

IMPLEMENTASI ALGORITMA DIJKSTRA UNTUK PENCARIAN LINTASAN TERPENDEK LOKASI TEMPAT-TEMPAT UMUM DI KABUPATEN BONDOWOSO BERBASIS WEB

¹*Rian Hasriandi (0910651224)*

²*Lutfi Ali Muharrom, S.Si* ³*Daryanto, S.Kom, M.Kom*

Jurusan Teknik Informatika Fakultas Teknik Universitas Muhammadiyah Jember

Email : killtimeboyz37@yahoo.com

ABSTRAK

Pemanfaatan internet dalam pencarian suatu lokasi sangat dibutuhkan masyarakat dengan mobilitas yang tinggi dan kurang pahami rute-rute daerah tertentu. Terlebih pada daerah yang terdapat banyak tempat-tempat wisata yang sebagian besar pengunjungnya dari luar daerah tersebut atau bahkan dari mancanegara, yang mungkin kesulitan dalam mencari lokasi tempat wisata dan tempat-tempat umum lainnya. Kawah Ijen yang terletak di kabupaten Bondowoso adalah salah satu destinasi wisata yang paling populer. Untuk memajukan wisata tersebut, dibutuhkan banyak hal dari pihak-pihak yang terkait, salah satunya adalah diperlukannya Sistem informasi geografis seperti peta yang memudahkan mendapatkan informasi serta akses menuju lokasi wisata dan tempat-tempat umum lainnya.

Perancangan dilakukan dengan menggunakan metode Dijkstra's yang merupakan salah satu algoritma yang berguna untuk mencari lintasan terpendek dari satu titik ke titik lain. Metode *Dijkstra* dipilih karena merupakan algoritma yang lebih efisien dibandingkan algoritma lainnya seperti *Warshall*.

Kata Kunci : Sistem Informasi Geografis, *Dijkstra*

IMPLEMENTATION OF DIJKSTRA ALGORITHM FOR SHORTEST PATH TO SEARCH LOCATIONS OF PUBLIC PLACES IN THE BONDOWOSO DISTRICT WEB-BASED

¹Rian Hasriandi (0910651224)

²Lutfi Ali Muharrom, S.Si ³Daryanto, S.Kom, M.Kom

Department of Informatics Faculty of Engineering,

University of Muhammadiyah Jember

Email : killtimeboyz37@yahoo.com

ABSTRACT

Utilization of the internet for searching a location is very needed by the peoples with high mobility and low knowledge about route of an areas. Especially in areas where there are many tourist attractions which most visitors from the outside of the region or even from abroad, who may have difficulty to finding the location of tourist attractions and other public places. Ijen crater is one of the most popular tourist destinations located in the Bondowoso district. To promote the tour, it takes a lot of things from related sides, one of which is geographic information systems as a map that makes it easy to get information and access to tourist sites and other public places.

The design using Dijkstra's algorithm, is useful for finding the shortest path from one point to another. Dijkstra method chosen because it is more efficient algorithm than other algorithms such as Warshall.

Keywords : Geographic Information Systems, Dijkstra

I. PENDAHULUAN

1.1 Latar Belakang

Di era modern saat ini, perkembangan teknologi yang sangat pesat, dan banyaknya warga masyarakat yang paham dan mengerti akan penggunaan internet dalam membantu dan mempermudah kegiatan manusia sehari-hari, seperti pencarian data, untuk komunikasi serta pencarian suatu lokasi. Pemanfaatan internet dalam pencarian suatu lokasi sangat dibutuhkan masyarakat dengan mobilitas yang tinggi dan kurang pahamnya rute-rute daerah tertentu. Terlebih pada daerah yang terdapat banyak tempat-tempat wisata yang sebagian besar pengunjungnya dari luar daerah tersebut atau bahkan dari mancanegara, yang

mungkin kesulitan dalam mencari lokasi tempat wisata dan tempat-tempat umum lainnya.

Kawah Ijen yang terletak di kabupaten Bondowoso adalah salah satu destinasi para pendaki dari dalam dan luar negeri yang ingin mengetahui keindahan gunung, serta kawahnya. Atau hanya sekedar tempat liburan keluarga, karena banyak wisata yang disajikan didaerah sekitar Gunung kawah ijen, seperti air terjun, pemandian air panas, perkebunan kopi, dan beberapa wisata lain. Untuk memajukan wisata tersebut, dibutuhkan banyak hal dari pihak-pihak yang terkait, salah satunya adalah diperlukannya informasi geografis seperti peta yang memudahkan mendapatkan informasi serta

akses menuju lokasi wisata dan tempat-tempat umum lainnya.

Berdasarkan uraian di atas, penulis mencoba mengangkat sebuah topik yaitu implementasi algoritma *Dijkstra* untuk pencarian lintasan terpendek lokasi tempat-tempat umum di Kabupaten Bondowoso berbasis *web*. Pada penelitian ini akan dibentuk suatu website yang dapat dimanfaatkan untuk pencarian lintasan terpendek menuju lokasi tempat-tempat umum di Kabupaten Bondowoso dan beberapa informasi tentang lokasi tersebut sehingga informasi dapat diperoleh secara online dengan menghasilkan lintasan optimal berdasarkan algoritma *Dijkstra*. Algoritma *Dijkstra* dipilih karena merupakan algoritma yang lebih efisien dibandingkan algoritma lainnya seperti *Warshall*.

1.2 Perumusan Masalah

Berdasar latar belakang masalah tersebut, maka masalah yang akan dikaji penulis dalam Tugas Akhir ini adalah :

1. Bagaimana merancang dan membangun aplikasi berbasis *web* yang mampu memberikan informasi tentang rute terpendek dari suatu lokasi ke lokasi yang lain.
2. Bagaimana mengimplementasikan algoritma *Dijkstra* ke dalam sebuah sistem informasi geografis sehingga dapat digunakan untuk mencari lintasan terpendek pada suatu rute.

1.3 Tujuan

Tujuan dari penulisan tugas akhir ini berdasar pada perumusan masalah diatas adalah sebagai berikut :

1. Merancang dan membangun aplikasi berbasis *web* yang fungsinya untuk mencari lintasan terpendek pada suatu rute.
2. Mengimplementasikan algoritma *Dijkstra* ke dalam sebuah sistem informasi geografis sehingga dapat menentukan lintasan terpendek dalam mencari lokasi fasilitas-fasilitas umum di Kabupaten Bondowoso berbasis *web*.

1.4 Batasan Masalah

Pada penyelesaian proyek akhir ini terdapat beberapa batasan masalah yang dikaitkan dengan perancangan dan implementasi algoritma *Dijkstra* untuk pencarian lintasan terpendek lokasi tempat-tempat umum di Kabupaten Bondowoso berbasis *web*, antara lain:

1. Perhitungan dalam pencarian lintasan terpendek menggunakan satu kendaraan dengan jenis dan tipe yang sama, dan kecepatan laju kendaraan yang sama dan konstan.
2. Permasalahan-permasalahan yang terjadi pada jalur lintasan dianggap tidak ada, seperti kerusakan jalan dan gangguan lainnya, konflik pergerakan seperti *merging*, *crossing*, dan *diverging* pada persimpangan, kendaraan lain yang melintas, serta *traffic light*, tanpa

mengesampingkan jalur lalu lintas, yaitu tetap memperhitungkan arah jalur, apakah itu jalur dua arah atau searah.

3. Menggunakan algoritma *Dijkstra* untuk mencari lintasan terpendek.
4. *Vertex* berupa fasilitas umum berupa instansi pemerintahan, rumah sakit, hotel, tempat ibadah, pusat perbelanjaan, terminal transportasi, dan lainnya yang dilalui jalan protokol ataupun persimpangan jalan protokol.

II. TINJAUAN PUSTAKA

2.1 Algoritma Dijkstra

Algoritma ini ditemukan oleh Edsger W. Dijkstra dan di publikasi pada tahun 1959 pada sebuah jurnal *Numerische Mathematik* yang berjudul “A Note on Two Problems in Connexion with Graphs”. Algoritma ini sering digambarkan sebagai algoritma *greedy* (tamak).

Dijkstra merupakan salah satu varian bentuk algoritma populer dalam pemecahan persoalan terkait masalah optimasi pencarian lintasan terpendek sebuah lintasan yang mempunyai panjang minimum dari *verteks* a ke z dalam graf berbobot, bobot tersebut adalah bilangan positif jadi tidak dapat dilalui oleh node negatif. Namun jika terjadi demikian, maka penyelesaian yang diberikan adalah *infiniti* (Tak Hingga). Pada algoritma *Dijkstra*, node digunakan karena algoritma Dijkstra menggunakan graph berarah untuk penentuan rute listasan terpendek. Berikut *Pseudo Code* dan *Flowchart Algoritma Dijkstra*.

2.2 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 titik. Banyak sekali struktur yang bias direpresentasikan dengan graf, dan banyak masalah yang bisa diselesaikan dengan bantuan graf. Graf juga digunakan untuk merepresentasikan suatu jaringan. Misalkan jaringan jalan raya dimodelkan graf dengan kota sebagai titik/node dan jalan yang menghubungkan setiap kota sebagai sisi (garis).

Sebuah graf linier (atau secara sederhana disebut graf) $G = (V, E)$ adalah suatu sistem yang terdiri atas suatu himpunan objek $V = \{ v_1, v_2, \dots \}$ yang disebut himpunan titik, dan sebuah himpunan $E = \{ e_1, e_2, \dots \}$ yang merupakan himpunan sisi sedemikian hingga tiap sisi ek dikaitkan dengan suatu pasangan tak-terurut (v_i, v_j) . Titik v_i dan v_j yang berkaitan dengan ek disebut titik-titik ujung sisi ek. (Sutarno, dkk. 2003:59). Titik pada graf dapat dilabeli dengan huruf, misalkan v, w, \dots , atau dengan menggunakan bilangan asli $1, 2, 3, \dots$, atau gabungan keduanya. Sedangkan sisi yang menghubungkan titik v_i dengan titik v_j dinyatakan dengan pasangan (v_i, v_j) , atau dengan lambang e_1, e_2, \dots . Dengan kata lain, jika e adalah sisi yang menghubungkan titik v_i dengan titik v_j , maka e dapat dituliskan sebagai $e = (v_i, v_j)$, dimana i, j adalah indeks angka bilangan asli $1, 2, 3, \dots$

Cara merepresentasikan sebuah graf yang paling umum adalah dengan diagram.

Tiap-tiap diagram memuat sekumpulan objek (titik) dengan garis-garis yang menghubungkan objek-objek tersebut. Garis bisa berarah ataupun tidak berarah. Garis yang berarah digunakan untuk menyatakan hubungan yang mementingkan urutan antar objek-objek. Urut-urutan objek akan mempunyai artitain jika arah garis diubah. Garis yang tidak berarah digunakan untuk menyatakan hubungan antar objek-objek yang tidak mementingkan urutan. Dalam menggambarkan sebuah graf, bentuk sisi dapat berupa ruas garis/sisi lurus atau lengkung.

Setiap sisi berhubungan dengan satu atau dua titik. Titik-titik tersebut dinamakan titik ujung. sisi yang hanya berhubungan dengan satu titik ujung disebut *Loop*. Dua sisi berbeda yang menghubungkan titik yang sama disebut sisi paralel. Dua titik dikatakan berhubungan langsung (*adjacent*) jika ada sisi yang menghubungkan keduanya. Titik yang tidak mempunyai sisi yang berhubungan dengannya disebut titik terasing (*Isolating Point*).

2.3 PHP

PHP singkatan dari *Hypertext PreProcessor* merupakan bahasa pemogram yang skript bersifat Open Source. Program ini bersifat server side, artinya tanpa adanya server yang berjalan disisinya script program PHP tidak dapat dijalankan (Nugroho, 2005). PHP adalah script yang ditanamkan dalam HTML untuk membuat halaman website dinamis yang bekerja secara otomatis dan berfungsi sebagai pengolahan data pada server di mana script tersebut di jalankan (Anhar, 2010).

Prasetio (2012), PHP adalah bahasa script yang ditanam disisi server memiliki

kemampuan untuk memisahkan kode dari HTML. Beberapa kelebihan dari PHP sebagai berikut.

1. Kesederhanaan, mudah pelajari karna banyak referensi serta bisa membuat website dinamis.
2. PHP bersifat *open source*, karena bersifat *open source* PHP mudah didapatkan dan tersedia secara versi-versi baru dalam jangka waktu yang cepat.
3. Stabilitas dan Kompatibilitas , PHP stabil di berbagai sistem operasi seperti linux dan Macs selain itu PHP juga terintergrasi secara baik dengan berbagai macam webserver termasuk 2 yang paling populer yaitu IIS dan Apache.
4. Kemampuan proses cepat dalam menampilkan halam web serta mampu berintraksi banyak *database*.

Dalam penulisan syntax kode PHP diawali dengan tanda “<?php” (atau cukup singkat bisa di tulis seperti ini “<?” dan ditutup dengan dengan menambahkan tanda tutup kurung seperti ini “?” di akhir blok kodenya. Pada setiap baris instruksi program di akhir dengan tanda titik koma “.” artinya walaupun menuliskan lebih dari satu baris tetapa akan di anggap satu baris. Selain itu banyak penggunaan tanda kurug dalam penulisan syntax kode PHP yang sering dilakukan adalah untuk memanggil fungsi. Secara sederhana, setiap fungsi PHP akan berbentuk seperti ini `print ();`.

2.4 XAMPP

XAMPP merupakan paket PHP dan MYSQL berbasis *open source* yang mampu membuat web dinamis. XAMPP aplikasi webserver yang mendukung

instalasi Linux dan Windows selain itu XAMPP juga termasuk aplikasi *open source*. XAMPP aplikasi yang terdiri dari beberapa paket aplikasi server seperti Apache Http Server, MySQL Database Server, Filezilla FTP Server XAMPP juga mudah di download berbagai situs jaringan.

Mengenal bagian XAMPP yang biasa digunakan pada umumnya yaitu :

1. htdoc adalah folder tempat meletakkan berkas-berkas yang akan dijalankan, seperti berkas PHP, HTML dan skrip lain.
2. phpMyAdmin merupakan bagian untuk mengelola basis data MySQL yang ada dikomputer. Untuk membukanya, buka browser lalu ketikkan alamat <http://localhost/phpMyAdmin>, maka akan muncul halaman phpMyAdmin.
3. Kontrol Panel yang berfungsi untuk mengelola layanan (*service*) XAMPP. Seperti menghentikan (*stop*) layanan, ataupun memulai (*start*).

2.5 MYSQL

MYSQL merupakan salah satu *database* yang bersifat *open source* yang populer di dunia, dimana saat ini digunakan lebih dari 100 juta pengguna diseluruh dunia MYSQL turunan dari SQL (Structured Query Language) yang sering digunakan oleh para programmer dalam pembuatan program. Sistem *database* MYSQL selain mudah memahami sintaks dan pengaksesan juga di dukung beberapa

fitur seperti Multithreaded, multi-user dan SQL *database* managemen sistem (Solichin, 2010).

Anhar (2010), MYSQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) yang mudah didapatkan secara gratis. Beberapa kelebihan MYSQL antara lain sebagai berikut:

1. MYSQL mendukung sistem operasi, seperti Windows, Linux, FreeBSD, Mac Os X Server solaris.
2. Bersifat *Open Source*, karena bersifat *open source* MYSQL mudah didapatkan secara gratis.
3. Bersifat Multiuser, MYSQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah.
4. MYSQL memiliki kemampuan kecepatan proses yang tinggi dalam memproses data.

2.6 Tempat-tempat Umum

Tempat umum adalah suatu tempat yang umumnya terdapat banyak orang yang berkumpul untuk melakukan suatu kegiatan baik secara sementara maupun secara terus menerus dan baik membayar maupun tidak membayar. Tempat umum juga dapat diartikan sebagai sarana yang diselenggarakan oleh pemerintah, swasta atau perorangan yang digunakan untuk kegiatan bagi masyarakat. Contoh dari tempat-tempat umum adalah SPBU, kantor polisi, terminal bus, stasiun kereta, pasar, dan lain sebagainya.

Kriteria yang harus dimiliki suatu tempat sehingga bisa dikategorikan

sebagai tempat umum adalah sebagai berikut :

1. Diperuntukkan bagi masyarakat umum.
2. Harus ada gedung atau tempat yang permanen.
3. Harus ada aktivitas (pengusaha, pegawai, dan pengunjung).
4. Harus ada fasilitas (SAB, WC, Urinoir, tempat sampah, dan lain-lain) sanitasi
5. Tempat-tempat umum, yaitu suatu usaha untuk mengawasi dan mencegah kerugian akibat dari tidak terawatnya tempat-tempat umum tersebut yang mengakibatkan timbul dan menularnya berbagai jenis penyakit.

2.7 Bing Maps

Bing Maps (sebelumnya *Live Search Maps*, *Windows Live Maps*, *Windows Live Local*, dan *MSN Virtual Earth*) adalah layanan pemetaan web yang disediakan sebagai bagian dari mesin pencari Microsoft 's *bing suite*. Bing Maps didasarkan pada teknologi Microsoft seperti Microsoft Map Point, dan TerraServer. Versi asli tidak memiliki banyak fitur yang membedakan, termasuk *Bird Eyes View* dan *3D Map*, dan fungsi koleksi terbatas pada satu "*Scratchpad*" pada tempat-tempat menarik. Setelah dirilis pada bulan Desember 2005, *Windows Live Local* menjadi tampilan dari platform *Virtual Earth*. Pada tanggal 6 November 2006, Microsoft menambahkan kemampuan untuk melihat peta dalam bentuk 3D menggunakan .NET. Microsoft kemudian mengeluarkan produk ini secara

resmi sebagai "*Live Search Maps*", diintegrasikan sebagai bagian dari *Live Search Services*. Pada tanggal 3 Juni 2009, Microsoft secara resmi diganti namanya dari "*Live Search Maps*" sebagai Bing Maps, dan *The Virtual Earth Platform* sebagai *Bing Maps for Enterprise*.

III. METODOLOGI PENELITIAN

3.1 Konsep Penelitian

Secara sistematis langkah-langkah dalam Tugas Akhir dijadikan dalam bentuk diagram alir seperti dibawah ini.



Gambar 3.1 Diagram Alir Penelitian

Penjelasan singkat tentang Diagram Alir diatas.

1. Studi Literatur

Pengerjaan tugas akhir ini dimulai dengan mengumpulkan bahan-bahan sebagai referensi baik dari buku, paper, jurnal, makalah, forum, milis, dan sumber-sumber lain yang berkaitan dan beberapa referensi lainnya untuk menunjang pencapaian tujuan tugas akhir.

2. Observasi

Metode ini dilakukan dengan melakukan pengamatan dan pengujian terhadap beberapa sistem informasi geografis dan aplikasi untuk mencari lintasan terpendek dengan melakukan penelusuran di internet. Dengan pengamatan secara langsung tersebut akan diperoleh pengetahuan bagaimana bentuk sistem yang ada dan telah diimplementasikan, serta untuk menentukan penggunaan *map* apa yang baik untuk perancangan aplikasi tersebut.

3. Analisis

Pada tahap ini dilakukan analisis permasalahan yang ada, batasan yang dimiliki dan kebutuhan yang diperlukan.

4. Perancangan dan Implementasi Algoritma

Metode ini akan dilaksanakan dengan melakukan perancangan sistem informasi geografis dan mengimplementasikan algoritma Dijkstra kedalam *map* yang telah ditentukan sebelumnya, sehingga terbentuknya sistem ini.

5. Pengujian

Pada tahap ini dilakukan pengujian terhadap sistem informasi geografis yang telah dibangun, bagaimana tampilan tatap mukanya, fungsi dari fitur-fitur yang tersedia apakah

bekerja dengan baik, dan proses penyimpanan data kedalam database, serta menguji kebenaran algoritma Dijkstra dalam pengaplikasian pencarian lintasan terpendek dalam suatu peta beserta koneksinya kedalam suatu database.

6. Penyusunan Laporan dan Kesimpulan Akhir

Metode ini akan dilaksanakan dengan melakukan pendokumentasian hasil analisis dan implementasi secara tertulis dalam bentuk laporan skripsi.

3.2 Spesifikasi Program

Beberapa fitur yang terdapat dalam program adalah sebagai berikut :

1. Tampilan

Hanya menampilkan graf Algoritma *Dijkstra*, yaitu berupa beberapa titik/node yang menandakan suatu lokasi beserta garis rute yang menandakan jalur yang menghubungkan node tersebut dengan node lain, sehingga diketahui jaraknya. Dan tampilan tabel yang berisi informasi tentang lokasi awal dan tujuan, beserta jarak yang ditempuh.

2. Add Vertex atau Tambah Lokasi

Fitur yang memungkinkan pengguna menambahkan titik lokasi baru atau vertex yang belum terdaftar pada peta sebelumnya, sehingga dapat menambah daftar lokasi baru pada peta beserta relasi dengan titik lokasi tujuan, sehingga dapat diketahui jarak antara titik asal ke titik tujuan.

3. Search atau pencarian

Pengguna dapat dengan mudah mencari lokasi pada peta dengan menginput nama lokasi, sehingga memudahkan dalam pencarian.

4. Pilihan rute

Bertujuan untuk memberi pilihan jalur/rute pada pengguna dari satu titik lokasi ke lokasi tujuan pengguna, sehingga dapat menentukan pilihan menggunakan rute terpendek atau tidak, karena alasan efisiensi waktu.

apakah titik tersebut dilalui atau tidak.

4. Menambahkan jarak antara tiap relasi titik awal menuju titik tujuan berdasarkan pada map, yang tujuannya untuk memberikan bobot pada graf, sehingga dapat diketahui rute terpendek pada graf berdasarkan algoritma dijkstra.

5. Mengimplementasikan hasil rancangan Graf kedalam aplikasi pencarian rute terpendek, sehingga dapat ditampilkan didalam aplikasi yang telah dibangun sebelumnya.

3.3 Integrasi Google Map dan Graf

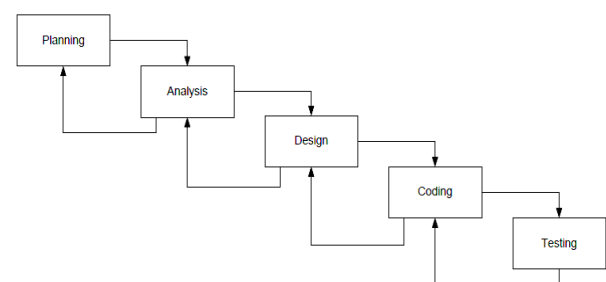
Dalam pengimplementasian algoritma dijkstra pada sistem pencarian rute terpendek ini, dibutuhkan map yang bertujuan untuk mencari lokasi yang akan dimasukan kedalam *database* beserta jaraknya dan menampilkan lokasi-lokasi tersebut ke dalam aplikasi yaitu berupa graf berbobot beserta jalurnya antara satu titik ke titik tujuan yaitu sebagai berikut.

1. Dibutuhkannya koneksi internet untuk mengakses map, dalam hal ini penulis menggunakan google map.
2. Menampilkan peta Kabupaten Bondowoso pada map, kemudian mencatat semua lokasi-lokasi tempat umum yang akan dimasukkan kedalam database, beserta nilai longitude dan latitude masing-masing lokasi.
3. Membuat gambar graf yaitu berupa node-node dari lokasi yang telah dipilih sebelumnya, beserta garis atau jalur penghubung antar vertex atau titik yang satu ke titik lain,

3.4 Metode Pengembangan Sistem

Metode Pengembangan Sistem yang digunakan dalam hal ini adalah menggunakan model *Waterfall*. Model *Waterfall* ini mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada sebuah Planning, analisis, desain, coding dan pengujian.

Untuk lebih jelasnya tahap-tahap dari paradigma waterfall dapat dilihat pada gambar berikut ini.



Gambar 3.2 *Waterfall* model pengembangan sistem

1. Planning

Pada tahap *planning* terdapat tiga bagian yaitu :

a. *Estimation*

Estimation

merupakan proses penetapan perkiraan semua komponen proyek, seperti perkiraan waktu, biaya dan sumberdaya yang dibutuhkan oleh sistem.

b. *Scheduling*

Scheduling

merupakan tahapan penjadwalan dari seluruh kegiatan pengembangan sistem.

2. *Analysis*

Analysis merupakan proses analisa terhadap kemungkinan resiko yang terjadi pada proses pengembangan sistem dilakukan.

3. *Design*

Tahap *design* merupakan tahap dilakukannya permodelan system. Pada system yang berorientasi objek, pendisainan biasanya menggunakan model *Unified Modeling Language* (UML).

4. *Coding*

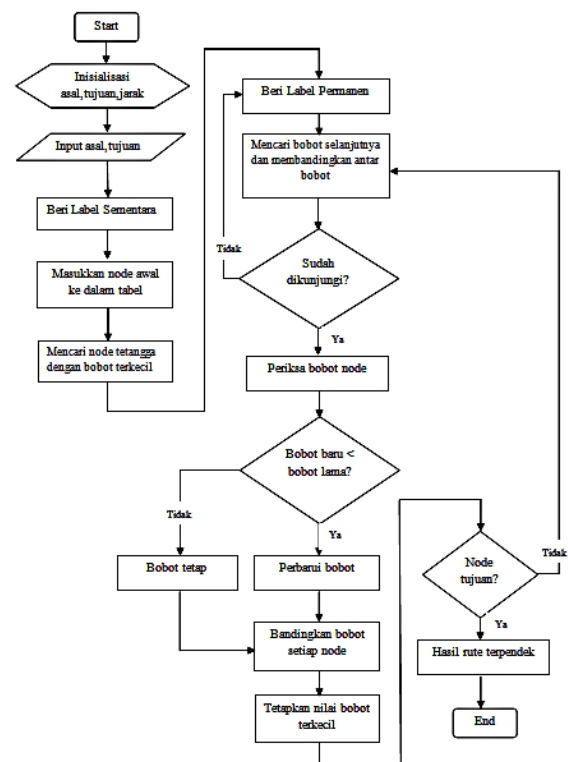
Tahap *coding* meliputi serangkaian pengkodean yang dilakukan untuk membuat *prototype* dan bagian-bagian program lainnya yang pada akhirnya akan diintegrasikan menjadi satu kesatuan yang membentuk sebuah system yang utuh.

5. *Testing*

Tahap *testing* merupakan serangkaian pengujian terhadap system secara menyeluruh.

Tahap ini biasanya menggunakan metode pengujian *black box testing* ataupun *white box testing*. *Black box testing* adalah uji coba yang memfokuskan pada keperluan fungsional dari *software*. Sementara itu *White box testing* merupakan pengujian dengan melihat kedalam modul untuk meneliti kode-kode program yang ada, dan menganalisis apakah ada kesalahan atau tidak.

3.5 Flowchart Algoritma Dijkstra



Gambar 3.3 Flowchart Algoritma Dijkstra

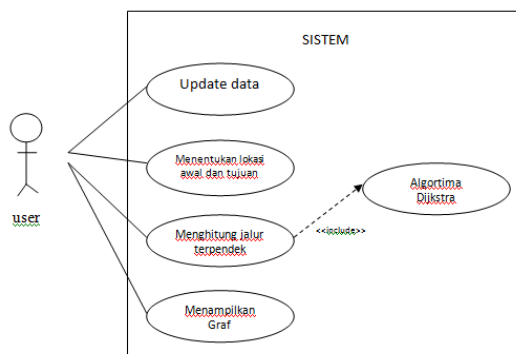
Pada flowchart di atas dapat dijelaskan proses algoritma dijkstra adalah sebagai berikut :

Masukkan : Graf berbobot.

Proses :

1. Inisialisasi vertex.
2. Inisialisasi jarak antar vertex.
3. Tentukan vertex awal (s) dan vertex tujuan (t).
4. Beri label permanen = 0 ke verteks awal (s) dan label sementara = ∞ ke verteks lainnya.
5. Untuk setiap verteks V yang belum mendapat label permanen, mendapat label sementara = $\min \{\text{label lama V}, (\text{label lama V} + D)\}$.
6. Cari nilai minimum diantara semua verteks yang masih berlabel sementara.
7. Jadikan verteks minimum yang berlabel sementara menjadi verteks dengan label permanen, jika lebih dari satu verteks pilih sembarang.
8. Ulangi langkah 5 sampai 7 hingga verteks tujuan mendapat label permanen.
9. Simpan hasil perhitungan. Tampilkan hasil perhitungan.

3.6 Use Case Diagram



Gambar 3.4 Use case diagram

Pada gambar diatas terlihat bahwa sistem ini hanya memiliki satu aktor utama yaitu pengguna. Pengguna dapat melakukan update data, menentukan lokasi awal dan tujuan, menghitung dan menentukan jalur terpendek dari lokasi awal menuju lokasi yang dituju menggunakan Algoritma *Dijkstra*, dan menampilkan hasil perhitungan jalur terpendek yaitu berupa *Graf*.

III. HASIL DAN PEMBAHASAN

4.1 Implementasi Ruang Lingkup

Untuk menjalankan aplikasi pencarian rute terdekat menggunakan algoritma *Dijkstra* berbasis *web*, sebelumnya harus mempersiapkan kebutuhan dari program yang akan di implementasikan baik dari segi perangkat keras maupun perangkat lunak komputer.

4.1.1 Ruang Lingkup Perangkat Keras

Perangkat keras yang di gunakan dalam pengembangan aplikasi pencarian rute terdekat ini adalah sebagai berikut.

1. Laptop ACER ASPIRE 4736, Intel Core 2 Duo T6570 (2.1 Ghz, 800 MHz FSB), RAM 1 GB, 14.0" HD LED LCD.

2. Mouse

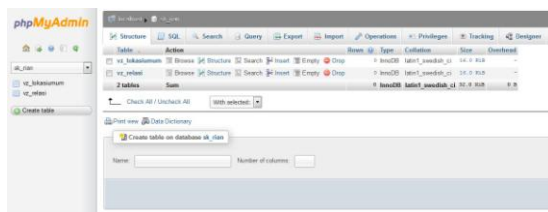
4.1.2 Ruang Lingkup Perangkat Lunak

1. Sistem Operasi Windows 7
2. Notepad++ v6.1.3
3. Xampp v1.7.7
4. Mozilla Firefox

4.1.3 Implementasi Basis Data

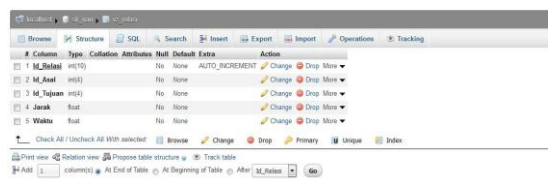
Pembangunan basis data dilakukan dengan menggunakan bahasa SQL, implementasi basis datanya sebagai berikut :

- a. Database yang dibuat hanya terdiri dari 2 tabel, yaitu tabel vz_lokasiumum dan tabel vz_relasi. Seperti berikut :



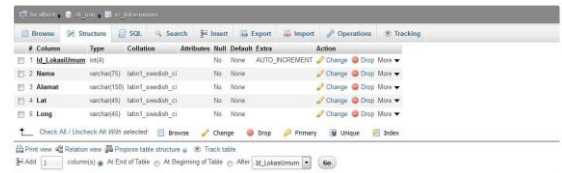
Gambar 4.1 Struktur tabel *database*

- b. Struktur dari tabel vz_relasi hanya terdiri dari 5 kolom, yaitu id_relasi, id_Asal, id_Tujuan, Jarak, dan Waktu. Untuk lebih jelasnyadapat dilihat pada gambar berikut ini.



Gambar 4.2 Struktur tabel relasi

- c. Struktur dari tabel vz_lokasiumum hanya terdiri dari 5 kolom, yaitu id_lokasiumum, Nama, Alamat, Latitude, dan Longitude. Untuk lebih jelasnya dapat dilihat pada gambar berikut ini.



Gambar 4.3 Struktur Tabel Lokasi Umum

- d. Bentuk Tabel Lokasi Umum yaitu seperti gambar berikut.

	id_lokasiumum	Nama	Alamat	Lat	Long
Edit	1	Masjid Al-Taqwa	-7.912723	113.821411	
Edit	2	RS Bhayangkara	-7.910872	113.822362	
Edit	3	Pujasera Kotakulon	-7.906301	113.809008	
Edit	4	Stadion Magenda	-7.915336	113.809793	
Edit	5	Hotel Palm	-7.917145	113.819084	
Edit	6	Pasar Induk	-7.913950	113.827688	
Edit	7	Gedung Serbaguna	-7.923280	113.822601	
Edit	8	Bank BCA	-7.912749	113.829884	
Edit	9	Stasiun KAI	-7.916968	113.829428	
Edit	10	Bank BRI	-7.920241	113.828135	
Edit	11	POM Bensin	-7.921096	113.827132	
Edit	12	Hotel Ijen View	-7.929070	113.824814	
Edit	13	Terminal	-7.915902	113.831757	
Edit	14	RS Mitra Medika	-7.920971	113.832218	

Gambar 4.4 Tabel lokasi umum

- e. Bentuk Tabel Relasi yaitu seperti gambar berikut.

	Id_Relasi	Id_Asal	Id_Tujuan	Jarak	Waktu
Edit	1	1	2	800	0
Edit	2	2	1	800	0
Edit	3	2	4	1600	
Edit	4	4	2	1600	
Edit	5	1	5	600	
Edit	6	2	5	1400	
Edit	7	1	4	1500	
Edit	8	3	5	2300	
Edit	9	3	4	1500	
Edit	10	1	6	1100	
Edit	11	2	6	1900	0
Edit	12	6	8	400	0
Edit	13	1	8	1500	0
Edit	15	6	9	400	0
Edit	16	1	9	1500	0
Edit	17	8	13	800	0
Edit	18	8	9	800	0
Edit	19	8	6	400	0
Edit	20	8	1	1500	0
Edit	21	8	2	2300	0
Edit	22	8	7	2100	0

Gambar 4.5 Tabel Relasi

4.2 Implementasi Antarmuka

Setelah semua komponen komputer yang mendukung proses aplikasi terinstall,

proses selanjutnya adalah penjelasan program. Penjelasan program merupakan petunjuk yang dapat digunakan dalam menjalankan suatu program. Pada Tugas Akhir ini akan dijelaskan mengenai penggunaan program yang ada, antara lain Menu Utama, Data Lokasi Umum dan Implementasi Dijkstra.

4.2.1 Menu Utama

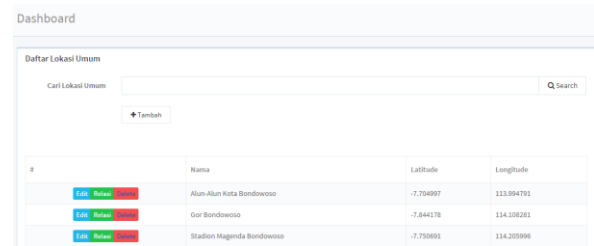
Menu Utama adalah tampilan awal program dimana pada menu utama ini info nama sistem yang dibangun. Selain itu terdapat menu lain seperti Data Lokasi Umum dan Implementasi Dijkstra. Adapun gambaran dari menu utama tampak pada gambar dibawah ini.



Gambar 4.6 Halaman Utama

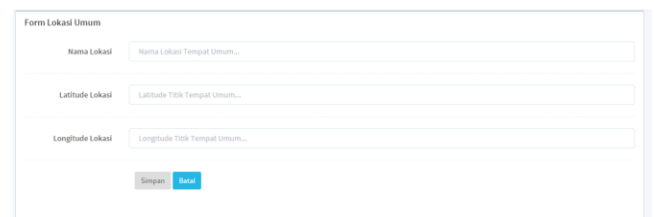
4.2.2 Menu Data Lokasi Umum

Menu data lokasi umum terdiri dari penambahan lokasi baru, pencari lokasi, relasi lokasi yang akan dihubungkan ketika di generate maps, serta edit dan delete daftar lokasi yang telah disebutkan.dalam tabel daftar lokasi umum berisi tentang nama lokasi, latitude, dan juga longitude lokasi pada peta. Adapun tampilan menu menu data lokasi seperti berikut ini :



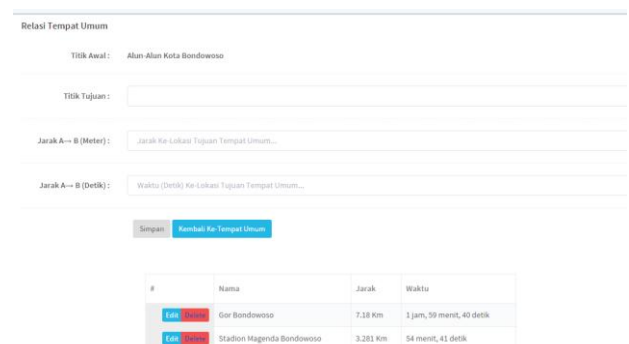
Gambar 4.7 Menu Data Lokasi

Pada halaman menu lokasi jika user menekan tombol tambah akan membuka halaman form input data lokasi umum, yaitu nama lokasi, latitude lokasi, dan longitude lokasi, seperti gambar dibawah ini:



Gambar 4.8 Form Lokasi Umum

Setelah user menambahkan data lokasi maka selanjutnya akan memberi alur yang terhubung antar lokasi umum, berikut tampilan relasi lokasi :



Gambar 4.9 Form Relasi Antar Lokasi

Pada form diatas, user diminta untuk menambahkan titik tujuan berdasarkan titik awal yang sudah ditentukan, jarak dan waktu yang ditempuh dari lokasi awal ke lokasi tujuan. Setelah data disimpan, maka akan tampil pada tabel relasi yang berisi tentang lokasi-lokasi yang mempunyai hubungan dengan titik awal, jarak beserta waktunya.

4.2.3 Menu Implementasi Dijkstra

Pembahasan mengenai pencarian rute terpendek tempat umum di Bondowoso menggunakan algoritma Dijkstra akan di jelaskan mulai proses penentuan posisi awal menuju lokasi tempat umum tujuan sampai penampakan hasil pencarian pada peta dan kemudian ditampilkan berupa graf berbobot. Berikut adalah urutan proses pencarian rute terpendek tersebut :

1) Penentuan posisi awal

Tahap awal yang di lakukan user yang ingin mencari rute terpendek tempat umum di Bondowoso adalah dengan melakukan input posisi awal keberangkatan menuju lokasi tujuan. Berikut ini adalah form input posisi awal keberangkatan.

The screenshot shows a web form titled "Implementasi Algoritma Dijkstra". It has two input fields: "Titik Awal:" and "Titik Tujuan:". Below these fields is a button labeled "Tampilkan".

Gambar 4.10 Posisi Awal Pemberangkatan

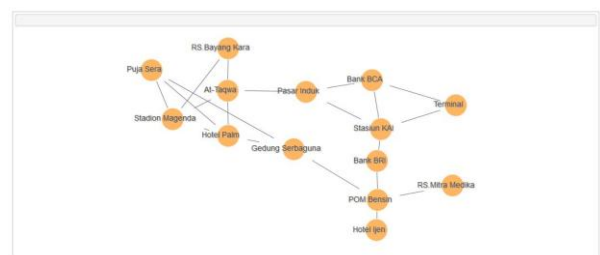
2) Hasil pencarian rute terpendek

Setelah melakukan input posisi awal keberangkatan dan lokasi tujuan kemudian menekan tombol “Tampilkan”, maka sistem akan menampilkan form hasil pencarian rute terpendek menuju tempat umum yang dituju dan form hasil pencarian rute terpendek menuju tempat umum tujuan dan juga tampilan graf lokasi umum tersebut.

Adapun tampilan hasil pencarian rute terpendek menggunakan algoritma Dijkstra di tunjukkan seperti berikut ini :

The screenshot shows a table with columns representing different locations (A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z). The rows show the shortest path and distance from the starting point to each location. For example, the first row shows the path from A to B as 1. Jalan, 2. Jalan, with a distance of 14 meters and 41 seconds.

Gambar 4.11 Pencarian Data Jalur Terpendek Untuk Bondowoso Map



Gambar 4.12 Graf Jalur terpendek lokasi umum

V. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah melakukan analisa, merancang dan mengimplementasikan perangkat lunak penentuan rute terpendek lokasi tempat-tempat umum di Kabupaten Bondowoso menggunakan algoritma Dijkstra, diperoleh kesimpulan dari hasil uji coba yaitu sebagai berikut:

1. Dengan algoritma Dijkstra dapat dicari rute terpendek yang ditempuh dari lokasi awal menuju lokasi tujuan tempat-tempat umum di kabupaten Bondowoso.
2. Sistem ini tidak dapat menentukan bobot graf serta jarak dari titik lokasi ke titik lokasi yang lain secara otomatis tanpa menambahkan data dan relasi antara titik lokasi awal dan tujuan secara manual.
3. Berdasar uji coba, fitur yang terdapat dalam aplikasi tersebut berjalan sebagaimana mestinya berdasarkan fungsi-fungsinya.
4. Hasil tampilan hanya berupa graf sederhana yang dirancang penulis tanpa koordinat, sehingga letak nodenya tidak begitu akurat.
5. Akurasi dalam proses pembuatan atau dijitasi jalan akan mempengaruhi akurasi hasil perutean, semakin akurat data jalan, maka semakin akurat pula hasil perutean yang akan ditampilkan.
6. Algoritma Dijkstra merupakan salah satu algoritma yang

digunakan untuk memecahkan permasalahan lintasan terpendek yang terdapat pada suatu graf.

7. Belum mampu mempertimbangkan kemacetan serta hambatan lain suatu jalan sehingga rute yang diberikan bisa jadi rute terpendek tetapi bukan rute tercepat.
8. *User* hanya bisa memilih tempat asal dan tujuan sesuai dengan yang sudah disediakan oleh sistem dan belum bisa mengetahui dimana *user* itu berada.
9. Algoritma ini digunakan pada graf berbobot dengan syarat bobot dari masing-masing sisi haruslah bernilai positif ($>=0$).
10. Salah satu implementasi dari algoritma Dijkstra ialah penyimpanan simpul dari suatu himpunan ke dalam suatu array atau list berkait (linked list).
11. Dalam mencari lintasan terpendek, algoritma yang paling banyak digunakan orang ialah algoritma Dijkstra, karena kerjanya efisien dan tidak membutuhkan waktu yang banyak. Tetapi Algoritma Dijkstra yang menerapkan prinsip greedy tidak selalu berhasil memberikan solusi optimum untuk kasus penentuan lintasan terpendek (single pair shortest path).

5.2. Saran

Berdasarkan hasil penelitian, penulis merekomendasikan beberapa saran yaitu sebagai berikut.

1. Lokasi-lokasi yang di sebutkan sangatlah terbatas dan hanya dalam area kabupaten Bondowoso, sehingga kedepannya diharapkan dapat menambah titik lokasi yang baru.
2. Sistem menjadi hidup jika user dapat mengontrol sistem, maksudnya user dapat menentukan selain daripada yang telah disebutkan.
3. Perlu dikembangkan lagi perangkat lunak penentuan rute disertai animasi sehingga dapat dihasilkan proses simulasi yang interaktif, namun tidak mengurangi segi validitas data.
4. Dapat dikembangkan lagi soal tampilan tentang map, sehingga dapat menampilkan map yang detil tidak hanya menampilkan graf saja.
5. Untuk sistem ini dapat juga dikembangkan suatu metode penentuan jalur terpendek yang lain sebagai pembanding.

DAFTAR PUSTAKA

- Anhar, S. T. 2010. Panduan menguasai PHP dan MySQL secara Otodidak. *Mediakita: Jakarta*.
- Dijkstra, Edsger W. 1959. *A note on two problems in connexion with graphs. Numerische mathematik*, 1.1: 269-271.
- Fitria, Apri Triansyah. 2013. *Implementasi Algoritma Dijkstra Dalam Aplikasi Untuk Menentukan Lintasan Terpendek Jalan Darat Antar Kota Di Sumatera Bagian Selatan*. Skripsi. Bandar Lampung : Universitas Sriwijaya.
- Knuth, Donald E. 1973. *Fundamental Algorithms: The art of computer programming*.
- Nugroho, B. 2005. *Database Relasional dengan MySQL*. Andi Offset: Yogyakarta.
- Nurhayati, Oky Dwi. 2010. *Algoritma Dan Pemrograman*. Universitas Diponegoro.
- Prasetyo, Verly Zulli. 2013. *Penerapan Algoritma Dijkstra Untuk Perutean Adaptif Pada Jaringan Pendistribusian Air PDAM Di Kabupaten Demak*. Skripsi. Semarang : Universitas Negeri Semarang.
- Prasetio, A. 2011. *Tip & Trik: Menjadi Master PHP*. MediaKita.
- Solichin, A. 2010. *MySQL5: Dari Pemula Hingga Mahir*. Achmad Solichin.
- Wicaksana, Dinar Anggit, et al. 2013. *SOLUSI TRAVELLING SALESMAN PROBLEM MENGGUNAKAN ALGORITMA FUZZY EVOLUSI (Studi Kasus PT. Jalur Nugraha Ekakurir (JNE) Semarang)*. PhD Thesis. Universitas Negeri Semarang.

