

Sistem Informasi Geografis Pencarian Rute Terdekat Bengkel Motor di Kota Surabaya Menggunakan Algoritma *Bellman-Ford*

Viery Lovega Pratama¹, Dodik Arwin Darmawan²

^{1,2} Jurusan Teknik Informatika, Prodi Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya
viery.18033@mhs.unesa.ac.id
dodikdermawan@unesa.ac.id

Abstrak— Setiap harinya kendaraan bermotor digunakan oleh pemiliknya untuk bekerja atau melakukan perjalanan ke suatu tempat. Kenyamanan dalam berkendara didapatkan dengan melakukan perawatan rutin *service*. Banyak kecelakaan kendaraan bermotor salah satunya diakibatkan kurangnya perawatan dari pemiliknya. Ketika kendaraan dalam keadaan mogok mendadak untuk sementara waktu bisa dibawa ke bengkel alternatif atau bengkel tidak resmi sebelum dibawa ke bengkel *service* resmi agar motor dapat kembali sesuai fungsinya. Dengan pencarian terdekat menuju bengkel motor disekitar tempat maka diperlukan sebuah jarak atau lintasan minimum agar cepat sampai tujuan serta tidak membuang banyak waktu serta tenaga yang dikeluarkan. Salah satu algoritma yang bisa digunakan dalam mencari rute terdekat atau jalur minimum adalah dengan algoritma pencarian rute terpendek. Salah satunya menggunakan algoritma *Bellman-Ford*. Penelitian ini bertujuan untuk membantu pengguna untuk menemukan bengkel motor terdekat di Kota Surabaya berbasis *website* dengan implementasi dari algoritma *Bellman-Ford*. Selama uji coba algoritma dibandingkan dengan penghitungan secara manual, dengan membuat *graf* dan diuji per iterasi sampai dengan titik akhir tujuan. Sehingga didapat rute terdekat atau jarak minimum yang pengguna bisa gunakan untuk menuju lokasi bengkel tersebut.

Kata Kunci— Bengkel, Rute, *Bellman-Ford*, algoritma, *Website*, *Shortest Path*.

I. PENDAHULUAN

Setiap harinya kendaraan bermotor digunakan oleh pemiliknya untuk bekerja atau melakukan perjalanan ke suatu tempat. Kenyamanan dalam berkendara didapatkan dengan melakukan perawatan rutin *service*, semua orang menginginkan nyaman dalam berkendara di jalan. Jumlah pemakai jalan di Kota Surabaya di dominasi oleh pengguna kendaraan bermotor. Banyak kecelakaan kendaraan bermotor yang diakibatkan salah satunya kurang perawatan oleh pemilik. Ketika kendaraan dalam keadaan mogok mendadak untuk sementara waktu bisa dibawa ke bengkel alternatif atau bengkel

tidak resmi sebelum dibawa ke bengkel *service* resmi agar motor dapat kembali sesuai fungsinya.

Pendatang dari luar Kota Surabaya jika terdapat kerusakan atau mogok di tengah jalan pada saat berkendara kurang mengetahui daerah sekitar, maka untuk mencari bengkel motor terdekat biasanya dilakukan dengan menyusuri setiap jalan atau bertanya kepada penduduk sekitar. Dalam mencari bengkel terdekat harus memastikan bahwa jarak yang dilalui untuk menyusuri bengkel tersebut diperlukan rute pencarian bengkel yang singkat dan terdekat. Permasalahan yang berhubungan dengan rute terdekat untuk menemukan jalur minimum adalah bagian dari teori *graf*, untuk menyelesaikan *graf* dengan adanya nilai dan berbobot dengan rute terdekat dengan meminimalkan jumlah bobot yang membentuk jalur pada setiap sisinya dengan memberikan *graf* yang berbobot[4].

Himpunan titik-titik ditandai ditandai alfabet dengan simpul yang saling terhubung (*vertex* atau *node*) pada setiap sisinya (*edge*) atau busur (*arc*) yang saling terhubung dengan lainnya hingga titik akhir merupakan pengertian dari *graf*.

titik - titik (lambang simpul) yang saling berkumpul lalu terhubung dengan garis-garis (lambang sisi) atau garis berpanah (lambang busur) dengan adanya titik awal sampai titik akhir merupakan gambaran dari *graf*[16].

Kegiatan untuk memperbaiki suatu benda atau melakukan kontruksi di dalam sebuah bangunan dengan peralatan merupakan pengertian dari Bengkel. Ilmu atau keterampilan dalam memperbaiki, mengubah bentuk, melakukan modifikasi benda dan merakit suatu benda yang baru atau benda lama menjadi lebih baik secara keindahan maupun fungsi merupakan pengertian dari perbengkelan. Tempat untuk memperbaiki maupun merubah bentuk motor agar dalam kembali berjalan baik atau sesuai dengan fungsi motor dengan baik

dan semestinya semestinya merupakan pengertian dari bengkel motor. Spare parts yang digunakan dalam perbaikan sepeda motor dapat menggunakan spare parts baru maupun *second* dengan melakukan modifikasi tertentu agar sepeda motor berjalan dengan baik [9].

Metode atau langkah yang tersusun dan runtut didalam penyelesaian masalah dengan sebuah tugas termasuk instruksi atau kegiatan yang ada merupakan pengertian dari algoritma. Serangkaian langkah yang runtut dan sistematis dalam menyelesaikan masalah dibantu dengan bantuan komputer dalam bidang pemrograman merupakan definisi dari algoritma [21].

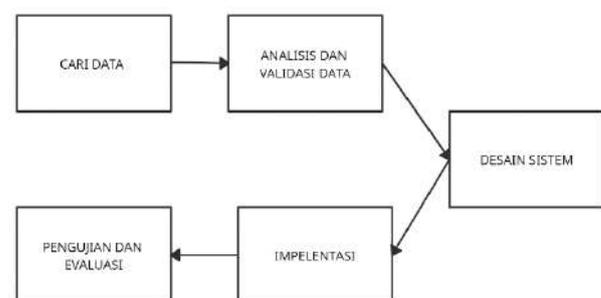
Pengembang dan penemu algoritma *Bellman-Ford* adalah 2 orang, Richard Bellman dan Lester Ford. Algoritma *Dijkstra* dan algoritma *Bellman-Ford* dalam penggunaannya sangat mirip namun kelebihan dari algoritma *Bellman-Ford* mampu menyelesaikan bobot negatif untuk mencari jalur terpendek dalam sebuah *graf*. Salah satu kelebihan dari algoritma *Bellman-Ford* jika ditemukan suatu *graf* tidak adanya *cycle* berbobot negatif untuk menuju ke sumber atau titik tujuan akhir maka algoritma dapat dinyatakan benar [1]. Jika keadaan tidak terdapat sisa *graf* yang berbobot negatif dan memiliki proses yang lebih cepat dibandingkan dengan algoritma *Bellman-Ford* merupakan perkembangan dari penggunaan algoritma *Dijkstra*. Untuk melakukan perhitungan biaya lintasan atau rute terdekat dari semua simpul yang ada sampai ke simpul lain yang terhubung hingga ke titik akhir pada *graf* dapat menggunakan algoritma *Bellman-Ford*. Ciri-ciri dari penggunaan algoritma *Bellman-Ford* yaitu: *edge* dengan bobot negatif tetap dapat diselesaikan, harus *directed edge* (pada *graf* yang tidak memiliki *cycle* dengan bobot negatif), seluruh rute terdekat (*shortest path*) dapat ditemukan dengan menggunakan Iterasi ke I sampai ke *i edge*, deteksi *cycle* menggunakan beban berbobot negatif jika ada [8].

Sebagaimana dengan penelitian ini penulis akan membuat sistem informasi geografis pencarian rute terdekat bengkel motor di Kota Surabaya menggunakan algoritma *Bellman-Ford* berbasis *website* untuk menyelesaikan masalah yang sudah dipaparkan diatas. Dengan keterbaruan dan

kehandalan dibantu *tools package* terbaru dari *LeafletJs*. Salah satunya dengan fitur *cluster* untuk mempermudah dalam desain dan pengelompokkan marker. Selain itu dengan *tools package* dari *Liedman* membantu dalam *tracking data* dan *routing machine*, untuk pengguna dalam mencari lokasi terdekat. Keunggulan dari *website* sifatnya yang mudah digunakan serta dapat diakses dimana saja sehingga sangat cocok untuk membantu aktifitas yang memiliki mobilitas dan *transfer* data secara cepat. Dengan *website* ini dapat mempermudah untuk mencari bengkel motor terdekat, selain itu juga dapat membantu penyedia bengkel motor mendapatkan pelanggan.

II. METODE PENELITIAN

Penelitian ini terdapat beberapa tahapan yang diperlukan agar berjalan dengan baik, maka dari itu diperlukan diagram alur. Penelitian ini mengacu pada diagram alur berikut yang terdapat pada Gbr 1. Alur penelitian dimulai dengan pencarian data, lalu analisis dan validasi data, setelah itu desain sistem, beranjak ke implementasi dan proses yang terakhir adalah pengujian dan evaluasi.



Gbr 1. Diagram alur metode penelitian

A. Cari Data

Untuk pencarian data sebagai kebutuhan penelitian diambil dari referensi yang ada. Dalam pengembangannya sebagai sistem informasi geografis data bengkel motor dan pemecahan per kecamatan. Oleh karena itu peneliti membagi pengumpulan data menjadi dua jenis, yaitu studi literatur dan observasi.

a. *Studi literatur*

Referensi yang digunakan dalam studi literatur diambil dengan mempertimbangkan data yang relevan dengan menggunakan data kecamatan Kota Surabaya. Data yang digunakan berjumlah 31 kecamatan [22]. Didalamnya ditambahkan 80 data bengkel motor secara manual di Kota Surabaya. Data dalam yang dipakai dalam penelitian ini termuat dalam internet.

b. *Observasi*

Selain studi literatur yang telah dibahas sebelumnya, peneliti juga melakukan observasi dengan terjun ke lapangan langsung untuk berkunjung ke beberapa bengkel motor di Kota Surabaya untuk mengetahui keberadaan dan lokasinya.

B. *Analisis Data*

Setelah data didapatkan pada tahap cari data. Tahap berikutnya adalah menganalisis data dan validasi data. Tujuan menganalisis dan validasi data adalah untuk melihat dan mencocokkan data yang di dapat di lapangan dengan literatur apakah sudah benar, Sesuai dengan alamat maupun pada letak *latitude* dan *longitude*. Data atribut yang akan digunakan pada penelitian ini dapat dilihat pada Tabel I.

TABEL I
TABEL DATA ATRIBUT

| Atribut | Tipe Data | Kegunaan |
|-----------------|-----------|--|
| Pengguna | Karakter | Untuk masuk dalam WebGIS |
| Nama Kecamatan | Karakter | Untuk mengenali nama kecamatan |
| Kode Kecamatan | Numerik | Diambil dari website resmi, sebagai Kode Kecamatan |
| Batas Kecamatan | Geojson | Sebagai penanda batas – batas antar kecamatan |

| | | |
|----------------|----------|--|
| Nama Bengkel | Karakter | Untuk memberi nama pada bengkel motor |
| Lokasi Bengkel | Geojson | Penanda lokasi <i>hotspot</i> atau <i>marker</i> bengkel motor |

c. *Desain Sistem*

a. *Tools Pengembangan Aplikasi*

Untuk membantu peneliti dalam mewujudkan aplikasi, terdapat beberapa *tools* yang dibutuhkan pada Gbr 2. Tools untuk pengembangan aplikasi



Gbr.2 Tools untuk pengembangan aplikasi

Pengembangan aplikasi terdapat 3 tools utama yaitu :

- a. Bootstrap
- b. XAMPP
- c. Php-MySQL

Tools Pertama yang dibutuhkan adalah Bootstrap. Bootstrap sangat berguna untuk peneliti dalam pengembangan aplikasi khususnya pada bagian front end dengan bebas (*framework open-source*). Dengan Mark Otto dan Jacob Thornton sebagai pembuat dari bootstrap. Untuk membuat situs *website* yang berjalan dengan responsif maka diperlukan bootstrap. Bootstrap juga dapat bekerja optimal untuk layar *smartphone* ataupun *laptop* pada *Interface website* atau *UI web* [18].

Untuk tools kedua dalam pengembangan aplikasi sistem informasi geografis diperlukan XAMPP. Untuk pembuatan *website* dalam dunia *web developer*

XAMPP merupakan sebuah aplikasi untuk *web server open source* (bebas). Dalam penggunaannya XAMPP sudah mendukung berbagai sistem operasi seperti, Linux, Windows, Mac OS, dan Solaris. Dalam mengganti fungsi dari *web hosting* dikarenakan mahal dan terdapat *expired date* maka diperlukan XAMPP karena dapat langsung dipanggil dalam *web browser* dengan mengetahui lokasi file. Tools XAMPP berada di naungan dan di *maintenance* oleh Apache Friends mulai tahun 2002, yang bisa didapatkan (*download*) secara gratis pada *website* resmi. tentunya sangat berguna untuk penelitian ini dengan penanda GNU (General Public License) [17].

Hypertext Preprocessor merupakan kepanjangan dari pemrograman bahasa komputer PHP, Bahasa pemrograman ini kurang lebih mirip dengan JavaScript(Js) dan Python(Py). Perbedaannya terdapat pada fungsi tiap bagian, untuk kebutuhan komunikasi kepada server maka dibutuhkan PHP, sedangkan untuk kebutuhan *user interface(frontend)* dan *backend (database dan web service)* maka dibutuhkan JavaScript. Sementara itu, Python digunakan terkadang untuk server (backend). Rasmus Lerdorf merupakan pencipta bahasa pemrograman PHP awalnya digunakan untuk memantau pengunjung yang mengunjungi *home page*. Seiring waktu, Bahasa pemrograman ini semakin populer, dan seiring berjalannya waktu, Lerdorf akhirnya merilis sebagai proyek open-source (semua orang boleh dalam bergabung untuk pengembangan bahasa PHP). Tahun demi tahun Para pengembang yang bergabung pun mulai memodifikasi, memperbaiki, dan menyempurnakan kode ini kedalam bentuk yang lebih mudah dan, hingga kemudian menjadi Bahasa penulisan skrip yang hingga kini banyak digunakan [20].

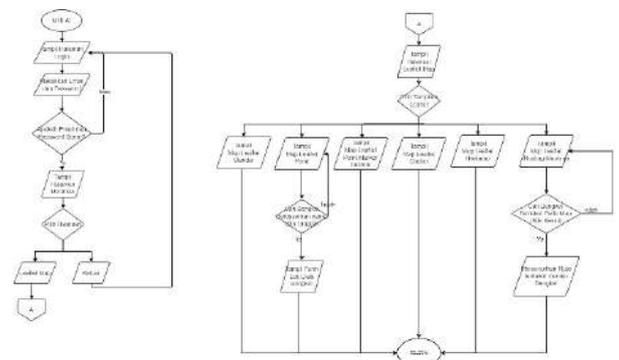
Tools selanjutnya yang digunakan untuk penelitian merupakan MySQL. MySQL sendiri digunakan untuk DBMS, DBMS sendiri merupakan kepanjangan dari (*Database Management System*) untuk *backend* pada pengembangan aplikasi ini. Untuk menggunakan MySQL perlu memahami perintah dan bahasa pemrograman SQL. Terdapat dua lisensi pada MySQL, lisensi pertama adalah *Free Software* dimana perangkat lunak dapat

digunakan untuk pengembangan oleh siapapun. Dan lisensi kedua MySQL adalah *Shareware* perangkat lunak memiliki pemilik serta memiliki batasan dalam penggunaannya.

MySQL juga termasuk dalam RDBMS. RDBMS memiliki kepanjangan (*Relational Database Management System*). Di dalam MySQL terdapat kolom, baris, serta tabel di dalam struktur database. Dalam pengambilan data pada MySQL menggunakan *relational database*, serta memiliki fungsi lain jembatan antara perangkat lunak (*software*) dan *database server*. Secara garis besar dan inti kegunaan MySQL melakukan manajemen dan melakukan olah data pada sisi server yang memiliki semua informasi serta data dengan menggunakan bahasa pemrograman SQL [19].

b. *Flowchart dan UML Diagram*

Dalam pengembangan aplikasi ini serta untuk memudahkan pengguna memahami alur aplikasi, maka dibuat perancangan alur dan prosesnya. seperti *flowchart dan usecase Diagram*.



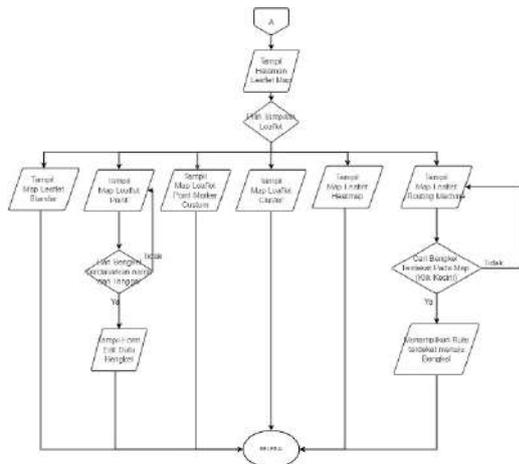
Gbr 3. Flowchart untuk Pengunjung

Pada Gbr 3. Dijelaskan alur program atau *flowchart* dari pengembangan aplikasi sistem informasi geografis bengkel motor untuk pengunjung atau *user*. Tampilan dari hak level untuk pengguna atau pengunjung ditunjukkan pada map dan detail data bengkel motor pada map dengan terperinci.

Sebelum menampilkan halaman beranda awal sistem diminta untuk pengguna atau pengunjung memasukkan data diri untuk masuk terlebih dahulu.

Jika berhasil maka akan muncul tampilan beranda, jika tidak maka akan dikembalikan pada halaman login dengan notifikasi salah untuk memasukkan email dan password.

Jika user benar dialihkan ke beranda, dengan ditambahkan hak level user untuk menandai bahwa pengguna yang masuk merupakan user ataupun nantinya admin dibedakan dengan penanda disamping nama dan fitur yang ada. setelah itu pengunjung dapat menggunakan aplikasi *website* bengkel motor dengan beberapa fitur yang ada untuk pengunjung yaitu ada 2 akses : leaflet map dan keluar.



Gbr. 4 Flowchart pengunjung akses Leaflet

Setelah pengunjung melewati akses login dan telah berada pada halaman beranda. sistem akan melakukan kalkulasi jumlah kecamatan dan marker. Sistem menampilkan data yang ada pada setiap halaman. Ada beberapa fitur sebanyak 6 fitur yang dapat diakses oleh pengguna yang berkaitan dengan sistem informasi geografis, beberapa halaman dijelaskan pada Gbr 4.

Pengunjung dapat mengakses 6 halaman map yang tersedia seperti halaman leaflet standar, leaflet point, leaflet point marker custom, leaflet cluster, leaflet heatmap, leaflet routing machine.

Pada menu leaflet map sidebar pilihan pertama terdapat standar. Saat pengunjung menekan klik navigasi pada sidebar standar maka akan dialihkan ke halaman standar. Halaman standar berisi sistem informasi geografis map Kota Surabaya dengan

Kecamatan yang dibagi dengan perbedaan warna terdapat 31 kecamatan.

Pada menu leaflet map sidebar pilihan ke-2 terdapat point. Ketika pengunjung menekan klik navigasi pada sidebar point maka akan dialihkan ke halaman point. Halaman point berisi sistem informasi geografis map Kota Surabaya dengan Kecamatan yang dibagi dengan perbedaan warna terdapat 31 kecamatan ditambah dengan data *marker* bengkel motor serta dibantu dengan fitur pencarian berdasarkan nama dan tanggal bengkel tersebut ditambahkan oleh admin untuk mempermudah pencarian oleh pengunjung.

Pada menu leaflet map sidebar pilihan ke-3 terdapat point marker custom Ketika pengunjung menekan navigasi pada sidebar point marker custom maka akan dialihkan ke halaman point marker custom. Halaman point marker custom berisi sistem informasi geografis map Kota Surabaya dengan Kecamatan yang dibagi dengan perbedaan warna terdapat 31 kecamatan ditambah dengan data *marker* bengkel motor yang dapat diedit dengan berbagai *icon* atau *marker* yang telah ditambahkan oleh admin untuk mempermudah dalam pengunjung menandai bengkel motor tertentu atau menjadi ciri khas suatu bengkel.

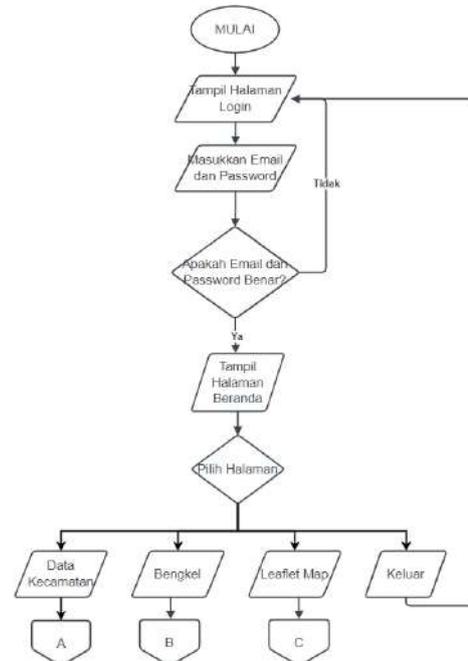
Pada menu leaflet map sidebar pilihan ke-4 terdapat cluster Ketika pengunjung menekan klik navigasi pada sidebar cluster maka akan dialihkan ke halaman cluster. Halaman cluster berisi sistem informasi geografis map Kota Surabaya dengan Kecamatan yang dibagi dengan perbedaan warna terdapat 31 kecamatan ditambah dengan pengelompokkan data *marker* bengkel motor yang dikelompokkan, Ketika di zoom maka akan memecah ke dalam bentuk yang lebih kecil. Cluster dapat memperindah tampilan dan mempermudah user dalam menggunakan dan menghitung berapa jumlah titik bengkel motor yang ada.

Pada menu leaflet map sidebar pilihan ke-5 terdapat heatmap Ketika pengunjung menekan klik navigasi pada sidebar heatmap maka akan dialihkan ke halaman heatmap. Halaman heatmap berisi sistem informasi geografis map Kota Surabaya dengan Kecamatan yang dibagi dengan perbedaan warna terdapat 31 kecamatan ditambah dengan data

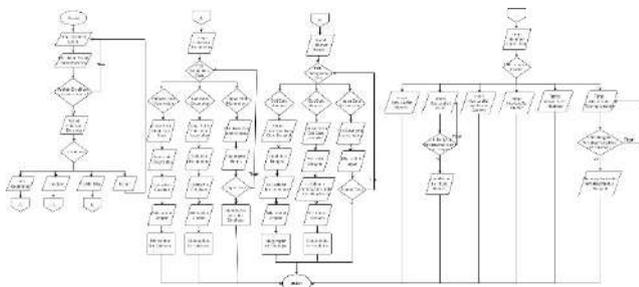
bengkel motor di tandai dengan bulatan merah bercampur dengan kuning dan biru untuk menandai lokasi mana yang terdapat banyak titik ditandai dengan berkumpul dan menyatunya titik tersebut.

Pada menu terakhir pada leaflet map sidebar pilihan ke-6 terdapat routing machine. Ketika pengunjung menekan navigasi pada sidebar routing machine maka akan dialihkan ke halaman routing machine. Halaman routing machine berisi sistem informasi geografis map Kota Surabaya dengan Kecamatan yang dibagi dengan perbedaan warna terpadat 31 kecamatan ditambah dengan data *marker* bengkel motor yang dapat diedit dengan berbagai *icon* atau *marker* yang telah ditambahkan oleh admin. Fungsi utama dari halaman ini tentunya untuk menentukan rute terpendek atau jalur terpendek untuk pengunjung menuju ke bengkel terdekat, untuk dapat menggunakan maka pengunjung dapat menentukan titik awal dimana untuk memulai perjalanan pada sisi kanan pojok atas lalu setelah itu dengan memilih lokasi bengkel mana yang akan dituju maka klik “Rute Menuju Lokasi ini?” maka setelah itu akan muncul rute yang dapat dilalui untuk menuju ke lokasi bengkel tersebut.

password nya dan dapat melakukan edit email dan password data sendiri maupun data user yang ingin diganti. Dikarenakan fitur pada halaman leaflet map sama dengan pengunjung. Admin dapat mengakses halaman tersebut, maka disini hanya akan dijelaskan pada penambahan, edit, serta hapus data pada kecamatan & bengkel.



Gbr. 6 Flowchart Login Admin

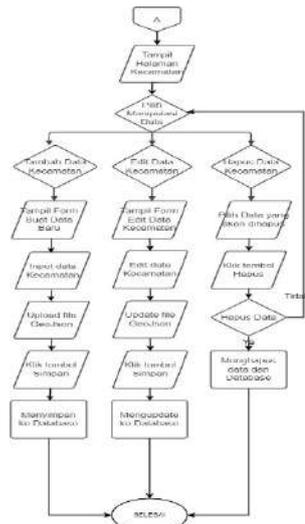


Gbr. 5 Flowchart untuk Admin

Pada Gbr. 5 diatas flowchart untuk admin. Admin memperoleh semua hak akses beserta fitur yang ada pada. Akun admin dapat didaftarkan secara manual dengan mendapatkan hak level admin yang dapat diatur.

Setelah admin dapat login dengan memasukkan data email dan password maka akan dialihkan pada halaman beranda sama seperti pengunjung. Dan jika password atau email salah admin dapat melihat dan mengakses database untuk mengetahui email dan

Admin juga mendapatkan hak akses penuh terdapat fitur yang tersedia pada *website* sistem informasi geografis ini seperti melakukan tambah data, edit data, hapus data di kecamatan maupun pada lokasi dan titik bengkel serta leaflet map. Tentunya hak level ini sangat diperlukan karena tidak semua orang dapat melakukan penambahan, maupun edit serta menghapus data tanpa hak sebagai admin, dikarenakan nantinya akan merusak data yang telah ada sebelumnya.



Gbr. 7 Flowchart CRUD Kecamatan

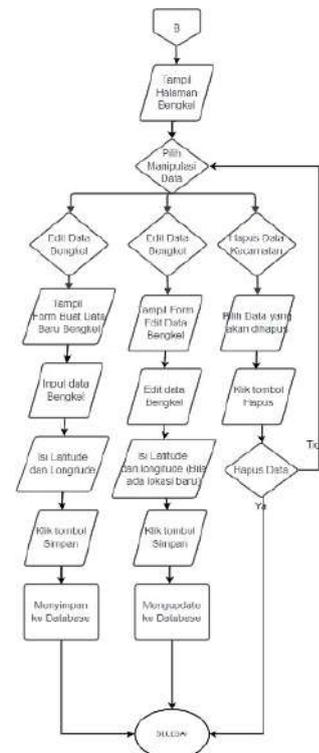
Pada Gbr.7 merupakan flowchart admin untuk dapat melakukan tambah data kecamatan, edit data kecamatan, serta menghapus data kecamatan. Hak level ini dibedakan karena admin mempunyai hak untuk melakukan itu jika terdapat batasan kecamatan yang dirasa kurang maupun misalkan nantinya terdapat kecamatan baru, atau bisa digunakan untuk pengembangan pada kota lain jika penelitian ini diteruskan.

Data kecamatan baru dapat ditambahkan oleh admin, Ketika tombol diklik maka sistem akan melakukan proses dan tampil *form* pada *user interface* data kecamatan yang baru akan ditambahkan yang terdiri kode kecamatan, nama kecamatan, file *GeoJson*, serta warna untuk membedakan dan memberi batas antar kecamatan. Kode pengelolaan didapatkan pada *web* resmi Kota Surabaya dan *Wikipedia* yang terdaftar atau terverifikasi. File *GeoJson* didapatkan dari pemetaan dalam *ArcGIS* secara manual atau bisa didapat pada laman internet dengan mengacu pada gambar maps. Setelah itu admin dapat menekan tombol simpan data, maka sistem akan menyimpan data yang baru ditambahkan tersebut kedalam database.

Melakukan update data bisa dilakukan melalui tombol Ubah oleh admin. Jika di klik maka akan dialihkan menuju *form* edit data. Data asli yang sebelumnya telah disimpan akan ditampilkan melalui form tersebut, apabila ada perubahan dapat admin memanipulasi data secara langsung. Setelah

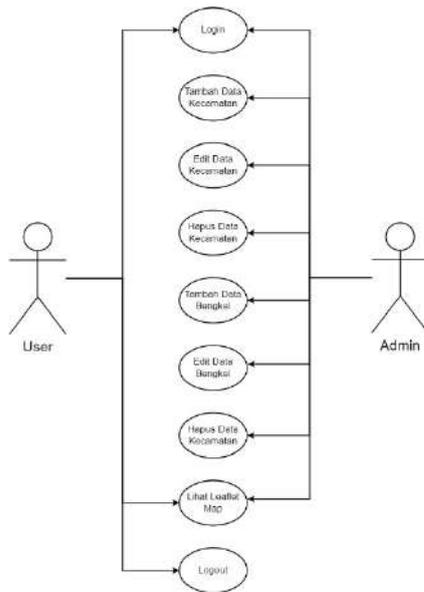
itu, Ketika admin menekan tombol simpan maka sistem akan memperbarui data tersebut kedalam database.

Admin dapat melakukan hapus data. Ketika menekan tombol hapus pada data kecamatan yang perlu untuk dihapus. Kemudian sistem akan menampilkan notifikasi konfirmasi untuk menghapus data, tekan *cancel* untuk membatalkan aksi hapus dan tekan ok untuk memproses hapus data yang diperlukan. Sistem akan menghapus data yang telah dipilih pada kecamatan tersebut di database.



Gbr. 8 Flowchart CRUD Bengkel

Pada Gbr.8 merupakan flowchart admin untuk dapat melakukan tambah data kecamatan, edit data kecamatan, serta menghapus data kecamatan. Hak level ini dibedakan karena admin mempunyai hak untuk melakukan itu jika terdapat batasan kecamatan yang dirasa kurang maupun misalkan nantinya terdapat kecamatan baru, atau bisa digunakan untuk pengembangan pada kota lain jika penelitian ini diteruskan.



Gbr. 9 Usecase diagram

Pada Gbr.9 diatas dijelaskan sistem yang dibuat ini terdapat dua actor, yaitu: pengunjung dan admin. Pengunjung merupakan pengguna yang menggunakan aplikasi diberikan hak level sebagai user, pengunjung hanya dapat melihat tampilan halaman leaflet map dengan 6 fitur yang tersedia. Sedangkan admin dapat melakukan login dengan hak level sebagai admin. Hak level admin memiliki kuasa penuh untuk CRUD (*create, read, update, delete*) atau disebut dengan manipulasi data kecamatan dan bengkel motor. Dikarenakan admin bertanggung jawab penuh untuk pemeliharaan serta perbaikan *web* jika nantinya akan *update* secara *back-end* maupun *front-end*.

D. Implementasi

Setelah melewati beberapa tahapan, mulai dari mencari data sampai dengan desain sistem. Tahap selanjutnya implementasi semua hasil rancangan dan desain sistem yang telah dibuat. Perancangan *website* ini menggunakan PHP sebagai bahasa pemrograman dengan sistem informasi geografis menampilkan map dan *marker* serta *routing machine* di Kota Surabaya sesuai dengan data penelitian yang telah didapat. Untuk membantu penyimpanan data serta manipulasi data, maka dibutuhkan suatu basis data. MySQL digunakan untuk melakukan basis data. Perhitungan dalam aplikasi antara lain uji

kinerja algoritma dan kesesuaian dengan implementasi. khususnya *Bellman-Ford*.

Untuk itu pada permasalahan mencari rute atau lintasan terdekat pada suatu permasalahan *graf*, untuk menuju ke tujuan akhir dimulai dengan titik awal (start) melalui simpul yang berhubungan hingga titik akhir (finish) *graf* bisa menggunakan algoritma *Bellman-Ford*. Bentuk umum notasi matematika algoritma *Bellman-Ford* sebagai berikut :

$$M[i, v] = \min(M[i - 1, v], (M[i - 1, n] + C_{vn}))$$

Keterangan :

M : Simpul atau titik untuk uji per-iterasi

i : Proses uji iterasi

v : Point atau titik untuk perhitungan uji iterasi

n : Point atau titik yang terhubung untuk uji iterasi

C : Bobot atau nilai antara simpul yang dilakukan uji iterasi

Dari penghitungan notasi matematika diatas yang telah dipaparkan, maka terdapat langkah-langkah untuk menentukan rute terpendek menggunakan algoritma *Bellman-Ford* sebagai berikut :

1. Peta yang telah ada diubah terlebih dahulu menjadi *graf* yang memiliki arah dan mempunyai bobot.
2. Setelah peta diubah menjadi *graf* berarah, setelah itu titik yang ada diklarifikasikan, untuk mulai dari titik awal (start) menuju titik akhir tujuan (finish), semua titik dan sisi diuraikan.
3. Memberi tanda atau symbol sebagai penanda. Penanda pada titik awal sering ditandai dengan "0" sedangkan titik lainnya sering ditandai dengan tanda ∞ .
4. Uji iterasi dimulai dari titik start ditandai dengan "0" hingga titik akhir (finish), sebelum itu terdapat titik-titik yang terhubung dengan titik "0" (start) dengan nilai atau bobot. Pada uji iterasi titik awal akan ditambahkan dengan titik lain yang terhubung nantinya akan diambil hasil sebagai acuan adalah hasil yang terkecil.

Bentuk umum notasi matematika algoritma *Bellman-Ford* dibawah ini :

$$M[i, v] = \min(M[i - 1, v], (M[i - 1, n] + C_{vn}))$$

5. Untuk memperoleh semua nilai hasil dari titik yang ada maka iterasi akan dilakukan secara berulang hingga akhir.
6. Setelah semua iterasi dilakukan hingga akhir maka akan diperoleh hasil.
7. Jika perhitungan sudah sampai semua titik dan mencapai akhir tujuan maka iterasi dihentikan, maka solusi yang ada sudah dapat dikatakan optimal.

E. Pengujian dan Evaluasi

Pada tahap ini, Ketika aplikasi telah dikembangkan maka dilakukan pengujian. Pengujian dilakukan untuk membuktikan dan mengetahui bahwa aplikasi yang telah dibuat dan dikembangkan sesuai dengan keinginan dan kebutuhan dari peneliti. Pemrograman yang telah dipilih dengan menggunakan algoritma *Bellman-Ford* berbasis *website* dengan pemrograman PHP.

III. HASIL DAN PEMBAHASAN

A. Hasil Penelitian

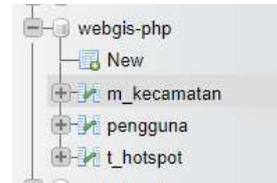
Website sebagai wadah dan bentuk implementasi dari perancangan sistem serta desain yang telah disimpulkan sebelumnya pada bab ke-2, penelitian ini dibuat dengan beberapa hal yang bersangkutan dengan hasil perancangan sistem informasi geografis di Kota Surabaya dengan menggunakan algoritma *Bellman-Ford*.

B. Pembahasan

dari penelitian dengan topik yang telah dijelaskan sebelumnya. Pembahasan akan dimulai dengan menu serta fitur yang tersedia pada *website* sistem informasi geografis, lalu database yang dibutuhkan dalam pembuatan dan diakhiri dengan contoh kasus mencari rute bengkel terdekat yang nantinya akan diimplementasikan menjadi *graf* dan dibagi menjadi beberapa tabel untuk uji per-

iterasi sampai dengan iterasi terakhir atau titik tujuan.

1. Pembuatan database



Gbr. 10 3 Tabel pada database

Pada Gbr. 10 dijelaskan pada dalam pembuatan *website* sistem informasi geografis terdapat 3 tabel utama yang diperlukan. Terdapat tabel *m_kecamatan*, *pengguna*, *t_hotspot*, tiap tabel mempunyai fungsi serta kegunaan yang berbeda-beda. Untuk *m_kecamatan* digunakan sebagai manipulasi data khusus untuk kecamatan, sedangkan untuk tabel *pengguna* digunakan untuk manipulasi data user atau admin, selanjutnya terdapat tabel *t_hotspot* sebagai manipulasi data dari titik bengkel.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra |
|---|-------------------|-------------|-------------------|------------|------|---------|----------|----------------|
| 1 | id_kecamatan | int(11) | | | No | None | | AUTO_INCREMENT |
| 2 | kd_kecamatan | varchar(10) | latin1_swedish_ci | | No | None | | |
| 3 | nm_kecamatan | varchar(30) | latin1_swedish_ci | | No | None | | |
| 4 | geopoin_kecamatan | varchar(30) | latin1_swedish_ci | | No | None | | |
| 5 | warna_kecamatan | varchar(10) | latin1_swedish_ci | | No | None | | |

Gbr. 11 Struktur tabel *m_kecamatan*

Pada Gbr. 11 diatas merupakan struktur tabel *m_kecamatan*. Di dalam tabel *m_kecamatan* terdapat 5 baris yang digunakan untuk memanipulasi data mulai dari tambah, edit, serta hapus data untuk kecamatan, admin yang mempunyai hak penuh untuk melakukan manipulasi data.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra |
|---|-------------|----------------------|-------------------|------------|------|---------|----------|----------------|
| 1 | id_pengguna | int(11) | | | No | None | | AUTO_INCREMENT |
| 2 | nm_pengguna | varchar(20) | latin1_swedish_ci | | No | None | | |
| 3 | kt_sandi | varchar(32) | latin1_swedish_ci | | No | None | | |
| 4 | level | enum('Admin','User') | latin1_swedish_ci | | No | User | | |

Gbr. 12 Struktur tabel *pengguna*

Pada Gbr. 12 diatas merupakan struktur tabel pengguna. Di dalam tabel pengguna terdapat 4 baris yang digunakan untuk memanipulasi data mulai dari tambah, edit, serta hapus data untuk kebutuhan pengguna, admin yang mempunyai hak penuh untuk melakukan manipulasi data.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra |
|---|--------------|--------------|-------------------|------------|------|---------|----------|----------------|
| 1 | id_hotspot | int(11) | | | No | None | | AUTO_INCREMENT |
| 2 | id_kecamatan | int(11) | | | No | None | | |
| 3 | lokasi | varchar(50) | latin1_swedish_ci | | No | None | | |
| 4 | keterangan | varchar(100) | latin1_swedish_ci | | No | None | | |
| 5 | lat | float(9,6) | | | No | None | | |
| 6 | lng | float(9,6) | | | No | None | | |
| 7 | tanggal | date | | | No | None | | |
| 8 | marker | varchar(50) | latin1_swedish_ci | | No | None | | |

Gbr. 13 Struktur tabel t_hotspot

Pada Gbr. 13 diatas merupakan struktur tabel t_hotspot. Di dalam tabel pengguna terdapat 8 baris yang digunakan untuk memanipulasi data mulai dari tambah, edit, serta hapus data untuk kebutuhan lokasi titik bengkel motor, admin yang mempunyai hak penuh untuk melakukan manipulasi data.

2. Tampilan Fitur pada website SIG

Gbr. 14 Tampilan form login

Pada Gbr. 14 diatas merupakan tampilan dari form login. Admin dan user dalam menggunakan aplikasi ini harus melakukan login pada website terlebih dahulu. email dan password dibutuhkan pada saat login, jika saat email dan password

salah maka sistem akan memunculkan peringatan notifikasi salah. Jika data email dan password benar maka akan dialihkan pada halaman beranda.



Gbr. 15 Tampilan fitur pada user dan admin

Pada Gbr. 15 dijelaskan bahwa terdapat perbedaan hak akses yang didapat sebagai user dan admin, pada hal ini terdapat 2 perbedaan fitur yang ada. Admin memiliki hak akses penuh dalam manipulasi data kecamatan serta manipulasi bengkel motor, sedangkan untuk user hanya diberikan akses untuk leaflet map dengan berisikan 6 tampilan dan fungsi map yang berbeda-beda.

Gbr. 16 Tampilan form manipulasi kecamatan

Pada Gbr. 16 dijelaskan admin memiliki hak akses penuh dalam manipulasi data kecamatan, seperti penambahan data baru kecamatan, edit data kecamatan yang sudah ada, hapus data kecamatan yang tidak diperlukan.

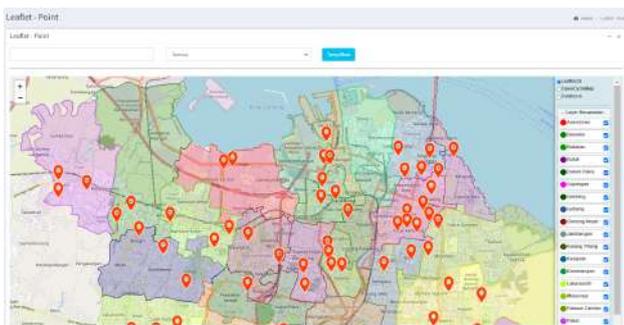
Gbr. 17 Tampilan form manipulasi bengkel

Pada Gbr. 17 dijelaskan admin memiliki hak akses penuh dalam manipulasi data bengkel, seperti penambahan data baru bengkel, edit data bengkel yang sudah ada, hapus data bengkel yang tidak diperlukan seperti telah bangkrut ataupun berpindah lokasi.



Gbr. 18 Tampilan leaflet map standar

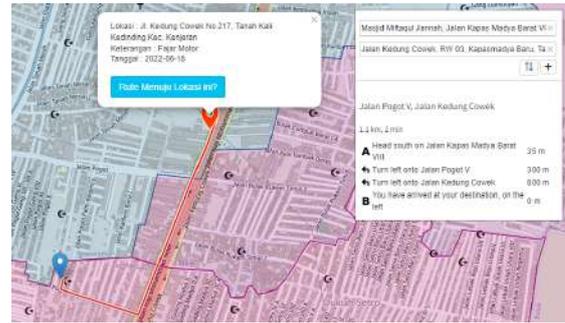
Pada Gbr. 18 diatas tampilan leaflet map dasar yang digunakan dan sebagai inti utama dari penelitian ini, sistem informasi geografis yang ditampilkan berupa map Kota Surabaya, dengan perbedaan warna sebagai symbol untuk perbatasan antar kecamatan, pada bagian kiri terdapat legenda dengan 31 daftar kecamatan yang ada beserta warnanya sebagai penanda.



Gbr. 19 Tampilan leaflet map point marker

Pada Gbr.19 diatas tampilan leaflet map point, digambarkan dengan map Kota Surabaya dengan batas antar kecamatan kemudian ditambahkan lokasi titik bengkel pada Kota Surabaya dengan menandai atau memberi tanda *marker*. Pada Gbr.19 diatas data

yang digunakan merupakan sebagian bengkel motor yang didaftarkan oleh admin.



Gbr. 20 Tampilan mencari rute terdekat bengkel

Gbr. 20 merupakan pencarian rute terdekat bengkel motor. Titik awal pada Gbr. 20 terletak pada bawah kiri sebagai marker, lalu titik tujuan terdapat pada marker yang memunculkan pop-up. Dalam penggunaan fitur ini pengguna dapat mencari lokasi terdekat dengan melakukan input atau menggeser lokasi awal ke titik awal yang diinginkan, setelah itu klik pada bagian marker bengkel, kemudian terdapat pop up. tekan pada menu button yang ada, akan muncul rute menuju tempat bengkel tersebut dengan garis track. Track jalan atau persimpangan yang dilalui dapat dilihat pada pojok kanan atas, terdapat berapa jarak yang dilalui serta kira-kira waktu yang dibutuhkan untuk menuju tempat tersebut.

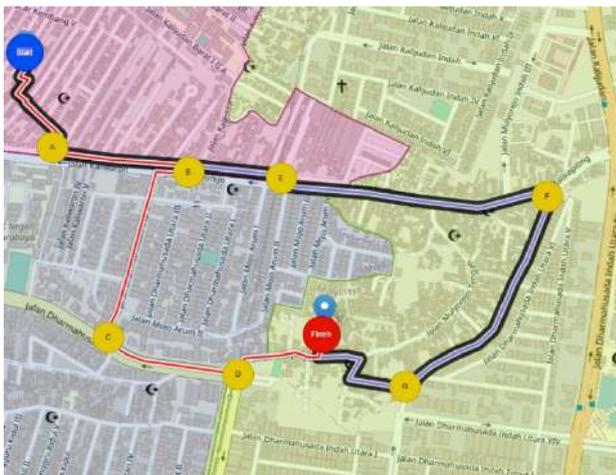
3. Implementasi algoritma dengan contoh soal

Tahap terakhir dalam penelitian ini dengan menguji sekaligus mengevaluasi kinerja algoritma *Bellman-Ford* dengan beberapa langkah dan persiapan yang telah disebutkan sebelumnya. Teknik yang nantinya dipakai adalah dengan menentukan jalur yang ingin diuji lalu diimplementasikan dalam *graf* berarah ditambahkan dengan bobot keseluruhan.



Gbr.21 Contoh Kasus uji kinerja algoritma

Pada contoh kasus lintasan terpendek, data ini peneliti gunakan data manual seseorang yang berada di Jalan Kali Kepiting Java VII, Surabaya yang ingin melakukan *service* motornya dikarenakan waktunya perawatan. Untuk menuju ke bengkel Dzetifa Jaya Motor. Untuk menuju bengkel tersebut terdapat beberapa titik-titik persimpangan jalan yang akan dilewati. Pada Gbr. 22 dibawah ini dijelaskan *graf* perjalanan pengguna menuju ke Dzetifa Jaya Motor beserta bobotnya:



Keterangan :

- Jalan Kali Kepiting Jaya VII (Tujuan awal)
- Persimpangan Jalan
- Dzetifa Jaya Motor (Tujuan akhir)

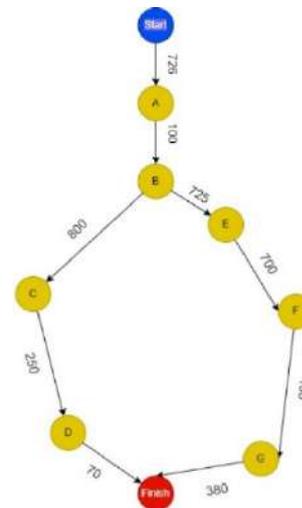
Gbr. 22 Graf rute dan keterangan menuju ke tujuan akhir

Pada Gbr. 22 dijelaskan *graf* tersebut ada 9 simpul atau titik, dimana titik berwarna biru adalah titik awal, titik hijau adalah persimpangan jalan, dan titik merah adalah tujuan akhir (Dzetifa Jaya Motor).

Graf perjalanan pada Gbr. 22 digunakan untuk mengetahui lintasan terpendek atau rute terdekat dimulai dari Jalan Kali Kepiting Java VII menuju Dzetifa Jaya Motor. Satuan yang digunakan adalah meter (m).

Maka untuk menyelesaikan permasalahan tersebut terdapat langkah-langkah yang digunakan menggunakan algoritma *Bellman-Ford* sebagai berikut :

Langkah 1: Peta yang telah ada diubah terlebih dahulu menjadi *graf* yang memiliki arah dan mempunyai bobot pada Gbr. 23



Gbr. 23 Diubah menjadi Graf berarah dan berbobot

Langkah 2: Setelah peta diubah menjadi *graf* berarah, setelah itu titik yang ada diklarifikasikan, untuk mulai dari titik awal (start) menuju titik tujuan (finish), semua titik dan sisi diuraikan. Terlihat pada Gbr. 23

Langkah 3: Memberi tanda atau symbol sebagai penanda. Penanda pada titik awal sering ditandai dengan “0” sedangkan titik lainnya sering ditandai dengan tanda ∞ . Langkah ketiga dijelaskan pada Gbr.24.

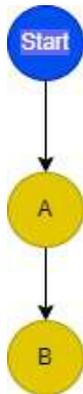
$$\begin{aligned}
 M[2, B] &= \min(M[1, B] + (M[1, A] + 726)) \\
 &= \min(\infty, [100 + 726]) \\
 &= \min(\infty, [826]) \\
 &= 826
 \end{aligned}$$

Untuk mengetahui titik yang telah memiliki nilai setelah dilakukan iterasi ke-2, dapat dilihat pada tabel V dibawah ini :

TABEL III
 TABEL ITERASI KEDUA

| ITERASI | START | A | B | C | D | E | F | G | FINISH |
|---------|-------|-----|-----|---|---|---|---|---|--------|
| 0 | 0 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| 1 | 0 | 726 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| 2 | 0 | 726 | 826 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |

Setelah dilakukan iterasi ke-2, maka titik B memiliki bobot atau nilai sebesar 826, sedangkan titik lainnya masih bernilai ∞. karena dalam perhitungan jika setelah dihitung akan tetap bernilai ∞ karena yang terhubung langsung dengan titik A hanya titik B. Pada Gbr. 26 dibawah ini adalah proses iterasi ke-2.



Gbr.26 Proses iterasi ke-2

Untuk memenuhi nilai pada titik-titik lainnya yang masih belum diketahui, maka dilakukan proses iterasi ke-3 terhadap semua titik yang terhubung mempunyai rute yang terhubung dengan titik B.

c. Uji iterasi ke-3

Perhitungan iterasi ke-3 adalah semua titik yang terhubung langsung dengan titik B yaitu ada 2 titik, C dan E. perhitungan iterasi ke-3 titik B yang

mempunyai nilai atau berbobot 826 ditambah dengan titik C yang bernilai 800, sedangkan pada sisi lainnya titik B juga terhubung dengan titik E, maka dari itu ditambah dengan titik E yang mempunyai nilai atau berbobot 725. Dibawah ini adalah proses perhitungan iterasi ke-3 :

$$\begin{aligned}
 M[3, C] &= \min(M[1, C] + (M[1, B] + 826)) \\
 &= \min(\infty, [800 + 826]) \\
 &= \min(\infty, [1626]) \\
 &= 1626
 \end{aligned}$$

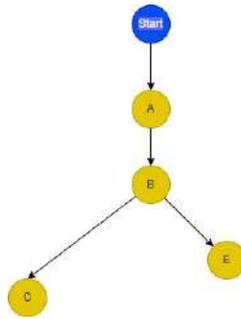
$$\begin{aligned}
 M[3, E] &= \min(M[1, E] + (M[1, B] + 826)) \\
 &= \min(\infty, [725 + 826]) \\
 &= \min(\infty, [1551]) \\
 &= 1551
 \end{aligned}$$

Untuk mengetahui titik yang telah memiliki nilai setelah dilakukan iterasi ketiga, maka dapat dilihat pada tabel VI dibawah ini:

TABEL VI
 TABEL ITERASI KETIGA

| ITERAS I | STAR T | A | B | C | D | E | F | G | FINIS H |
|----------|--------|-----|-----|------|---|------|---|---|---------|
| 0 | 0 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| 1 | 0 | 726 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| 2 | 0 | 726 | 826 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| 3 | 0 | 726 | 826 | 1626 | ∞ | 1551 | ∞ | ∞ | ∞ |

Setelah dilakukan iterasi ketiga, titik C dan E memiliki bobot atau nilai sebesar 1626 untuk C sedangkan E bernilai 1551, sedangkan titik lainnya masih bernilai ∞, karena dalam perhitungan jika setelah dihitung akan tetap bernilai ∞ karena yang terhubung langsung dengan titik B hanya titik C dan E. Pada Gbr. 27 dibawah ini adalah proses iterasi ke-3.:



Gbr.27 Proses iterasi ke-3

Untuk memenuhi nilai pada titik-titik lainnya yang masih belum diketahui, maka dilakukan proses iterasi ke-4 terhadap semua titik yang terhubung mempunyai rute yang terhubung dengan titik C dan E.

d. Uji iterasi ke-4

Perhitungan iterasi keempat adalah semua titik yang terhubung langsung dengan titik C dan E yaitu titik D berhubungan dengan C, sedangkan titik E berhubungan dengan F. perhitungan iterasi ke-4 titik C yang mempunyai nilai atau berbobot 1626 ditambah dengan titik D dengan nilai atau bobot 250, sedangkan titik lainnya yaitu E yang terhubung dengan titik F. perhitungan iterasi ke-4 titik E yang mempunyai nilai atau berbobot 1551 terhubung dengan titik F dengan nilai atau bobot 700. Dibawah ini adalah proses perhitungan iterasi ke-4 :

$$\begin{aligned}
 M[4, D] &= \min(M[1, D] + (M[1, C] + 1626)) \\
 &= \min(\infty, [250 + 1626]) \\
 &= \min(\infty, [1626]) \\
 &= 1876
 \end{aligned}$$

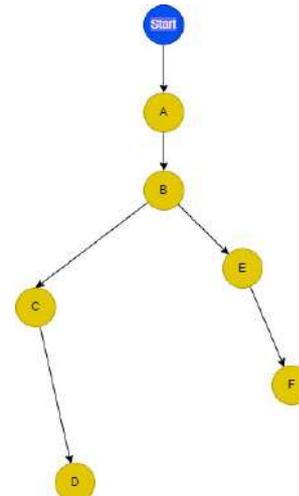
$$\begin{aligned}
 M[4, F] &= \min(M[1, F] + (M[1, E] + 1551)) \\
 &= \min(\infty, [700 + 1551]) \\
 &= \min(\infty, [1626]) \\
 &= 2251
 \end{aligned}$$

Untuk mengetahui titik yang telah memiliki nilai setelah dilakukan iterasi ke-4, maka dapat dilihat pada tabel VII dibawah ini:

TABEL VII
 TABEL ITERASI KEEMPAT

| ITERASI | START | A | B | C | D | E | F | G | FINISH |
|---------|-------|-----|-----|------|------|------|------|---|--------|
| 0 | 0 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| 1 | 0 | 726 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| 2 | 0 | 726 | 826 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| 3 | 0 | 726 | 826 | 1626 | ∞ | 1551 | ∞ | ∞ | ∞ |
| 4 | 0 | 726 | 826 | 1626 | 1876 | 1551 | 2251 | ∞ | ∞ |

Setelah dilakukan iterasi ke-4 titik D dan F memiliki bobot atau nilai sebesar 1876 untuk D sedangkan F bernilai 2251, sedangkan titik lainnya masih bernilai ∞, karena dalam perhitungan jika setelah dihitung akan tetap bernilai ∞ karena yang terhubung langsung dengan titik D hanya titik F dan E. Pada Gbr. 27 dibawah ini adalah proses iterasi ke-4.:



Gbr.28 Proses iterasi ke-4

e. Uji iterasi ke-5

Perhitungan iterasi ke-5 adalah semua titik yang terhubung langsung dengan titik D dan F yaitu titik D berhubungan dengan tujuan akhir (Finish), sedangkan titik F berhubungan dengan G. perhitungan iterasi kelima titik D yang mempunyai nilai atau berbobot 1876 ditambah dengan titik Finish dengan nilai atau bobot 70, sedangkan titik lainnya yaitu F yang terhubung dengan titik G. perhitungan iterasi kelima titik F yang mempunyai nilai atau berbobot 2251 ditambah dengan titik G

dengan nilai atau bobot 150. Dibawah ini adalah proses perhitungan iterasi ke-5 :

$$\begin{aligned} M[5, Finish] &= \min(M[1, Finish] + (M[1, D] + 1876)) \\ &= \min(\infty, [70 + 1876]) \\ &= \min(\infty, [1946]) \\ &= 1946 \end{aligned}$$

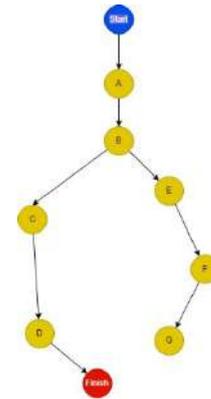
$$\begin{aligned} M[5, G] &= \min(M[1, G] + (M[1, F] + 2251)) \\ &= \min(\infty, [150 + 2251]) \\ &= \min(\infty, [2401]) \\ &= 2401 \end{aligned}$$

Untuk mengetahui titik yang telah memiliki nilai setelah dilakukan iterasi ke-5, maka dapat dilihat pada tabel VIII dibawah ini:

TABEL VIII
 TABEL ITERASI KELIMA

| ITERASI | START | A | B | C | D | E | F | G | FINISH |
|---------|-------|-----|-----|------|------|------|------|------|--------|
| 0 | 0 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| 1 | 0 | 726 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| 2 | 0 | 726 | 826 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| 3 | 0 | 726 | 826 | 1626 | ∞ | 1551 | ∞ | ∞ | ∞ |
| 4 | 0 | 726 | 826 | 1626 | 1876 | 1551 | 2251 | ∞ | ∞ |
| 5 | 0 | 726 | 826 | 1626 | 1876 | 1551 | 2251 | 2401 | 1946 |

Setelah dilakukan iterasi ke-5, titik Finish dan G telah memiliki bobot atau nilai sebesar 1946 untuk Finish dengan melalui jalur C, sedangkan G bernilai 2401, sedangkan titik lainnya yang masih tersisa bernilai ∞ merupakan titik finish dengan melalui jalur satunya jalur E, karena dalam perhitungan jika setelah dihitung akan tetap bernilai ∞ karena yang terhubung langsung dengan titik D hanya titik Finish (tujuan akhir) dan G . Pada Gbr. 29 dibawah ini adalah proses iterasi ke-5.:



Gbr. 29 Proses iterasi ke-5

Selanjutnya akan dilakukan proses iterasi ke-6 terhadap titik Finish yang terhubung langsung dengan titik G.

f. Uji iterasi ke-6

Perhitungan iterasi ke-6 adalah semua titik yang terhubung langsung dengan titik Finish yaitu tersisa titik G berhubungan dengan tujuan akhir (Finish). perhitungan iterasi ke-6 titik G yang mempunyai nilai atau berbobot 2401 ditambah dengan titik Finish dengan nilai atau bobot 380, sedangkan titik lainnya telah sampai pada tujuan akhir, jadi hanya titik G saja yang dihitung untuk iterasi ke-6. Dibawah ini adalah proses perhitungan iterasi ke-6 :

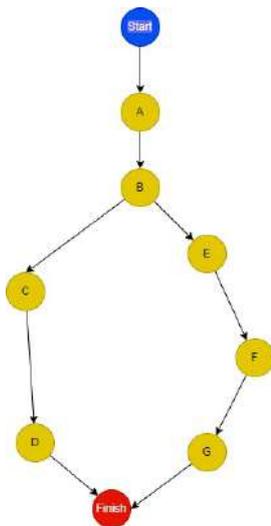
$$\begin{aligned} M[6, Finish] &= \min(M[1, Finish] + (M[1, G] + 2251)) \\ &= \min(\infty, [380 + 2401]) \\ &= \min(\infty, [2781]) \\ &= 2781 \end{aligned}$$

Setelah melakukan 6 iterasi secara runtut, titik yang tadinya belum diketahui nilainya sudah terdapat nilainya atau dapat dikatakan sudah terhitung dan memperoleh bobot / nilai untuk lintasan terpendek menuju Dzetifa Jaya Motor, setelah semua telah ditemukan maka proses iterasi bisa dihentikan pada iterasi keenam karena sudah mendapat hasil akhir sampai ke titik Finish. Pada tabel IX dibawah ini merupakan hasil dari semua proses iterasi beserta nilai yang diperoleh :

TABEL IX
 TABEL ITERASI KEENAM

| ITERASI | START | A | B | C | D | E | F | G | FINISH |
|---------|-------|---------|---------|----------|----------|----------|----------|----------|--------|
| 0 | 0 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| 1 | 0 | 72 6 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| 2 | 0 | 72 6 | 82 6 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| 3 | 0 | 72 6 | 82 6 | 162 6 | ∞ | 155 1 | ∞ | ∞ | ∞ |
| 4 | 0 | 72 6 | 82 6 | 162 6 | 187 6 | 155 1 | 225 1 | ∞ | ∞ |
| 5 | 0 | 72 6 | 82 6 | 162 6 | 187 6 | 155 1 | 225 1 | 240 1 | 1946 |
| 6 | 0 | 72 6 | 82 6 | 162 6 | 187 6 | 155 1 | 225 1 | 240 1 | 2781 |

Setelah dilakukan iterasi ke-6, Titik finish jalur yang dilalui E telah memiliki bobot atau nilai sebesar 2781. Sedangkan pada iterasi sebelumnya pada iterasi ke-5 bahwa nilai saat menuju titik finish melalui C diperoleh dengan bobot atau nilai 1946. Pada Gbr. 29 dibawah ini adalah proses iterasi ke-6:



Gbr. 30 Proses iterasi ke-6

Setelah 6 iterasi dilakukan didapatkan 2 nilai minimum untuk menuju titik Finish, terdapat pada iterasi ke-5 dan iterasi ke-6. Untuk nilai lintasan minimum ke-5 yaitu 1946 sedangkan untuk iterasi ke-6 2781. Maka dipilih nilai atau bobot minimum terkecil untuk memudahkan pengguna sampai ke tempat tujuan (finish) yaitu 1946. Maka lintasan yang dapat

dilewati oleh pengguna yaitu Start menuju ke A, B, C, D, dan finish. Dengan itu didapatkan rute terdekat dari titik awal start sampai dengan ke titik akhir tujuan (finish) berdasarkan proses hasil iterasi sebagai berikut :

1. Start menuju titik A

Perjalanan dimulai dari titik awal (start) dengan titik A, pada titik start hanya ada titik A yang terhubung langsung dengan titik A sehingga titik A merupakan rute minimum yang didapatkan berdasarkan algoritma *Bellman-Ford*. Maka iterasi pertama yang di dapatkan yaitu :



2. A menuju titik B

Dari titik A hanya terdapat satu titik B yang terhubung langsung. Maka untuk mengambil rute minimum didapatkan dari titik A ke titik B yaitu :



3. B menuju titik C

Dari titik B didapatkan 2 jalur yang bisa digunakan untuk memperoleh rute minimum dengan melewati C dan E. lalu diseleksi nilai atau bobot paling kecil maka didapatkan C. Dengan itu rute minimum dari titik B ke C yaitu



4. C menuju titik D

Dari titik C terhubung dengan titik D. lalu di jalur lain terdapat titik F yang terhubung dengan titik E. untuk rute minimum dari titik C menuju D yaitu



5. D ke titik Finish

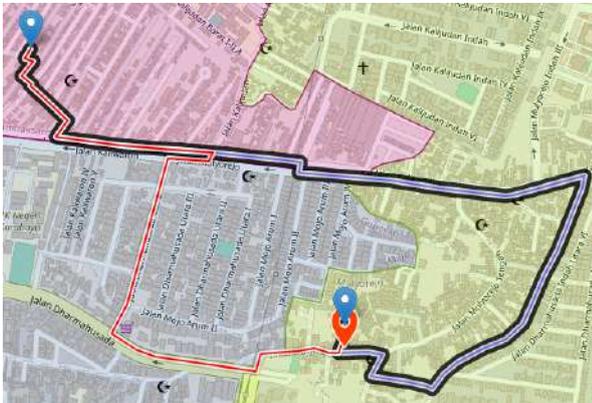
Dari titik D menuju titik Finish ada beberapa titik yang akan dilewati yaitu titik A, B, C, dan D. sedangkan pada jalur lain terdapat titik A, B, E, F, G, lalu ke titik Finish. Dengan rute minimum didapatkan titik D ke titik Finish yaitu



Dengan proses uji iterasi yang telah diselesaikan , pada iterasi kelima, 1 lintasan diberikan 5 titik persimpangan jalan yang akan dilewati pengguna ke

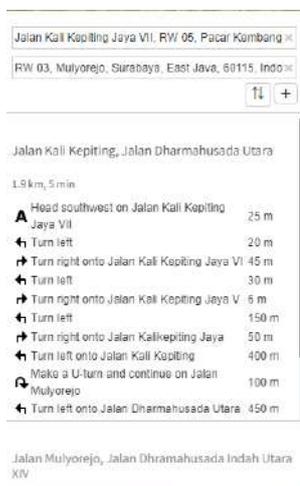
Dzetifa Jaya Motor di Kota Surabaya dengan jarak yang harus ditempuh 1946 m atau 1,946 km. Hasil yang didapat merupakan implementasi lintasan atau rute minimum menggunakan algoritma *Bellman-Ford*.

Terdapat 2 jalur untuk menuju ke Dzetifa Jaya Motor jalur yang tercepat dan memiliki jarak minimum akan ditampilkan terlebih dahulu, pada contoh kasus ini melalui Jalan Dharmahasada Utara, lalu untuk jalur alternatif kedua bisa melewati Mulyorejo dan Jalan Dharmahasada Indah Utara XIV.



Gbr. 31 Rute yang dilewati ke Dzetifa Jaya Motor

Pada Gbr. 31 terdapat 2 jalur yang dapat dilalui untuk menuju ke Dzetifa Jaya Motor jalur yang memiliki tanda tebal merupakan jalur alternatif sebagai pengganti jalur utama, namun memiliki kekurangan jarak yang dilalui lebih jauh dan waktu yang ditempuh lebih lama.



Gbr. 32 Rute persimpangan jalan yang dilewati menuju Dzetifa Jaya Motor

Pada Gbr. 32 terdapat rute persimpangan yang dilalui untuk menuju ke Dzetifa Jaya Motor.

IV. KESIMPULAN

Maka dari itu, kesimpulan yang didapatkan setelah penelitian ini telah selesai dan berhasil diterapkan pada *website* sistem informasi geografis di Kota Surabaya dengan algoritma *Bellman-Ford*. Dengan bahan awal pencarian data hingga melakukan uji dengan kinerja algoritma apakah sesuai dan cocok untuk diterapkan pada aplikasi *website* ini, melakukan perbandingan hasil data dengan manual menggunakan notasi matematika dengan uji per-iterasi lalu diimplementasikan ke dalam program komputer yang ada dengan baik dan berhasil.

V. SARAN

Saran yang bisa penulis sampaikan kepada peneliti yang ingin meneruskan atau mengembangkan penelitian yang telah dilakukan antara lain. Telah dijelaskan bahwa sistem ini pada intinya hanya pergerakan awal titik tujuan bukan secara realtime namun memasukkan data lokasi awal secara manual, kedepannya diharapkan data titik awal didapatkan melalui realtime lokasi. Kemudian sebaiknya perlu pembandingan dengan algoritma lain untuk mengetahui algoritma terbaik pada kasus seperti ini. Lalu terakhir saran dari penulis adalah media aplikasi sangat boleh sekali diterapkan dengan teknologi yang modern di penelitian mendatang.

UCAPAN TERIMA KASIH

Terima kasih banyak untuk pertama saya panjatkan terlebih dahulu kepada Allah SWT pencipta semesta serta seisinya. penelitian ini dapat diselesaikan dan tercipta sesuai yang telah direncanakan dengan baik. Tidak lupa sholawat dan salam kepada Nabi Muhammad SAW. Kemudian saya ucapkan banyak terima kasih kepada kedua orang tua saya, baik Ibu dan Ayah yang telah mendukung dan mendoakan saya serta dibantu dengan bimbingan moral dan matril. Tidak lupa terimakasih akan saya berikan kepada Dodik Arwin Darmawan, ST., SST., MT. sebagai dosen pembimbing, berkat arahan dan bimbingan dari

beliau penelitian ini dapat terlaksana dengan baik hingga selesai. Berikutnya akan saya berikan kepada teman-teman saya dan juga pihak yang telah menyukseskan penelitian ini baik itu berupa ilmu ataupun dukungan kepada saya yang telah menemani saya hingga selesai penelitian ini.

REFERENSI

- [1] J. Demitra Bawole and P. Hanna Chernovita, "Algoritma Bellman-Ford untuk Menentukan Jalur Terpendek dalam Survey Klaim Asuransi (Studi Kasus: PT.Asuransi Sinar Mas,Jakarta)," *INOBIS: Jurnal Inovasi Bisnis dan Manajemen Indonesia*, vol. Volume 03 No.1, 2019.
- [2] P. Rully and S. Nadya, Algoritma Bellman-Ford Untuk Menentukan Jalur Tercepat Dalam Sistem Informasi Geografis, *Jurnal Penelitian Ilmu Komputer, System Embedded & Logic*, p-ISSN:2303-3304, e-ISSN: 2620-3533, 2018.
- [3] O. Jason Agung, E. Thomas and A. Halim, Analisis Perbandingan Algoritma Floyd-Warshall Dengan Algoritma Bellman-Ford Dalam Pencarian Rute Terpendek Menuju Museum di Jakarta, *Kalbisscentia*, vol. Volume 5, No.1, 2018.
- [4] I. Bayu Ahmad Prasetyo and M. Andi "ANALISIS PERBANDINGAN PADA ALGORITMA BELLMAN FORD DAN DIJKSTRA PADA GOOGLE MAP," *Open Journal System*, e-ISSN : 2621-9441, p-ISSN : 2623-1041 Volume 3, No.2, 2020.
- [5] K. Tomas, T. Minija and W. Florentin, Combining optimal path search with task-dependent learning in a neural network, arXiv:2201.11104v2 [cs.LG], 2022.
- [6] A. She Yesti, IMPLEMENTASI ALGORITMA *BELLMAN-FORD* DALAM MENENTUKAN LINTASAN TERPENDEK TRUK PEMBUANGAN SAMPAH, Riau Pekanbaru:Fakultas Sains dan Teknologi, UIN Sultan Syarif Kasim, 2022.
- [7] T. Estri H. Hutasoit, Pencarian Rute Terpendek Menggunakan Algoritma Bellman-Ford (Studi Kasus: PT.JNE Medan), *Jurnal Sistem Komputer dan Informatika (JSON)*, ISSN : 2685-998X Volume 1, No,1, 2019.
- [8] S. Akbar, Z. Muhammad and H. Dedy, "Perbandingan Algoritma Dijkstra dan Bellman-Ford Dalam Pencarian Jarak Terpendek Pada SPBU," *Seminar Nasional Sains & Teknologi Informasi (SENSASI)*, ISBN : 978-602-52720-2-8, 2019.
- [9] N. Yesryl FrenDOS and Ahyuni, "PERKEMBANGAN BENGKEL SEPEDA MOTOR DI KECAMATAN PADANG UTARA KOTA PADANG TAHUN 2000-2019," *JURNAL BUANA : JURUSAN GEOGRAFI FAKULTAS ILMU SOSIAL-UNP*, E-ISSN : 2615-2630 vol. Vol-4 No.6, 2020.
- [10] A. IGN Widhiananda, IM Arsa Suyadnya and O. Komang Saputra, RANCANG BANGUN APLIKASI RESERVASI SERVICE UNTUK BENGKEL SEPEDA MOTOR BERBASIS WEB, *E-Journal SPEKTRUM*, vol. 4 No.2, 2017.
- [11] S. Saniati, Z. Mhd. Siambaton and H. Tasliyah, Aplikasi Pemetaan Objek Wisata dan Pencarian Jalur Terpendek Berbasis Web-Gis Menggunakan Algoritma Dijkstra di Kota Subulussalam, *Jurnal Ilmu Komputer*,ISSN 2829-8616, 2022.
- [12] D. Surya, PENERAPAN ALGORITMA DIJKSTRA APLIKASI GIS UNTUK PANDUAN PENCARIAN WISATA TERDEKAR, *Cyberarea.id*, Vol. 2 (8), 2022.
- [13] R. Ilham Furqon and D. Theresia Wismarini, Model Sistem Informasi Geografis Berbasis

- Web Rute Terdekat Menuju Restoran di Kota Semarang Dengan Algoritma A*, *Jurnal Ilmiah Teknik Informatika dan Sistem Informasi*, vol. Vol. 11 No. 1, e-ISSN : 2685-0893, p-ISSN : 2089-3787, 2022.
- [14] H. Sabrina, A. Desy and P. Panji, Perancangan Sistem Informasi Geografis Perlengkapan Jalan Berbasis Daerah Rawan Kecelakaan, *Jurnal Keselamatan Transportasi Jalan (Indonesian Journal of Road Safety)*, vol. Vol.9 No.1, ISSN : 2338-4247, ISSN : 2721-7248, 2022.
- [15] Masnur, A. Syahirun and I. Muhammad, APLIKASI SISTEM INFORMASI GEOGRAFIS (SIG) PEMETAAN LAHAN PERTANIAN DAN KOMODITAS HASIL PANEN DI KABUPATEN SIDRAP BERBASIS WEB, *JURNAL SINTAKS LOGIKA*, Vol.2 No,1, E-ISSN : 2775-412X, 2022.
- [16] A. Risky Ramadhani, "IMPLEMENTASI GRAPH COLORING DALAM PEMETAAN KECAMATAN DI KABUPATEN KEDIRI," *Jurnal SIMETRIS*, Vol.7 No.2, ISSN : 2252-4983, 2016.
- [17] S. Anisa, "XAMPP adalah: Pengertian, fungsi, 5 komponen, dan cara menggunakannya," 15 September 2021. [Online]. Available: <https://www.ekrut.com/media/xampp-adalah>. [Accessed 27 Juni 2022].
- [18] C. Ariata, "Apa itu Bootstrap dan Fungsinya? Panduan Bagi Pemula," 13 Agustus 2021. [Online]. Available: <https://www.hostinger.co.id/tutorial/apa-itu-bootstrap>. [Accessed 28 Juni 2022].
- [19] R. Muhammad Adani, "Apa itu MySQL: Pengertian, Fungsi, beserta kelebihan," 15 Agustus 202. [Online]. Available: <https://www.sekawanmedia.co.id/blog/pengertian-mysql/>. [Accessed 2 Juli 2022].
- [20] A. Faradilla, "Apa Itu PHP? Pengertian PHP untuk Pemula," 24 Januari 2022. [Online]. Available: <https://www.hostinger.co.id/tutorial/apa-itu-php/>. [Accessed 28 Juni 2022].
- [21] N. Rafael Ramadhon. "Pengertian Algoritma," 9 November 2021. [Online]. Available : <https://www.unida.ac.id/teknologi/artikel/pengertian-algoritma.html>. [Accessed 2 Juli 2022].
- [22] Bidang Informasi dan Komunikasi Publik serta Statistik Dinas Komunikasi dan Informatika Kota Surabaya. "Pembagian Kecamatan Per Wilayah," 2022. [Online]. Available : <https://www.surabaya.go.id/id/page/0/8166/kecamatan>. [Accessed 2 Juli 2022].