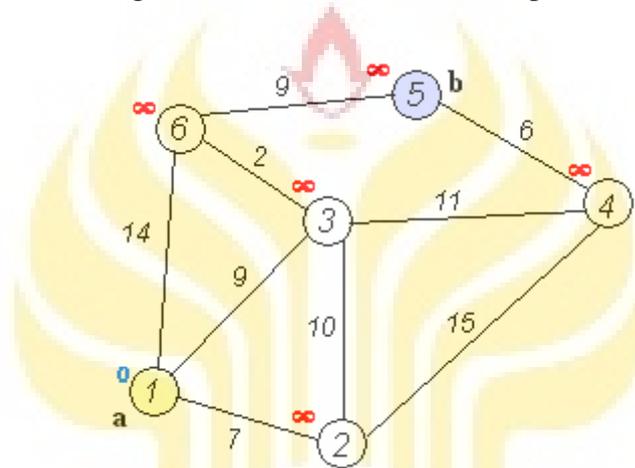
```

function Dijkstra(Graph, source):
    for each vertex v in Graph:
        dist[v] := infinity ;
        previous[v] := undefined ;
    end for
    dist[source] := 0 ;
    Q := the set of all grafs in Graph ;
    while Q is not empty:
        u := vertex in Q with smallest distance in
        dist[];
        remove u from Q ;
        if dist[u] = infinity:
            break ;
        end if
        for each neighbor v of u:
            alt := dist[u] + dist_between(u, v) ;
            if alt < dist[v]:
                dist[v] := alt ;
                previous[v] := u ;
                decrease-key v in Q;
            end if
        end for
    end while
    return dist;
endfunction

```

2.2.6.2. Metode Pencarian Jalur terpendek Algoritma *Dijkstra*

Algoritma ini bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Misalkan titik menggambarkan gedung dan garis menggambarkan jalan, maka algoritma *Dijkstra* melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik.



Gambar 2.1 Contoh keterhubungan antar titik dalam algoritma *Dijkstra*

Pertama-tama tentukan titik mana yang akan menjadi *node* awal, lalu beri bobot jarak pada *node* pertama ke *node* terdekat satu per satu, *Dijkstra* akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap. Inilah urutan logika dari algoritma *Dijkstra*:

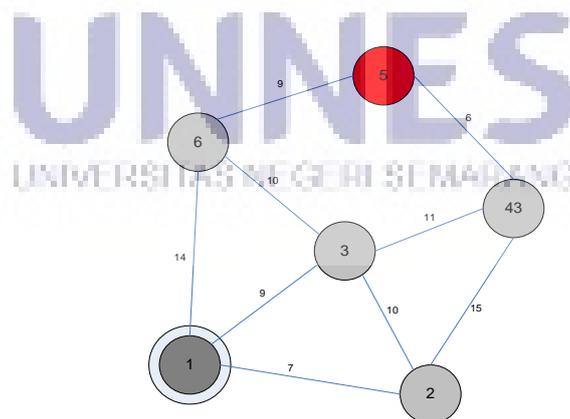
1. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada *node* awal dan nilai tak hingga terhadap *node* lain (belum terisi)
2. Set semua *node* “Belum terjamah” dan set *node* awal sebagai “*Node* keberangkatan”
3. Dari *node* keberangkatan, pertimbangkan *node* tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan. Sebagai contoh, jika titik

keberangkatan A ke B memiliki bobot jarak 6 dan dari B ke *node* C berjarak 2, maka jarak ke C melewati B menjadi $6+2=8$. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.

4. Saat kita selesai mempertimbangkan setiap jarak terhadap *node* tetangga, tandai *node* yang telah terjamah sebagai “*Node* terjamah”. *Node* terjamah tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
5. Set “*Node* belum terjamah” dengan jarak terkecil (dari *node* keberangkatan) sebagai “*Node* Keberangkatan” selanjutnya dan lanjutkan dengan kembali ke *step* 3

Dibawah ini penjelasan langkah per langkah pencarian jalur terpendek secara rinci dimulai dari *node* awal sampai *node* tujuan dengan nilai jarak terkecil.

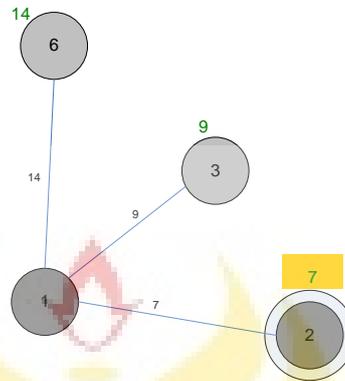
1. *Node* awal 1, *Node* tujuan 5. Setiap *edge* yang terhubung antar *node* telah diberi nilai.



Gambar 2.2 Contoh kasus *Dijkstra* - Langkah 1

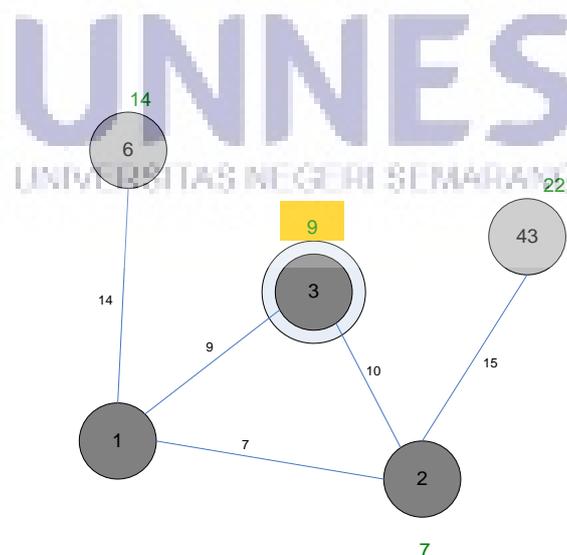
2. *Dijkstra* melakukan kalkulasi terhadap *node* tetangga yang terhubung langsung dengan *node* keberangkatan (*node* 1), dan hasil yang didapat adalah *node* 2

karena bobot nilai *node 2* paling kecil dibandingkan nilai pada *node* lain, nilai $= 7 (0+7)$.



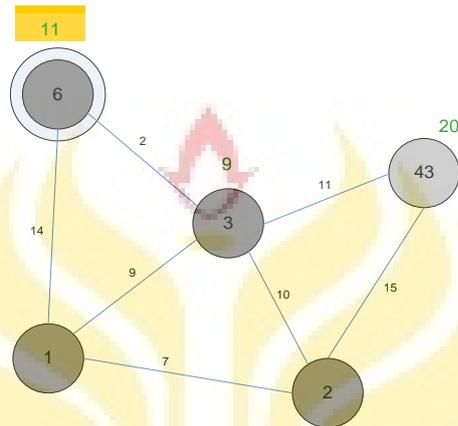
Gambar 2.3 Contoh kasus *Dijkstra* - Langkah 2

3. *Node 2* diset menjadi *node* keberangkatan dan ditandai sebagai *node* yang telah terjamah. *Dijkstra* melakukan kalkulasi kembali terhadap *node-node* tetangga yang terhubung langsung dengan *node* yang telah terjamah. Dan kalkulasi *Dijkstra* menunjukkan bahwa *node 3* yang menjadi *node* keberangkatan selanjutnya karena bobotnya yang paling kecil dari hasil kalkulasi terakhir, nilai $9 (0+9)$.



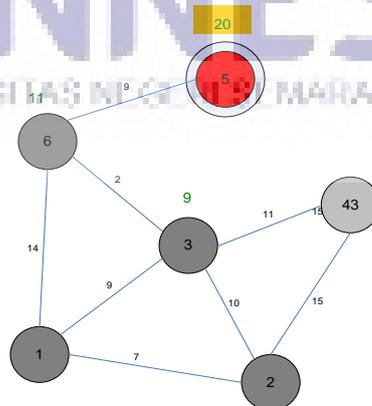
Gambar 2.4 Contoh kasus *Dijkstra* - Langkah 3

4. Perhitungan berlanjut dengan *node* 3 ditandai menjadi *node* yang telah terjamah. Dari semua *node* tetangga belum terjamah yang terhubung langsung dengan *node* terjamah, *node* selanjutnya yang ditandai menjadi *node* terjamah adalah *node* 6 karena nilai bobot yang terkecil, nilai 11 ($9+2$).



Gambar 2.5 Contoh kasus *Dijkstra* - Langkah 4

5. *Node* 6 menjadi *node* terjamah, *Dijkstra* melakukan kalkulasi kembali, dan menemukan bahwa *node* 5 (*node* tujuan) telah tercapai lewat *node* 6. Jalur terpendeknya adalah 1-3-6-5, dan nilai bobot yang didapat adalah 20 ($11+9$). Bila *node* tujuan telah tercapai maka kalkulasi *Dijkstra* dinyatakan selesai.



Gambar 2.6 Contoh kasus *Dijkstra* - Langkah 5

2.2.8. *Google maps*

Google maps adalah sebuah jasa peta *globe virtual* gratis dan *online* disediakan oleh *Google* dapat ditemukan di <http://maps.google.com/>. *Google maps* menawarkan peta yang dapat diseret dan gambar satelit untuk seluruh dunia dan baru-baru ini, dan juga menawarkan perencanaan rute dan pencari letak bisnis di U.S., Kanada, Jepang, Hong Kong, Cina, UK, Irlandia (hanya pusat kota) dan beberapa bagian Eropa.

Google maps diperkenalkan pada Februari 2005 dan merupakan revolusi bagaimana peta dalam *web*, yaitu dengan membirkan *user* untuk menarik peta sehingga dapat menavigasinya. Solusi peta ini pada saat itu masih baru dan membutuhkan *server* khusus. Beberapa saat setelahnya, ada yang berhasil *hack Google Maps* mengambil kesimpulan bahwa mereka membutuhkan API dan pada bulan Juni 2005, *Google Maps API* dirilis secara publik.

Google Maps API adalah suatu *library* berbentuk *JavaScript*. Cara membuat *Google Maps* untuk ditampilkan pada suatu *web* atau *blog* sangatlah mudah hanya dengan membutuhkan pengetahuan mengenai HTML serta *JavaScript*, serta koneksi internet yang stabil. Dengan menggunakan *Google Maps API*, dapat menghemat waktu dan biaya untuk membangun aplikasi pada peta digital yang handal, sehingga yang difokuskan hanya pada data-data yang akan ditampilkan. Dengan kata lain, kita hanya memanfaatkan fasilitas dari *Google Maps* dengan menggunakan satu data dan peta yang ditampilkan adalah milik *Google* sehingga tidak dipungsingkan dengan membuat peta suatu lokasi, bahkan dunia.

2.2.9. Xampp

Menurut Yogi Wicaksono (2008:7) “XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data MySQL dikomputer *local*”. XAMPP berperan sebagai *server web* pada komputer anda. XAMPP juga dapat disebut sebuah *CPanel server virtual*, yang dapat membantu anda melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus online atau terakses dengan internet.

Xampp sendiri terdiri dari beberapa program yang berada didalamnya, sebagai berikut :

1. *Apache HTTP Server*

Apache HTTP Server merupakan aplikasi untuk *server web* yang terpopuler di dunia, dengan pengguna pada bulan maret 2009 mencapai 46% dari seluruh *website* yang ada di internet dunia dijalankan menggunakan *Apache*. Dan pada tahun juga *Apache HTTP Server* tercatat sebagai *server web* pertama yang menembus angka pengguna mencapai 100 juta situs *web* di dunia.

2. *MySQL Database Server*

MySQL yang dipaketkan dalam Xampp merupakan aplikasi *server database* yang mumpuni, dan banyak digunakan pada aplikasi berbasis *web*. Bahkan banyak *website* besar dengan *traffic* yang tinggi memanfaatkan MySQL untuk penyimpanan basis data. *Website* besar yang menggunakan MySQL sebagai basis data diantaranya adalah *Facebook, Wikipedia, Google, dan YouTube*.

3. FileZilla FTP Server

FileZilla FTP Server merupakan aplikasi untuk server FTP (*File Transfer Protocol*), yang memungkinkan untuk menyediakan mekanisme berbagi file secara lebih aman, terstruktur, dan tanpa pembatasan jumlah sesi koneksi. Dengan FileZilla Server kita bisa membuat *user* dengan *home folder* terpisah dan memiliki *password* masing – masing, sehingga cocok untuk mekanisme pengumpulan tugas.



BAB V

PENUTUP

5.1. Kesimpulan

Bedasarkan rumusan masalah penelitian, maka diperoleh kesimpulan sebagai berikut :

1. Salah satu cara yang dapat memudahkan wisatawan dalam mencari informasi objek wisata di Kabupaten Pati dengan membuat sistem informasi berbasis *web* dengan mengimplementasikan algoritma *Dijkstra* untuk pencarian rute terpendek objek wisata di Kabupaten Pati.
2. Wisatawan dapat menentukan rute terpendek dari objek wisata satu ke objek wisata lainnya dengan memilih lokasi awal objek wisata dan lokasi akhir objek wisata yang terdapat pada halaman lokasi wisata pada sistem informasi objek wisata di Kabupaten Pati, dan akan diproses menggunakan algoritma *Dijkstra*.
3. Dalam menerapkan algoritma *Dijkstra* pada sistem pencarian rute terpendek objek wisata di Kabupaten Pati yaitu dengan memasukkan algoritma *Dijkstra* pada sistem pencarian rute terpendek sebagai penghitung jarak dari lokasi awal objek wisata sampai dengan lokasi akhir objek wisata. *Dijkstra* akan menghitung setiap jarak terpendek dari setiap titik atau objek wisata dari lokasi awal hingga lokasi akhir objek wisata. *Dijkstra* menyimpan setiap nilai terkecil pada setiap jarak dari tiap titik sampai memperoleh total jarak terpendek.

4. Wisatawan dapat mengetahui objek wisata terdekat di Kabupaten Pati dari posisi dimana mereka berada dengan meng-klik bagian *map* yang ada pada peta Lokasi Wisata sistem informasi objek wisata Kabupaten Pati.

5.2 Saran

Kekurangan sistem dari pencarian rute terpendek menggunakan algoritma *Dijkstra* ini adalah titik/*node* hanya sebatas antar objek wisata, tidak menambahkan titik/*node* disetiap persimpangan yang ada, karena sistem sudah langsung menentukan jarak antar objek wisata dari objek ke objek lainnya. Selain itu, pada sistem ini dalam mengoptimalkan jalur wisata hanya memperhatikan jarak tanpa memperhatikan variabel lain seperti lampu lalu lintas, tingkat kepadatan, maupun lebar jalan. Oleh karena itu, disarankan penelitian selanjutnya dapat mengembangkan sistem yang bisa mengoptimalkan algoritma *Dijkstra* dengan sempurna.

DAFTAR PUSTAKA

- Ardiani, F., 2011. *Penentuan Jarak Terpendek dan Waktu Tempuh menggunakan Algoritma Dijkstra dengan Pemrograman Berbasis Objek*. Skripsi. Yogyakarta: Universitas Islam Negeri Sunan Kalijaga.
- Ashar, D., 2014. *Sistem Informasi Geografis Lokasi Pangkalan Ojek Kabupaten Kendal Berbasis Web*. Skripsi. Semarang: Universitas STIKUBANK.
- Gusmão, A., Pramono, S.H. & Sunaryo, 2013. Sistem Informasi Geografis Pariwisata Berbasis Web Dan Pencarian Jalur Terpendek Dengan Algoritma Dijkstra. *Jurnal EECCIS*, 7(2): 125-130.
- Nugroho, B., 2004. *Aplikasi Pemrograman Web Dinamis dengan PHP dan MySQL*. Yogyakarta: Gava Media.
- Pressman, R.S., 2002. *Rekayasa Perangkat Lunak Pendekatan Praktisi Buku I*. Yogyakarta.
- Priatmoko, S.B., 2014. Algoritma Dijkstra Untuk Pencarian Jalur Terdekat Dan Rekomendasi Objek Pariwisata Di Pulau Bali. Semarang.
- Pugas, D.O., Somantri, M. & Satoto, K.I., 2011. Pencarian Rute Terpendek Menggunakan Algoritma Dijkstra Dan Astar (A*) Pada SIG Berbasis Web Untuk Pemetaan Pariwisata Kota Sawahlunto. Semarang.
- Purwananto, Y., Purwitasari, D. & Wibowo, A.W., 2005. Implementasi dan Analisis Algoritma Pencarian Rute Terpendek. *Jurnal Penelitian dan Pengembangan TELEKOMUNIKASI*, 10(2): 94-101.
- Puspika, B.N., Rachmat C, A. & Kurniawan, E., 2012. Implementasi Algoritma Dijkstra Dalam Penentuan Jalur kpendek Di Yogyakarta Menggunakan Gps dan Qt Geolocation. *Jurnal INFORMATIKA*, 8(2): 141-149.
- Sanan, S., Jain, L. & Kappor, B., 2013. Shortest Path Algorithm. *Journal of Application or Innovation in Engineering & Management*, 2(7): 316-320.
- Satyananda, D., 2012. *Struktur Data*. Malang: Universitas Negeri Malang.
- Solichin, A., 2010. *MySQL 5 : Dari Pemula Hingga Mahir*. [Online] Available at: <http://achmatim.net/download/18/> [Accessed 7 Maret 2016].

Undang-Undang Republik Indonesia Nomor 10 Tahun 2009 Kepariwisata. 16
Januari 2009

Wicaksono, Y., 2008. *Membangun Bisnis Online dengan Mambo*. Jakarta: PT. Elex
Media Komputindo.

