

Penentuan Jarak Terpendek Menggunakan Metode Dijkstra Pada Data *Spasial Openstreetmap* (Studi Kasus : Pada Perusahaan Pengantaran Barang Wahana Logistik Kota Malang)

Yan Watequlis Syaifudin¹, Dwi Puspitasari², Muhammad Ashari Fajar Nugroho³

^{1,2,3} Politeknik Negeri Malang

¹qulis@polinema.com, ²dwi.puspitasari@gmail.com, ³ashari42@live.com

ABSTRAK

Openstreetmap adalah sebuah proyek berbasis *web* untuk membuat peta seluruh dunia yang gratis dan terbuka, dibangun sepenuhnya oleh sukarelawan dengan melakukan survei menggunakan GPS, mendigitasi citra satelit dan mengumpulkan serta membebaskan data geografis yang tersedia dipublik. Banyak jenis peta digital yang tersedia di internet, namun sebagian besar memiliki keterbatasan legalitas. Hal ini membuat *user* tidak dapat menggunakan peta tersebut secara bebas. Dan juga pada *openstreetmap* masih belum mendukung untuk pencarian jalur dengan *input* data lebih dari 1 tujuan. Wahana logistik merupakan salah satu jasa pengiriman logistik yang ada di kota Malang. Namun pada *openstreetmap*, lokasi wahana logistik masih belum ada sama sekali. Maka dari itu dibuatlah sebuah sistem penentuan jalur terpendek menggunakan algoritma dijkstra pada studi kasus wahana logistik. Sistem tersebut berupa *web* menggunakan *framework codeigniter*, dengan pertimbangan bahwa *web* tersebut dapat diakses oleh semua pihak dari wahana logistik tanpa membedakan platform yang digunakan oleh *user* tersebut. Metode dijkstra merupakan salah satu metode untuk menentukan jalur terpendek, namun sebelum masuk ke perhitungan metode, data jalur akan diolah menggunakan haversine formula untuk mengetahui jarak anatar *node*. Untuk proses pencarian jalur, sistem ini memberikan fasilitas berupa pencarian dengan beberapa *input* tujuan. Jadi jalur yang ditentukan bisa *flexible* dengan data *input* sesuai urutan data yang diinput oleh *user*.

Kata Kunci: Haversine Formula, Algoritma Dijkstra, Jalur Terpendek, *Codeigniter*

ABSTRACT

Openstreetmap is a web-based project to create a free and open worldwide map, built entirely by volunteers by conducting surveys using gps, digitizing satellite imagery and collecting and freeing available geographic data in the public. Many types of digital maps are available on the internet, but most have legality limitations. This makes the user unable to use the map freely. And also on *openstreetmap* still does not support to search the track with data input of more than 1 destination. Wahana Logistics is one of the logistics shipping services in Malang. But in *openstreetmap*, the location of the Wahana logistics is still not available at all. Therefore, a system for determining the shortest pathway was made using the Dijkstra algorithm at a Wahana Logistics case study. The system is in the form of a web using a *codeigniter* framework, with the consideration that the web can be accessed by all members of Wahana Logistics without differentiating platform used by user. The dijkstra is one of methods to determine shortest path, but before entering method calculation, path data will be processed using haversine formula to determine the distance between the node s. To process a path search, this system provides search facility with multiple input purpose. So, the resulting paths according with data entered by user.

Keywords: Haversine Formula, Dijkstra Algoritm, Shortest Path, *Codeigniter*

1. PENDAHULUAN

Era teknologi informasi saat ini mengalami perkembangan yang pesat. Faktor ketersediaan informasi dan kemudahan pengguna dalam mengakses informasi diterapkan hampir dalam semua bidang, salah satu teknologi yang dapat dimanfaatkan untuk mengatasi hal ini adalah penggunaan teknologi nirkabel. Tidak terkecuali dalam menyediakan peta atau rute. Pemetaan merupakan proses pengumpulan data untuk dijadikan sebagai langkah awal dalam pembuatan peta, dengan menggambarkan penyebaran kondisi alamiah tertentu secara meruang, memindahkan keadaan sesungguhnya kedalam peta dasar, yang dinyatakan dengan penggunaan skala peta [1]. Jalur

yang ada di kota Malang memiliki akses jalan yang baik namun bila ada pendatang yang masih awam dengan jalur di kota Malang, pendatang tersebut akan mengalami kesulitan. Maka dari itu dibutuhkan sebuah aplikasi untuk menentukan jalur terpendek guna memberikan informasi pada *user* yang masih awam mengenai jalur terpendek untuk menuju tujuan yang diinginkan oleh *user*. Untuk menentukan jalur tersebut dibutuhkan sebuah algoritma yang detail terperinci. Yaitu menggunakan algoritma dijkstra. Algoritma ini didasarkan pada representasi *adjacency matrix* untuk sebuah *graph*. Algoritma ini tidak hanya menemukan jalur terpendek dari satu *verteks* tertentu ke *verteks* lain, tetapi juga jalur terpendek dari *verteks* tertentu ke semua *verteks* lain. Algoritma ini sering digunakan

pada *routing*. Algoritma *dijkstra* mencari lintasan terpendek dalam sejumlah langkah [2]. Namun ada syarat untuk menggunakan algoritma *dijkstra* tersebut, yaitu titik *node* dan sebuah jalur antar *node* tersebut. Selain itu, dibutuhkan juga berapa jarak antar jalur tersebut. Maka dari itu dibutuhkan sebuah teori pengukuran jarak. Pada penelitian ini menggunakan teori haversine formula untuk proses pengukuran jarak. Teori Haversine Formula adalah sebuah persamaan yang penting dalam bidang navigasi, untuk mencari jarak busur antara dua titik pada bola dari *longitude* dan *latitude*. Ini merupakan bentuk persamaan khusus dari trigonometri bola, law of haversines, mencari hubungan sisi dan sudut pada segitiga dalam bidang bola [3].

OpenStreetMap (OSM) adalah sebuah proyek berbasis *web* untuk membuat peta seluruh dunia yang gratis dan terbuka, dibangun sepenuhnya oleh sukarelawan dengan melakukan survei menggunakan GPS, mendigitasi citra satelit, dan mengumpulkan serta membebaskan data geografis yang tersedia di publik. Melalui *Open Data Commons Open Database License 1.0*, kontributor OSM dapat memiliki, memodifikasi, dan membagikan data peta secara luas [4].

Wahana logistik merupakan salah satu jasa pengiriman logistik yang ada di kota Malang. Namun pada *openstreetmap*, lokasi wahana logistik masih belum ada sama sekali. Sistem tersebut berupa *web* dengan pertimbangan bahwa *web* tersebut dapat diakses oleh semua pihak dari wahana logistik tanpa membedakan *platform* yang digunakan oleh *user* tersebut. Dan juga *web* tersebut dapat dikembangkan dengan mudah. *Web* tersebut menggunakan *framework* Codeigniter.

CodeIgniter menjadi sebuah *framework* PHP dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi *web*. Selain ringan dan cepat, *CodeIgniter* juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodenya [5].

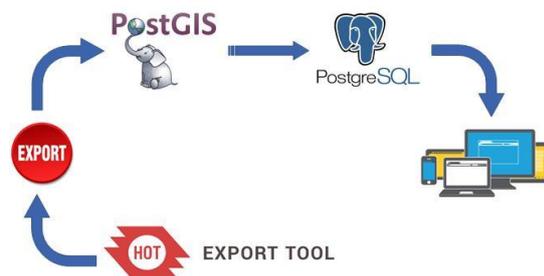
Untuk output dari sistem tersebut berupa jalur terpendek dari cabang wahana awal yang diinputkan oleh *user* dan cabang wahana tujuan sesuai yang diinputkan oleh *user*. Namun yang membedakan dari aplikasi pencarian jalur lainnya yaitu bahwa pencarian jalur pada aplikasi ini dapat langsung menentukan jalur dengan 1 lokasi tujuan maupun lebih.

2. METODOLOGI PENELITIAN

Pengumpulan Data

Untuk mendapatkan data uji penelitian ini maka penulis menggunakan cara yang telah dilakukan oleh OSM (*OpenStreetMap*) yaitu dengan cara mengekspor peta (*download*) sesuai dengan

daerah yang saya pilih yaitu Kota Malang. Namun sebagian data diperoleh dengan *input* secara manual oleh *Administrator*.

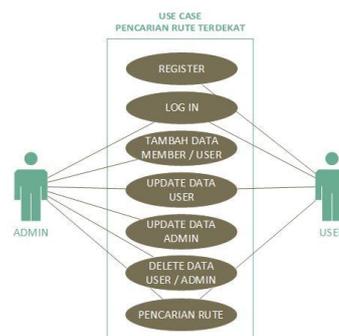


Gambar 1. Metode Pengumpulan Data

Openstreetmap merupakan sebuah map yang bersifat *open source*, yang berarti bahwa aplikasi tersebut membebaskan *user* untuk mengambil data dan merubah data pada map tersebut. Berbeda dengan aplikasi map dari *provider* lainnya. Pada aplikasi map selain *openstreetmap*, tidak ada yang bisa memberikan data secara *opensource* dikarenakan semua data diolah oleh admin secara keseluruhan. Berbeda dengan *openstreetmap*, yang keseluruhan data dapat diakses dan diolah oleh *user*.

Perancangan Sistem

Struktur menu pada aplikasi pencarian rute terdekat dengan menggunakan data spasial *Openstreetmaps* dirancang menggunakan bahasa pemrograman PHP. Diagram *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Berikut adalah gambar dari diagram *Use Case*.



Gambar 2. Use Case Diagram

Desripsi *use case* per-Sub-Sistem

- Admin
 1. Admin dapat *Login* dilanjutkan mengelola data.
 2. Admin dapat menambah data Admin dan *user*.
 3. Merubah data Admin maupun *User*.
 4. Menghapus data Admin maupun *User*.
 5. Mencari rute terdekat.
- *User*

1. *Register data user.*
2. *Login.*
3. *Merubah informasi user.*
4. *Mencari rute terdekat.*

Metode

Haversine Formula

Setelah data berhasil disimpan pada *database*, selanjutnya sistem akan mengambil *Latitude* dan *Longitude* dari titik awal sampai titik akhir, kemudian data diproses oleh perhitungan Geometri Haversine Formula, untuk mendapatkan berapa jarak antar *node* .

Dalam penerapannya terhadap bumi, haversine formula ini harus dikalikan dengan jari-jari dari lingkaran bumi yang nilainya 6371 km. Untuk nilai *latitude* dan *longitude* yang berbentuk derajat desimal maka harus di udah menjadi radians dengan cara mengkalikan nilai *latitude* dan *longitude* dengan 1 derajat atau 0.01745329251994 rad.

$$r = 6371 \text{ km}$$

$$d = 0.01745329251994$$

jarak

$$= 2 * 6371 * \text{ASIN}(\text{SQRT}(\text{SIN}((0.017453293 * (\text{Lat}2 - \text{Lat}1))/2)^2 + \text{COS}(\text{Lat}2 * 0.017453293) * \text{COS}(\text{Lat}1 * 0.017453293) * \text{SIN}(((0.01745329 * (\text{Long}2 - \text{Long}1))/2)^2)))$$

$$* \text{COS}(\text{Lat}1 * 0.017453293)$$

$$* \text{SIN}(((0.01745329 * (\text{Long}2 - \text{Long}1))/2)^2))$$

Algoritma Dijkstra

Setelah proses perhitungan Geometri Haversine Formula membuahkan hasil, selanjutnya akan diproses oleh Algoritma Dijkstra untuk menghitung semua kemungkinan jarak antar *node* yang akan dilewati dari titik awal menuju titik tujuan. Untuk prosesnya menggunakan logika seperti dibawah ini :

- a. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada *node* awal dan nilai tak hingga terhadap *node* lain (yang belum terisi).
- b. Set semua *node* “Belum terjamah” dan set *node* awal sebagai “*Node* keberangkatan”.
- c. Dari *node* keberangkatan, pertimbangkan *node* tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan.
- d. Setelah 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.
- e. Set “*Node* belum terjamah” dengan jarak terkecil (dari *node* keberangkatan) sebagai “*Node* Keberangkatan” selanjutnya dan lanjutkan dengan kembali ke step 3.

3. HASIL DAN PEMBAHASAN

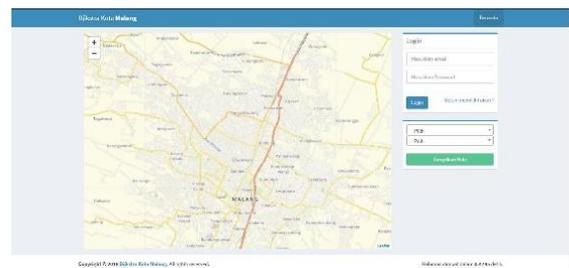
Tabel 1. Data Lokasi

Id_titik	Nama	Latitude	Longitude
L1	Wahana Prestasi Logistic Cabang Langsep	-7.97669	112.61275
L2	Wahana Prestasi Logistik Mojolangu	-7.94626	112,63737
L3	Ekspedisi Wahana Centra Niaga Ijen (Kasin-Tanjung-Mergan) Malang	-7,98336	112,61605
L4	Wahana Prestasi Logistik / OPM Express Logistic Kalpataru	-7,9486	112,62654
L5	wahana logistic Kauman Klojen	-7,97932	112,62948
L6	WAHANA EXPRES Gading Kasri, Klojen	-7,96978	112,61706
L7	Wahana Prestasi Logistik Kiduldalem, Klojen	-7,98132	112,63348

Pada Tabel 1 merupakan data lokasi dari cabang wahana logistik diarea kota Malang. Pada data tersebut memiliki atribut *id_titik*, atribut tersebut berfungsi sebagai kode unik dari tiap lokasi. Atribut nama merupakan atribut berisi nama dari tiap cabang wahana logistik. Atribut *lat* dan *long* berisi *latitude* dan *longitude* dari tiap lokasi.

Hasil Pengujian

Halaman Beranda



Gambar 3. Halaman Beranda

Gambar Diatas menggambarkan rancangan awal. Halaman tersebut digunakan sebagai profil halaman awal sistem.

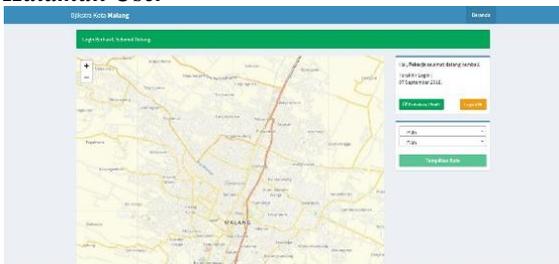
Halaman Register User



Gambar 4. Halaman Register User

Gambar diatas merupakan halaman dimana *user* melakukan pendaftaran dan diwajibkan untuk mengisi 4 kolom, meliputi kolom nama, alamat email, nomor *handphone user* dan *password*.

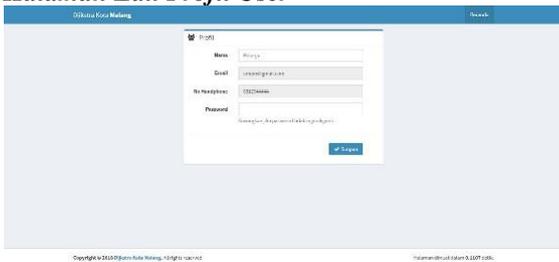
Halaman User



Gambar 5. Halaman User

Gambar diatas merupakan tampilan awal setelah *user* melakukan *login*, hampir sama dengan beranda, namun yang membedakan adalah pada *form login* berubah menjadi sebuah kolom yang berisi selamat datang pada *user* dan pemberitahuan kapan terakhir *user* tersebut *login*.

Halaman Edit Profil User



Gambar 6. Halaman Edit Profil User

Gambar diatas adalah sebuah halaman untuk *user* melakukan *editing* data, namun yang dapat dirubah sebatas nama dan *password* saja.

Halaman Beranda Admin



Gambar 7. Halaman Beranda Admin

Gambar diatas adalah halaman *dashboard* yang berarti bahwa halaman tersebut merupakan halaman awal setelah admin melakukan *login*.

Halaman Admin – Manajemen User



Gambar 8. Halaman Admin – Manajemen User

Gambar diatas merupakan halaman *admin* untuk mengelola data *user*. Pada halaman tersebut admin dapat mengelola data dari *user* tersebut yaitu untuk menambahkan *user* baru, menghapus *user* dan merubah isi dari data *user* tersebut.

Halaman Admin – Manajemen Admin



Gambar 9. Halaman Admin – Manajemen Admin

Gambar diatas merupakan halaman *admin* untuk mengelola data *user*. Pada halaman tersebut admin dapat mengelola data dari *user* tersebut yaitu untuk menambahkan *user* baru, menghapus data *user* dan merubah isi dari data *user* tersebut.

Halaman Admin – Profil



Gambar 10. Halaman Admin – Profil

Gambar diatas merupakan informasi profil dari admin itu sendiri, admin dapat merubah isi dari profil tersebut. Dan admin juga dapat *upload* foto dari admin itu sendiri.

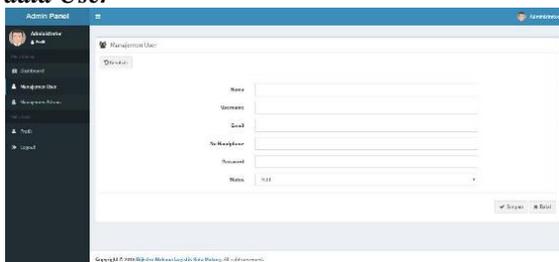
Halaman Admin – Edit Profil



Gambar 11. Halaman Admin – Edit Profil

Gambar diatas adalah sebuah halaman untuk admin melakukan *editing* data, namun yang dapat dirubah sebatas nama dan *password* saja.

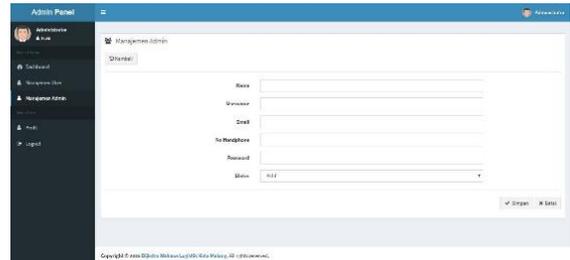
Halaman Admin – Manajemen User – Tambah data User



Gambar 12. Halaman Admin – Manajemen User – Tambah data User

Gambar diatas merupakan halaman dimana admin melakukan pendaftaran untuk *user* baru dan diwajibkan untuk mengisi 6 kolom, meliputi kolom nama, *username*, alamat email, nomor *handphone user* dan password dan status *user*.

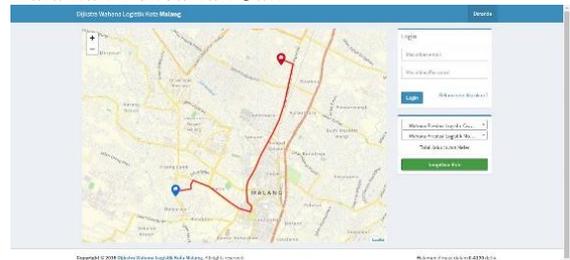
Halaman Admin – Manajemen Admin – Tambah data Admin



Gambar 13. Halaman Admin – Manajemen Admin – Tambah data Admin

Gambar diatas merupakan halaman dimana admin melakukan pendaftaran untuk admin baru dan diwajibkan untuk mengisi 6 kolom, meliputi kolom nama, *username*, alamat email, nomor *handphone user* dan *password* dan status *user*.

Halaman Pencarian Jalur



Gambar 14. Halaman Pencarian Jalur

Pada halaman diatas merupakan hasil dari pencarian jalur terpendek menggunakan algoritma dijkstra. Sebagai contoh mencari jalur dengan titik awal pada pasar Blimbing dan titik tujuan ada pada Kelurahan Tunggulwulung.

Hasil Analisa

Hasil analisa penentuan jalur terpendek menggunakan data spasial *openstreetmap* pada studi kasus wahana logistik menghasilkan sebuah jalur. Jalur tersebut didapatkan dari *user* setelah memberikan *inputan* titik awal menuju titik tujuan. Dan *user* dapat memilih lebih dari 1 tujuan dengan berkesinambungan. Untuk tujuannya adalah antar kantor wahana logistik yang berada di Kota Malang saja. Aplikasi tersebut berjalan lancar pada *web browser* apapun tanpa membedakan *platform* yang digunakan oleh *user*



Gambar 15. Hasil Perhitungan Dijkstra

Gambar merupakan hasil dari perhitungan dijkstra, sebagai contoh *user* tersebut memberikan *input* dari lokasi L1 menuju L2 dan L2 menuju L3. dengan hasil berupa jalur terpendek dari L1 menuju L2 dilanjutkan dari L2 menuju L3.

4. KESIMPULAN DAN SARAN

Kesimpulan yang dapat diambil dari laporan Penelitian dengan judul Pencarian Rute Terdekat dengan menggunakan Data Spasial *Openstreetmaps* yaitu :

1. Sistem dapat mencari rute dengan baik menggunakan Algoritma Dijkstra, namun sebelum mencari rute tersebut dibutuhkan Algoritma Geometri Haversine Formula yang berfungsi untuk menghitung antar *latitude* dan *longitude* satu dengan lainnya.
2. Cara kerja sistem tersebut adalah memprioritaskan tujuan yang dipilih oleh *user*, kemudian untuk menentukan jalur selanjutnya akan dimulai dari lokasi yang sudah ditentukan sebelumnya, dan proses tersebut akan berkelanjutan pada pencarian jalur menuju titik tujuan *user* sesuai dengan prioritas yang diinginkan *user*.
3. Dapat mempermudah *user* untuk mencari jalur yang akan dituju, khususnya bagi petugas wahana logistik area kota Malang

Berdasarkan hasil dari 16 kuisisioner yang diberikan kepada kurir Wahana Logistik, menyatakan bahwa kurir Wahana Logistik menyatakan puas dengan tingkat kepuasan rata-rata sebesar 3,8. Kuisisioner tersebut menggunakan penilaian skala 1 sampai 5, dengan keterangan bahwa nilai 1 = sangat tidak puas, 2 = tidak puas, 3 = cukup puas, 4 = puas, 5 = sangat puas.

Saran dari penulis, untuk penelitian kedepannya perlu dikembangkan pengolahan data spasial pada peta OSM dan memperluas fitur. Serta bagi pengembang dapat mengembangkan peta *openstreetmaps* terutama untuk pencarian jalur terpendek dan ketepatan lokasi dari tiap *node*. Dengan adanya pengembangan tersebut diharapkan lebih memudahkan pihak pengguna atau masyarakat untuk mendapatkan informasi mengenai jalur terpendek.

5. REFERENSI

- [1] A. Rizal, S. A. S. Mola, T. Widiastuti, J. I. Komputer, and U. N. Cendana, "Penerapan Algoritma Dijkstra Pada Permasalahan Lintasan," *Jicon*, vol. 2, no. 1, pp. 1–9, 2014.
- [2] A. W. R. Emanuel and A. Wiraguna, "Aplikasi Perencanaan Perjalanan Kota Bandung dengan Memanfaatkan OpenStreetMap XML dan .NET Compact Framework di PDA dengan Teknologi Pencarian Rute Algoritma A*," *J. Inform.*, vol. 3, no. 1, 2007.
- [3] A. Adiwilaga, "Teori Pengukuran Jarak," 2014. [Online]. Available: <https://blogs.itb.ac.id/anugraha/2014/09/10/teori-pengukuran-jarak/>.
- [4] N. S. Kanthi, "Aplikasi OpenStreetMap (OSM) untuk Mendukung Pemetaan Desa," Universitas Gadjah Mada, 2016.
- [5] P. C. H. Indonesia, "Mengenal Framework Codeigniter," *Indonesia, PT Cloud Hosting*, 2018. [Online]. Available: <https://idcloudhost.com/panduan/mengenal-apa-itu-framework-codeigniter/>.