

**PENGEMBANGAN APLIKASI *MOBILE* PENCARIAN RUTE
TERPENDEK MENUJU LOKASI KULINER KOTA MALANG
MENGUNAKAN ALGORITMA *DIJKSTRA***

SKRIPSI

Digunakan Sebagai Syarat Maju Ujian Diploma IV
Politeknik Negeri Malang

Oleh:

MICHAEL EDUARDO KARAMBUT NIM.1341180137



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG**

2017

**PENGEMBANGAN APLIKASI *MOBILE* PENCARIAN RUTE
TERPENDEK MENUJU LOKASI KULINER KOTA MALANG
MENGUNAKAN ALGORITMA *DIJKSTRA***

SKRIPSI

Digunakan Sebagai Syarat Maju Ujian Diploma IV
Politeknik Negeri Malang

Oleh:

MICHAEL EDUARDO KARAMBUT NIM.1341180137



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG**

2017

HALAMAN PENGESAHAN

PENGEMBANGAN APLIKASI *MOBILE* PENCARIAN RUTE TERPENDEK MENUJU LOKASI KULINER KOTA MALANG MENGUNAKAN ALGORITMA *DIJKSTRA*

Disusun oleh:

MICHAEL EDUARDO KARAMBUT

NIM.1341180137

Skripsi ini telah diuji pada tanggal 12 September 2017

Disetujui oleh:

- | | | | |
|------------------|---|--------------------------------------|-------|
| 1. Penguji I | : | <u>Putra Prima Arhandi,ST.,M.KOM</u> | |
| | | NIP. 198611032014041001 | |
| 2. Penguji II | : | <u>Arief Prasetyo,S.KOM.M.KOM</u> | |
| | | NIP. 197903132008121002 | |
| 3. Pembimbing I | : | <u>Ely Setyo Astuti, ST., MT.</u> | |
| | | NIP. 197605152009122001 | |
| 4. Pembimbing II | : | <u>Hendra Pradibta, SE., MSC</u> | |
| | | NIP. 198305212006041003 | |

Mengetahui,

Ketua Jurusan
Teknologi Informasi

Ketua Program Studi
Teknik Informatika

Rudy Ariyanto, S.T., M.Cs.
NIP. 19711110 199903 1 002

Ir. Deddy Kusbianto P., M.MKom.
NIP. 19621128 198811 1 001

PERNYATAAN

Dengan ini saya menyatakan bahwa Laporan Skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Malang, 11 Juni 2017

Michael Eduardo Karambut

ABSTRAK

Karambut, M Eduardo. “Pengembangan Aplikasi *Mobile* Pencarian rute terpendek menuju lokasi Kuliner kota Malang menggunakan Algoritma *Dijkstra*”.
Pembimbing: (1) Ely Setyo Astuti, ST., MT., (2) Hendra Pradibta, SE., MSC.

Skripsi, Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang, 2017.

Hari ini kuliner menjadi bagian gaya hidup bagi masyarakat kota. Kuliner merupakan perpaduan aktivitas menikmati suatu makanan dalam suasana yang menyenangkan, seperti saat bersantai menikmati liburan, mengunjungi tempat-tempat yang menyediakan makanan khas, salah satunya kota Malang.

Kota Malang merupakan salah satu destinasi kuliner yang banyak dikunjungi, oleh karena itu para pengusaha rumah makan, restoran, kafe berlomba-lomba untuk menyajikan kuliner yang menarik untuk mahasiswa, wisatawan yang datang dan untuk warga Malang sendiri. Dalam hal ini tentunya Masyarakat membutuhkan sebuah aplikasi mobile yang dapat memberikan layanan pencarian lokasi kuliner di kota Malang. Pembuatan aplikasi bertujuan untuk membantu pencarian lokasi kuliner di kota Malang.

Penelitian ini mengembangkan sebuah aplikasi yang bisa mengoptimalkan pencarian jalur terpendek menuju lokasi kuliner di kota Malang menggunakan algoritma *Dijkstra Min (DestValue, MarkedValue+EdgeWeight)*. Pencarian tersebut digunakan untuk mencari bobot atau nilai terkecil pada setiap simpul yang akan dilewati menuju lokasi kuliner.

Kata Kunci : Pencarian, Kuliner, *Dijkstra*, Aplikasi *Mobile*, Rute terpendek.

ABSTRACT

Karambut, M Eduardo. *“Mobile Application Development to find Shortest route to Culinary spot in Malang using Dijkstra Algorithm”* **Advisors : (1) Ely Setyo Astuti, ST., MT., (2) Hendra Pradibta, SE., MSC.**

Thesis, Informatics Engineering Study Program, Department of Information Technology, State Polytechnic of Malang, 2017.

Culinary becomes a part of lifestyle for people who life in a city. Culinary is an activity combines of art of enjoying food is leisure time such on vacation, by visiting a certain place which is serve a traditional cuisine such as Malang.

Malang is one of the most visited culinary spot in East Java, so that many vendors, entrepreneur, restaurants and café owner challenged to make a fascinating culinary destination for tourist and local residents, such as student. Related to this issue, a mobile application is needed to be develop, to provide a service to search a culinary destination around Malang. The aim of developing this application is to find a shortest route to culinary destination in Malang.

In this research, the author developed an application that can optimize searching a culinary spot in Malang using the Dijkstra algorithm with Min (DestValue, MarkedValue + EdgeWeight). The search application to find the possible of smallest value on each node that will be passed.

Keywords: Search, Culinary, Djikstra, Mobile Application, Shortest Route.

KATA PENGANTAR

Puji Syukur penulis panjatkan kehadiran Tuhan yang Maha Esa atas segala rahmat penulis dapat menyelesaikan laporan skripsi dengan judul “PENGEMBANGAN APLIKASI *MOBILE* PENCARIAN RUTE TERPENDEK MENUJU LOKASI KULINER KOTA MALANG MENGGUNAKAN ALGORITMA *DIJKSTRA*”. Laporan skripsi ini penulis susun sebagai persyaratan untuk menyelesaikan studi program Diploma IV Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang.

Penulis Menyadari tanpa adanya dukungan dan kerja sama dari berbagai pihak, kegiatan laporan skripsi ini tidak akan dapat berjalan dengan baik. Untuk itu, penulis ingin menyampaikan rasa terima kasih kepada:

1. Bapak Rudy Ariyanto, ST., MCs. selaku Ketua Jurusan Teknologi Informasi Politeknik Negeri Malang.
2. Bapak Ir. Deddy Kusbianto P., M.MKom. selaku Ketua Program Studi Teknik Informatika Politeknik Negeri Malang.
3. Ibu Ely Setyo Astuti, ST., MT. selaku Dosen Pembimbing Skripsi.
4. Bapak Hendra Pradibta, SE., MSC. selaku Dosen Pembimbing Skripsi.
5. Dan seluruh pihak yang telah membantu dan mendukung lancarnya pembuatan Laporan Skripsi dari awal hingga akhir yang tidak dapat kami sebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan laporan akhir ini, masih banyak terdapat kekurangan dan kelemahan yang dimiliki penulis baik itu sistematika penulisan laporan maupun penggunaan bahasa. Untuk itu penulis mengharapkan saran dan kritik dari berbagai pihak yang bersifat membangun demi penyempurnaan laporan ini. Semoga laporan ini berguna bagi pembaca secara umum dan penulis secara khusus. Akhir kata, penulis ucapkan banyak terima kasih.

Malang, 11 Juni 2017

Michael Eduardo Karambut

DAFTAR ISI

	Halaman
HALAMAN COVER.....	i
HALAMAN PENGESAHAN.....	ii
PERNYATAAN.....	iii
ABSTRAK	iv
ABSTRACT.....	v
KATA PENGANTAR	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xi
DAFTAR LAMPIRAN.....	xii
BAB I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan.....	2
1.4 Batasan Masalah.....	2
1.5 Sistematika Penulisan.....	2
BAB II. LANDASAN TEORI	5
2.1. Metode Pencarian Jalur Terpendek (Dijkstra Algorithm).....	5
2.2. Android.....	8
2.3. Google Maps API.....	9
2.4. PHP (Hypertext Preprocessor)	9
2.5. Kuliner.....	10
2.6. Aplikasi Mobile.....	10
BAB III. METODOLOGI PENELITIAN	11
3.1. Pengumpulan Kebutuhan	12
3.2. Membangun Prototype	12
3.3. Evaluasi Prototype.....	13
3.4. Pembuatan Sistem	13
3.5. Menguji Sistem.....	13
3.6. Evaluasi Sistem	14

BAB IV. ANALISA DAN PERANCANGAN.....	15
4.1. Analisis Kebutuhan	15
4.1.1. Deskripsi Umum	15
4.1.2. Data Requirement.....	15
4.2. Perancangan Sistem.....	16
4.2.1. Flowchart	16
4.2.2. Desain Arsitektur	17
4.2.3. Perancangan Algoritma Dijkstra.....	17
4.2.4. Desain Use Case.....	21
BAB V. IMPLEMENTASI	28
5.1 Implementasi Basis Data	28
5.1.1 Tabel Kuliner	28
5.1.2 Tabel Super Admin Kuliner	29
5.1.3 Tabel Rute	29
5.1.4 Tabel Perbandingan Hasil	30
5.1.5 Tabel Perbandingan Hasil Vertex	30
5.2 Implementasi Antar Muka Super Admin Sistem	31
5.2.1 Halaman Login.....	31
5.2.1 Halaman Utama Admin.....	31
5.2.2 Halaman Input Data Kuliner	32
5.2.3 Halaman Data Kuliner.....	32
5.2.4 Halaman Edit Data Kuliner.....	33
5.3 Implementasi Antar Muka User Kuliner	33
5.3.1 Halaman Utama.....	33
5.3.2 Halaman Menu Map.....	34
5.3.3 Halaman Menu Perhitungan Algoritma Dijkstra	34
BAB VI. PENGUJIAN DAN PEMBAHASAN	35
6.1 Uji Coba	35
6.1.1 Pengujian Spesifikasi Perangkat Uji Coba.....	35
6.1.2 Pengujian Performa fungsionalitas Sistem.....	36
6.2. Analisis Hasil Uji Coba Sistem	37
6.3. Hasil Uji Coba dengan Responden.....	37
6.4. Pengujian Algoritma Dijkstra.....	39
BAB VII. KESIMPULAN	42

7.1. Kesimpulan.....	42
7.2. Saran	42
DAFTAR PUSTAKA	43
LAMPIRAN-LAMPIRAN.....	44

DAFTAR GAMBAR

	Halaman
Gambar 3. 1. Diagram <i>Prototype Model</i>	11
Gambar 4. 1 <i>Flowchart</i>	16
Gambar 4.2 Desain arsitektur.....	17
Gambar 4.3 Perancangan Algoritma <i>Dijkstra</i>	18
Gambar 4.4 Perhitungan Algoritma <i>Dijkstra</i>	20
Gambar 4.5 Hasil Perhitungan Algoritma <i>Dijkstra</i>	20
Gambar 4.6 Desain <i>use case</i>	21
Gambar 5.1 Basis data.....	28
Gambar 5.2 Tabel kuliner	28
Gambar 5.3 Tabel super admin kuliner.....	29
Gambar 5.4 Tabel rute.....	29
Gambar 5.5 Tabel perbandingan hasil	30
Gambar 5.6 Tabel perbandingan hasil_vertex	30
Gambar 5.7 Halaman login	31
Gambar 5.8 Halaman utama admin.....	31
Gambar 5.9 Halaman input data kuliner	32
Gambar 5.10 Halaman data kuliner	32
Gambar 5.11 Halaman edit kuliner	33
Gambar 5.12 Halaman utama.....	33
Gambar 5.13 Halaman menu map.....	34
Gambar 5.14 Halaman menu perhitungan algoritma djikstra	34
Gambar 5.15 Pengujian Algoritma Dijkstra.....	39
Gambar 5.16 Halaman jalur A-D-C-J-K-L	41

DAFTAR TABEL

	Halaman
Tabel 4.1 Jarak antar titik.....	18
Tabel 4.2 Jarak antar titik 2.....	19
Tabel 4.3 Tabel Fungsi aktor use case	21
Tabel 4.4 Definisi use case.....	21
Tabel 4.5 Deskripsi Use Case "Login"	22
Tabel 4.6 Deskripsi Use Case "Logout"	23
Tabel 4.7 Deskripsi Use Case "Lihat data Kuliner"	24
Tabel 4.8 Deskripsi Use Case "Olah data Kuliner"	25
Tabel 4.9 Deskripsi Use Case " Marker Lokasi Kuliner"	26
Tabel 4.10 Deskripsi Use Case "Memilih Kuliner "	27
Tabel 6.1 Spesifikasi perangkat keras PC / Laptop.....	35
Tabel 6.2 Spesifikasi Perangkat Keras Smartphone	35
Tabel 6.3 Spesifikasi perangkat lunak PC / Laptop	36
Tabel 6.4 Spesifikasi Perangkat Lunak Smartphone.....	36
Tabel 6.5 Fungsionalitas Sistem	36
Tabel 6.6 Data Hasil Kuesioner	38
Tabel 6.7 Jarak	39
Tabel 6.8 Jarak 2	40
Tabel 6.9 Pengujian algoritma dijkstra	40

DAFTAR LAMPIRAN

- Lampiran 1. Data Lokasi Kuliner
- Lampiran 2. Script Metode Algoritma *Dijkstra*
- Lampiran 3. Lembar Bimbingan Pembimbing 1
- Lampiran 4. Lembar Bimbingan Pembimbing 2
- Lampiran 5. Form Revisi Skripsi Penguji 1
- Lampiran 6. Form Revisi Skripsi Penguji 2
- Lampiran 7. Form Verifikasi Abstrak Bahasa Inggris dan Tata Tulis Buku Skripsi
- Lampiran 8. Kuesioner Pengujian Aplikasi
- Lampiran 9. Biodata Penulis

BAB I. PENDAHULUAN

1.1 Latar Belakang

Saat ini perkembangan teknologi informasi sangat berkembang dengan pesat bahkan tidak terbandung lagi. Baik masalah komunikasi, yang belakangan ini seakan tidak ada jarak lagi, segala sesuatu terasa dekat, itu semua karena teknologi. Di Indonesia perkembangan teknologi semakin terasa manfaatnya tidak hanya dalam komunikasi saja, dalam hal informasi suatu tempat pemanfaatan informasi berpengaruh. Salah satu teknologi yang sedang banyak digunakan adalah *Smartphone*. Dalam perkembangannya *smartphone* kini hadir dengan berbagai macam merk, dan spesifikasi serta harga yang semakin ramah bagi kantong masyarakat Indonesia yang kebanyakan memiliki perekonomian menengah ke bawah. Dengan demikian *smartphone* saat ini tidak lagi menjadi barang yang mewah, sehingga kebutuhan dan pemanfaatan *smartphone* semakin tinggi. Seiring dengan perkembangan ilmu dan teknologi, muncul suatu perkembangan teknologi yaitu GPS (*Global Positioning System*). Bertujuan untuk mengetahui letak tempat yang akan dituju dan mengetahui dimana pengguna berada dengan bantuan sinyal satelit, GPS sendiri dapat memberikan informasi yang tepat dan akurat mengenai posisi, kecepatan, arah dan waktu. GPS ini sendiri sudah tertanam pada *smartphone* canggih dengan sistem operasi Android, dengan demikian setiap pengguna dapat mengetahui dimana posisi pengguna tanpa takut tersesat disuatu tempat yang tidak diketahuinya.

Kota Malang adalah sebuah kota yang terletak di Provinsi Jawa Timur, Indonesia. Kota ini terletak 90 km sebelah selatan Surabaya dan merupakan kota terbesar di kedua di Jawa Timur setelah Surabaya, serta merupakan salah satu kota terbesar di Indonesia menurut jumlah penduduk. Kota Malang dikenal sebagai salah satu kota tujuan Pendidikan terkemuka di Indonesia karena banyak Universitas dan Politeknik Negeri maupun Swasta yang terkenal hingga seluruh Indonesia dan menjadi salah satu tujuan pendidikan berada di kota ini. Kota Malang terkenal sebagai penghasil apel yang enak dan manis, kota Malang juga merupakan salah satu destinasi kuliner yang banyak dikunjungi, oleh Karena itu para pengusaha rumah makan, restoran, kafe sampai kaki lima, mereka berlomba-

lomba untuk menyajikan kuliner yang enak untuk mahasiswa , wisatawan yang datang dan untuk warga Malang sendiri, dalam hal ini tentunya Masyarakat membutuhkan sebuah Aplikasi mobile yang dapat memberikan layanan pencarian lokasi kuliner kota Malang. Pembuatan Aplikasi ini diharapkan mampu meningkatkan efisiensi pencarian rute terdekat menuju lokasi Kuliner di Kota Malang dengan memanfaatkan algoritma pencarian rute terdekat dengan algoritma Dijkstra.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah adalah sebagai berikut:

Bagaimana merancang dan membangun aplikasi *mobile* pencarian Kuliner yang user *friendly* dan dapat membantu pengguna dalam mencari rute terdekat menuju lokasi kuliner?

1.3 Tujuan

Penelitian ini bertujuan untuk merancang dan membangun aplikasi *mobile* pencarian Kuliner di Kota Malang dalam membantu pengguna mencari rute terdekat menuju Kuliner di kota Malang.

1.4 Batasan Masalah

Agar aplikasi ini dapat berjalan sesuai dengan rencana sesuai dengan tujuan awal, maka penulis memberikan batasan-batasan masalah, yaitu:

1. Aplikasi *mobile* pencarian kuliner ini hanya diwilayah Kota Malang.
2. Aplikasi ini menggunakan platform android, sehingga hanya dapat dijalankan pada perangkat (*handphone*) yang berbasis android.
3. Titik awal hanya diwilayah kampus Politeknik Negeri Malang.

1.5 Sistematika Penulisan

Dalam menyusun proposal ini, sistem penulisan yang di gunakan oleh penulis yaitu dengan cara membagi masalah menjadi beberapa tahapan. Dimana pembahasan setiap babnya sebagai berikut :

BAB I PENDAHULUAN

Pada bab ini dijelaskan mengenai hal – hal yang bersifat umum seperti latar belakang, rumusan masalah, batasan masalah, tujuan penulisan, metodologi, dan sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab ini berisikan teori-teori yang mendasari dan berkaitan dengan masalah perencanaan dan pembuatan aplikasi yang digunakan acuan untuk memudahkan pemahaman dan pemecahan terhadap masalah yang ada.

BAB III METODOLOGI PENELITIAN

Pada bab ini terdiri dari metode, teknik, prosedur apa dan tools yang akan digunakan sehingga setiap tahap penelitian dapat dilakukan dengan tepat. Metodologi penelitian antara lain metode pengambilan data, metode pengembangan sistem, fase-fase pengembangan sistem.

BAB IV ANALISIS DAN PERANCANGAN

Pada bab ini menjelaskan Analisis dan Perancangan Aplikasi yang akan dibuat.

BAB VI IMPLEMENTASI

Pada bab ini membahas tentang Pengembangan Aplikasi Mobile Pencarian rute terdekat menuju lokasi Kuliner Kota Malang menggunakan algoritma Dijkstra.

BAB VI PENGUJIAN DAN PEMBAHASAN

Pada bab ini membahas tentang Pengembangan Aplikasi Mobile Pencarian rute terdekat menuju lokasi Kuliner Kota Malang menggunakan algoritma Dijkstra.. Pembahasan berisi tentang hasil ujicoba aplikasi kepada user.

BAB VII PENUTUP

Pada bab ini berisi kesimpulan yang diperoleh setelah melakukan analisis masalah, perancangan, implementasi, pengujian dan analisa hasil pengujian terhadap aplikasi dan saran-saran.

BAB II. LANDASAN TEORI

2.1. Metode Pencarian Jalur Terpendek (*Dijkstra Algorithm*)

Algoritma *Dijkstra* ditemukan oleh orang yang bernama Edsger Wybe *Dijkstra*. Ia merupakan seorang ahli komputer asal Belanda yang lahir di Rotterdam pada tanggal 1 Mei 1930 dan meninggal dunia pada tanggal 6 Agustus 2002. Algoritma *Dijkstra* adalah suatu algoritma dimana algoritma ini digunakan untuk mencari rute permasalahan terpendek antara simpul sumber dan simpul tujuan untuk sebuah graf berarah berdasarkan bobot pada sisi yang bernilai tidak negatif. Algoritma *Dijkstra* bekerja dengan cara mengunjungi simpul-simpul yang ada, dimulai dari simpul sumber. Kemudian algoritma ini memilih simpul-simpul yang lokasinya terdekat dan dilakukan secara berulang lalu kemudian menghitung total bobot semua sisi yang dilewati untuk mencapai simpul tujuan.

Pencarian jalur terpendek merupakan salah satu persoalan dalam teori graf. Persoalan ini bisa diselesaikan dengan algoritma *dijkstra* karena lebih mudah dan menarik, adapun beberapa keuntungan yang kita peroleh dari Algoritma *Dijkstra* yaitu :[7]

1. Menggunakan Algoritma *Dijkstra* mempermudah kita dalam mengetahui jarak atau lintasan terpendek dari suatu titik tertentu ke semua titik yang lain.
2. Menggunakan Algoritma *Dijkstra* dalam penerapan di dalam sistem geografis akan menampilkan visualisasi data dalam bentuk peta
3. Pada penampilan rute atau peta Algoritma *Dijkstra* lebih mudah di baca dan di pahami.
4. Pada rute atau peta dan lintasannya dapat diberikan warna, sehingga penampilan Algoritma *Dijkstra* lebih menarik dan lebih mudah untuk membedakan dari suatu titik awal ke titik tujuan lain.

Adapaun rumus sebagai berikut:

$\text{Min} (\text{DestValue}, \text{MarkedValue} + \text{EdgeWeight})$

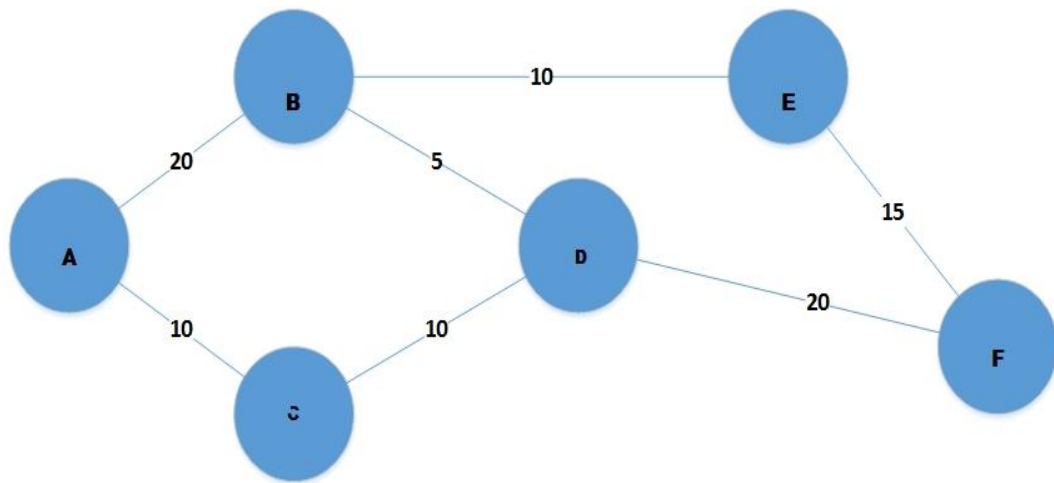
Ket:

DestValue=Nilai dalam vertex tujuan

MarkedValue=Nilai dalam vertex awal

EdgeWeight=bobot dari sisi yang menghubungkan vertex

Dibawah ini adalah contoh kasus perhitungan metode algoritma dijkstra sebagai berikut:[6]



Gambar 3.2 Contoh Graph

Pada penelitian ini penulis menggunakan metode algoritma dijkstra untuk mencari rute terpendek dari titik A menuju ke titik F. Pada kolom *marked* berisikan titik yang telah di *marked*, karena *starting point* di *marked* A di beri nilai 0, sedangkan untuk kolom B-F diberi nilai ∞ karena nilainya belum diketahui.

Tabel 2.1 titik A mencari titik terkecil hasilnya pada titik C

V	A	B	C	D	E	F
A	0	∞	∞	∞	∞	∞
C	0	$\min(\infty, 0+20)$ 20	$\min(\infty, 0+10)$ 10	∞	∞	∞

Kemudian cari titik yang berhubungan dengan titik A yaitu titik B dan C. maka gunakan minimum formula untuk mencari nilai terkecil antara titik B dan titik C yang hasilnya adalah 10. Setelah mendapatkan nilai terkecil antar titik maka nilai terkecil tersebut ditandai dengan angka di tebalkan.

Tabel 2.2 titik A menuju titik C berlanjut pada titik D

V	A	B	C	D	E	F
A	0	∞	∞	∞	∞	∞
C	0	$\min(\infty, 0+20)$ 20	$\min(\infty, 0+10)$ 10	∞	∞	∞
D	0	20	10	$\min(\infty, 10+10)$ 20	∞	∞

Setelah titik C ditentukan sebagai titik bernilai terkecil maka perhitungan dilanjutkan untuk mencari titik yang berhubungan dengan titik C pada contoh graph, titik C hanya berhubungan dengan titik D. Setelah itu masukkan nilai minimum dari titik D.

Tabel 2.3 titik D menuju titik F

V	A	B	C	D	E	F
A	0	∞	∞	∞	∞	∞
C	0	$\min(\infty, 0+20)$ 20	$\min(\infty, 0+10)$ 10	∞	∞	∞
D	0	20	10	$\min(\infty, 10+10)$	∞	∞

				20		
B	0	Min20,20+5) 20	10	20	∞	min(∞ ,20+20) 40

Selanjutnya menghitung yang terhubung dengan titik D yaitu titik B dan F. dari titik D ke titik B akan membandingkan nilai terkecil antara dua jalur titik A-B dan titik A-C-D-F menjadi hasil akhir.

Tabel 2.4 Hasil titik akhir

V	A	B	C	D	E	F
A	0	∞	∞	∞	∞	∞
C	0	min(∞ ,0+20) 20	min(∞ ,0+10) 10	∞	∞	∞
D	0	20	10	min(∞ ,10+10) 20	∞	∞
B	0	min(20,20+5) 20	10	20	∞	min(∞ ,20+20) 40
E	0	20	10	20	min(∞ ,20+10) 30	40
F	0	20	10	20	30	min(40,30+15) 40

2.2. Android

Pada awal peluncurannya, Google meyakini bahwa platform perangkat mobile Android memiliki kesempatan yang sangat besar dalam pengembangan aplikasi. Google mengumumkan Open Handset Alliance (OHA) dan platform Android pada November 2007, dan meluncurkan Android Software Development Kit (SDK) pertama yang masih dalam versi beta di waktu yang sama. Dalam

waktu yang tidak lama, lebih dari satu juta orang mengunduh Android SDK dari website Google. Di Amerika Serikat, T-Mobile mengumumkan perangkat mobile Android bergelar G1 pada Oktober 2008, dan diperkirakan ratusan ribu perangkat G1 terjual pada akhir tahun yang sama. Android memiliki potensi yang besar untuk menghilangkan batasan dan kendala yang selama ini muncul dalam mengembangkan suatu perangkat lunak versi mobile phone. Dari berbagai hal yang telah disiapkan Google, Android berhasil mengembangkan pasar aplikasi 19 perangkat mobile, serta memberikan kesempatan besar untuk para pengembang perangkat lunak untuk mengambil keuntungan dari aplikasi yang dibuatnya [3].

2.3. Google Maps API

Google Maps API adalah kumpulan data suatu peta yang dapat membantu dalam menambahkan data peta di google. Dan dapat membantu dalam membuat aplikasi web atau seluler (android) yang menarik dengan platform pemetaan dari Google, termasuk dalam basis data citra satelit, street view, profil ketinggian, petunjuk arah, analisis yang basis data peta besar. Dengan cakupan global yang paling akurat di dunia, dan komunitas pemetaan yang aktif memperbarui setiap kali pengguna dapat manfaat dari layanan yang ditingkatkan secara terus menerus.[6]

2.4. PHP (Hypertext Preprocessor)

PHP yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan teks yang terintegrasi dengan HTML dan berada pada *server*. *Script PHP* biasa digunakan untuk membuat halaman website dinamis, yang berarti halaman ditampilkan dan dibuat saat halaman itu diminta oleh *client*. Sistem mekanisme ini menyebabkan informasi diterima client selalu terbaru atau terupdate, yang script PHP dieksekusi pada *server* dimana teks tersebut dijalankan.[5]

2.5. Kuliner

Kuliner adalah hasil olahan yang berupa masakan. Masakan tersebut berupa lauk pauk, makanan dan minuman. Karena setiap daerah memiliki cita rasa tersendiri, maka tak heran jika setiap daerah memiliki tradisi kuliner yang berbeda-beda. Kuliner merupakan sebuah gaya hidup yang tidak dapat dipisahkan dari kehidupan sehari-hari.[8]

2.6. Aplikasi Mobile

Aplikasi mobile saat ini sangat dibutuhkan karena alat-alat telekomunikasi yang tersebar di seluruh dunia membutuhkan aplikasi-aplikasi yang dapat mempermudah pekerjaan penggunanya dimanapun dan kapanpun terutama dalam hal informasi. Aplikasi ini dapat diakses melalui perangkat nirkabel seperti pager, seperti telepon seluler dan PDA.[10]

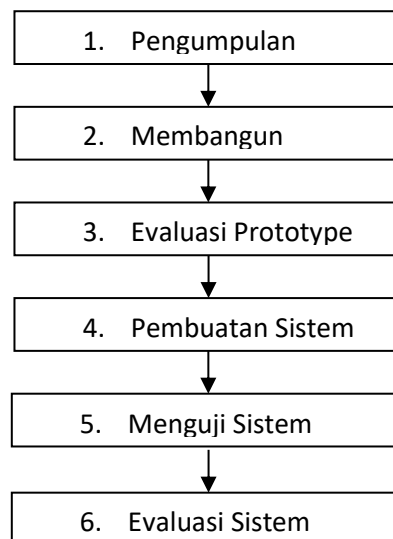
BAB III. METODOLOGI PENELITIAN

Pada bagian ini akan dibahas metodologi yang digunakan peneliti dalam pembuatan Pengembangan Aplikasi *Mobile* Pencarian Rute Terpendek menuju Lokasi Kuliner Kota Malang. Metode penelitian yang digunakan adalah metode *prototype*. *Prototype Model* adalah salah satu metode pengembangan perangkat lunak yang banyak digunakan. Dengan metode ini pembuat program dan *client* bisa berinteraksi selama proses pembuatan sistem berlangsung. Seringkali seorang pelanggan mendefinisikan serangkaian sasaran umum bagi perangkat lunak, tetapi tidak mengidentifikasi kebutuhan input, pemrosesan, ataupun *output detail*. Dalam situasi seperti ini salah satu model yang cocok digunakan adalah model prototipe (*prototyping paradigm*) [2].

Adapun keunggulan dari metode *prototype* adalah:

- a. Adanya komunikasi baik antara pengembang dengan pelanggan.
- b. Pengembang dapat bekerja lebih baik untuk memenuhi kebutuhan pelanggan.
- c. Pelanggan berperan aktif dalam pengembangan sistem.
- d. Menghemat waktu dalam pengembangannya.

Proses kegiatan yang ada pada metode *prototyping* bisa dijelaskan pada Gambar 3.1 sebagai berikut:



Gambar 3. 1. Diagram *Prototype Model*

3.1. Pengumpulan Kebutuhan

Pada tahap pengumpulan kebutuhan, ada beberapa hal yang harus dipenuhi antara lain:

a. Kebutuhan Informasi

Dalam mengimplementasikan Pengembangan Aplikasi *Mobile* Pencarian rute terpendek menuju lokasi Kuliner Kota Malang menggunakan algoritma Dijkstra, diperlukan pengetahuan dan informasi yang terkait dengan topik yang diambil. Pengumpulan informasi dilakukan dengan berbagai sumber berupa paper, jurnal dalam bentuk cetak maupun melalui website.

b. Kebutuhan Data

Kebutuhan data yang diperlukan pada Pengembangan Aplikasi *Mobile* Pencarian rute terpendek menuju lokasi Kuliner Kota Malang adalah data kuliner yang terdapat di beberapa kota Malang. Data tersebut berupa nama kuliner, alamat, yang akan digunakan dalam pengimplentasi aplikasi *mobile* pencarian kuliner.

c. Kebutuhan *Software* dan *Hardware*

Software yang digunakan dalam Pengembangan Aplikasi *Mobile* Pencarian rute terpendek menuju lokasi Kuliner Kota Malang menggunakan algoritma Dijkstra, adalah *Android Studio*, Sublime Text 3 (text editor), XAMPP Sedangkan *Hardware* yang dibutuhkan adalah laptop dan perangkat *mobile* (*Handphone Android*)

3.2. Membangun *Prototype*

Pada tahap ini akan dibuat rancangan mengenai sistem yang akan dibuat untuk Pengembangan Aplikasi *Mobile* Pencarian rute terpendek menuju lokasi Kuliner Kota Malang menggunakan algoritma Dijkstra. Rancangan tersebut berupa algoritma perhitungan Dijkstra yang menjelaskan tentang pencarian rute terdekat dalam proses perhitungan bobot terkecil dengan algoritma Dijkstra terkait dengan pengembangan Aplikasi *mobile* Pencarian Kuliner. Dan akan dijelaskan juga tentang gambaran sistem yang berjalan pada Aplikasi *Mobile* Pencarian Kuliner Kota Malang.

3.3. Evaluasi Prototype

Tahap ini akan mengevaluasi perancangan yang sudah dibuat pada tahap sebelumnya, yaitu melakukan evaluasi terhadap perhitungan algoritma Dijkstra dan gambaran Aplikasi *Mobile* Pencarian Kuliner Kota Malang. Apabila hasil dari evaluasi tersebut sesuai dengan tujuan yang diinginkan maka dilanjutkan pada tahap selanjutnya. Namun apabila belum sesuai, maka harus dilakukan perbaikan mulai tahap pertama.

3.4. Pembuatan Sistem

Proses pembuatan sistem atau yang biasa disebut implementasi merupakan tahap untuk menerapkan yang telah diperoleh dari kegiatan sebelumnya. Dalam hal ini yang dilakukan adalah implementasi pengkodean sistem dari perancangan Pencarian Kuliner Kota Malang.

3.5. Menguji Sistem

Proses Pengembangan Aplikasi *Mobile* Pencarian rute terpendek menuju lokasi Kuliner Kota Malang menggunakan algoritma Dijkstra. dilakukan untuk mengetahui apakah sistem pada aplikasi sudah dapat berjalan dengan baik dan sesuai kebutuhan sistem yang sudah dirancang.

Tahap uji coba sistem dibagi menjadi 2 yaitu uji coba fungsional dan uji coba akurasi. Uji coba fungsional dilakukan untuk mengetahui apakah sistem yang dibangun sudah menyediakan fungsi-fungsi yang sesuai dengan kebutuhan. Sedangkan uji coba akurasi dilakukan untuk mengetahui performa dari algoritma Dijkstra dalam Aplikasi *Mobile* Pencarian Kuliner Kota Malang. Dalam hal ini, uji coba akurasi dilakukan dengan cara membuat titik koordinat manual pada tiap Kuliner yang ada di kota Malang kemudian pengguna mencari rute terdekat menuju tempat kuliner dengan cara menghitung bobot terkecil atau jarak terpendek.

3.6. Evaluasi Sistem

Tahap ini digunakan untuk mengevaluasi dan memperbaiki Pengembangan Aplikasi *Mobile* Pencarian rute terpendek menuju lokasi Kuliner Kota Malang menggunakan algoritma Dijkstra. jika Aplikasi *mobile* tersebut belum berjalan sesuai dengan kebutuhan dan hasil pengujian sistem yang diharapkan.

BAB IV. ANALISA DAN PERANCANGAN

4.1. Analisis Kebutuhan

Dibagian bawah ini adalah pembahasan untuk kebutuhan perangkat lunak yang dibutuhkan dalam pembahasan aplikasi dan perancangan desain sistem untuk pembuatan alur proses aplikasi.

4.1.1. Deskripsi Umum

Sistem Aplikasi Pencarian rute terpendek menuju lokasi Kuliner Kota Malang menggunakan algoritma Dijkstra yang dapat membantu pengguna untuk memilih kuliner dan mencari rute terdekat menuju lokasi Kuliner di kota Malang.

Aplikasi ini memiliki 2 pengguna yaitu *admin sistem aplikasi*, dan *Client* berbasis aplikasi *mobile*. *Admin sistem aplikasi* memiliki peran dalam manajemen data (penambahan, perubahan dan pengurangan) data Kuliner. *Client* memiliki peran memilih kuliner dan mencari rute terdekat menuju Kuliner yang mereka inginkan di kota Malang.

4.1.2. Data Requirement

Sistem Informasi ini akan mengelola data Kuliner di Kota Malang sesuai batasan masalah tersebut antara lain :

❖ **Data Kuliner**

Data Kuliner yang digunakan adalah kuliner yang ada di Kota Malang. Data tersebut berisi, nama kuliner, alamat.

❖ **Data Google**

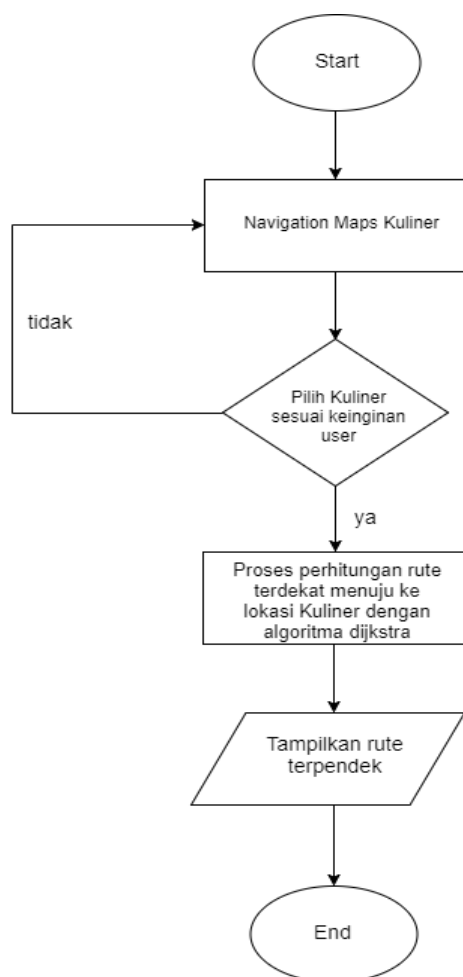
Data dari google *maps* digunakan untuk melakukan pencarian lokasi sesuai pencarian yang dilakukan oleh *Client*.

- *Latitude dan Longitude Kuliner*

4.2. Perancangan Sistem

Perancangan sistem yang digunakan untuk menggambarkan alur proses sistem Aplikasi Mobile Pencarian rute terdekat menuju lokasi Kuliner kota Malang menggunakan algoritma Dijkstra yaitu sebagai berikut :

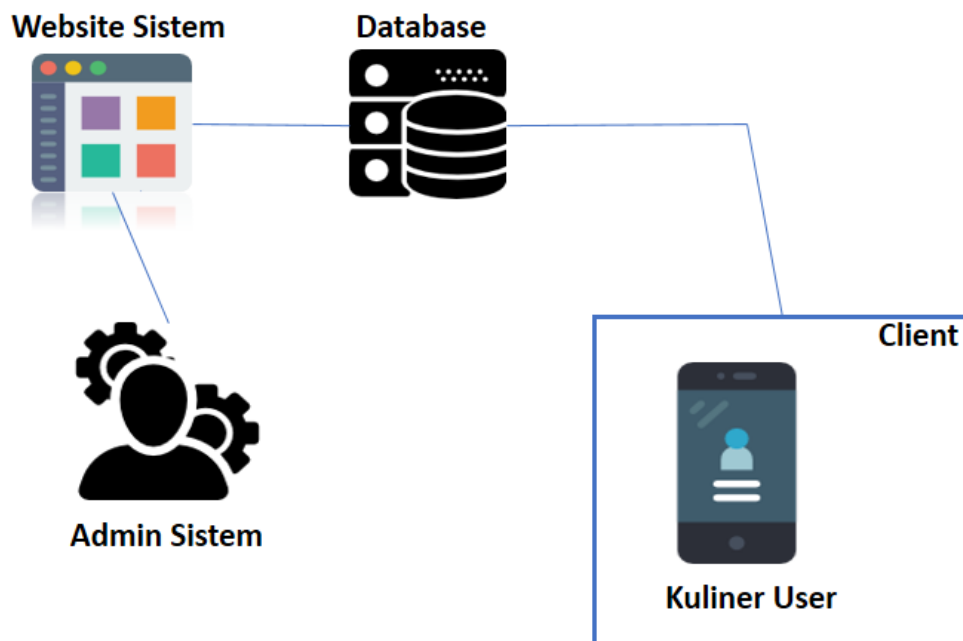
4.2.1. Flowchart



Gambar 4. 1 Flowchart

Ketika membuka aplikasi pertama kali pengguna akan memilih maps berdasarkan navigation kemudian memilih kuliner yang diinginkan, setelah itu proses perhitungan pencarian rute terdekat ke lokasi Kuliner menggunakan algoritma Dijkstra.

4.2.2. Desain Arsitektur



Gambar 4.2 Desain arsitektur

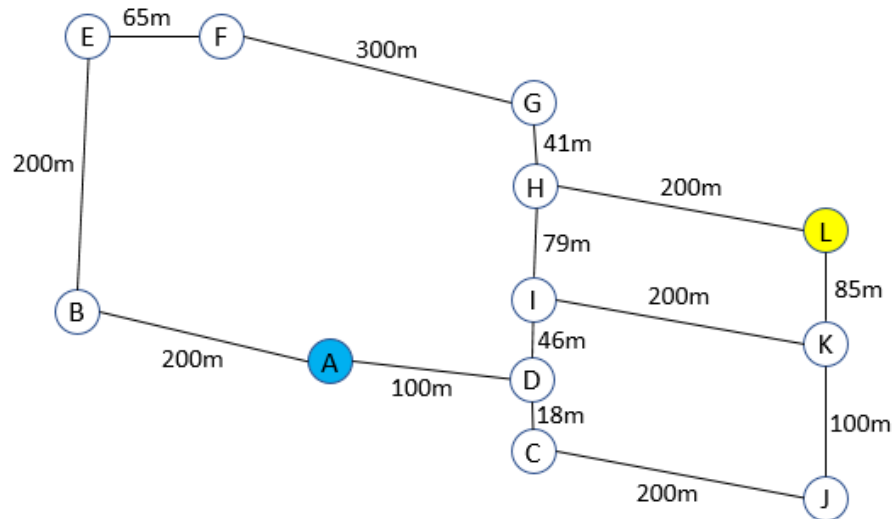
Aplikasi ini memiliki 2 pengguna yaitu *admin sistem aplikasi*, dan *Client* berbasis aplikasi *mobile*. *Admin sistem aplikasi* memiliki peran dalam manajemen data (penambahan, perubahan dan pengurangan) data Kuliner. *Client* memiliki peran memilih kuliner dan mencari rute terdekat menuju Kuliner yang mereka inginkan di kota Malang.

4.2.3. Perancangan Algoritma Dijkstra

Pada perancangan metode *dijkstra* di bawah ini ditunjukkan start awal atau titik awalnya berada pada titik A (Pertigaan pintu keluar belakang kampus politeknik negeri malang) yang berada pada koordinat -7.944981318454669, 112.61453568935394 dan untuk titik tujuannya berada pada titik L (warung bu sri rahayu) yang berada pada koordinat -7.94457405686306, 112.61815935373306, gambar perancangan metode *dijkstra* di tunjukan pada gambar 4.3 Di antara setiap titik - titik memiliki jarak untuk detail setiap jarak antar titik di tunjukan pada tabel 4.1 dan tabel 4.2 Pada gambar 4.4 adalah perhitungan algoritma dijkstra

yang dilakukan di microsoft excel dan hasil perhitungan algoritma dijkstra ditunjukkan pada gambar 4.5.

Di bawah ini adalah perancangan metode dijkstra akan ditunjukkan pada gambar 4.3



Gambar 4.3 Perancangan Algoritma *Dijkstra*

Di bawah ini adalah jarak antar titik akan ditunjukkan pada tabel 4.1

Dari		Jarak
A	B	200m
A	D	100m
B	A	200m
B	E	200m
C	D	18m
C	J	200m
D	A	100m
D	C	18m
D	I	46m
E	B	200m
E	F	65m

Tabel 4.2 Tabel jarak antara titik 2

Dari		Jarak
F	E	65m
F	G	300m
G	F	300m
G	H	41m
H	G	41m
H	I	79m
H	L	200m
I	D	46m
I	H	79m
I	K	200m
J	C	200m
J	K	100m
K	I	200m
K	J	100m
K	L	85m
L	H	200m
L	K	85m

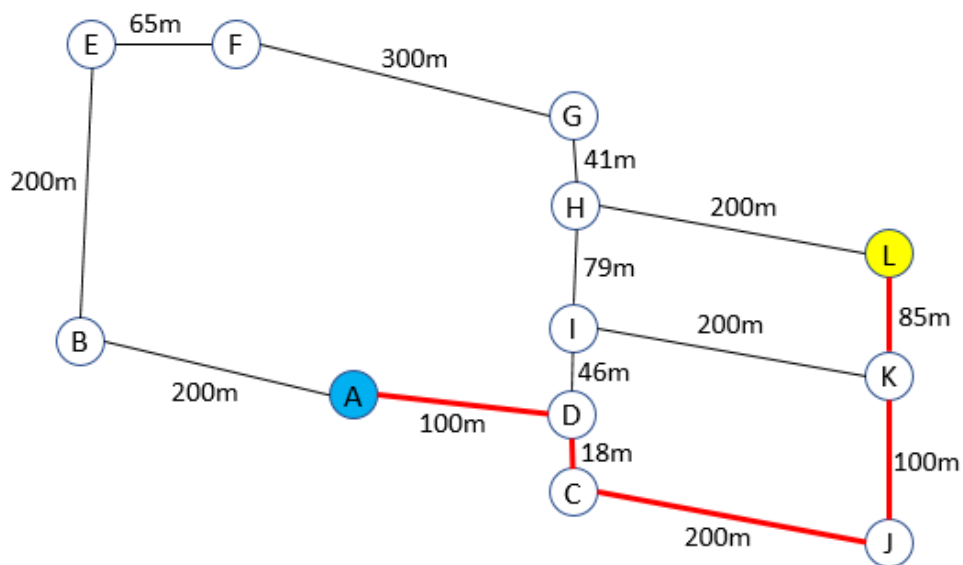
Data pada tabel 4.1 dan tabel 4.2 di atas di dapat dari node yang sudah secara manual lalu di simpan pada *database* android_api tabel rute *field* id, id_tujuan, lattitude, longitude, titik_tujuan, simpul_1, simpul_2, simpul_3 untuk data jarak di dapat dari bantuan aplikasi google maps lalu di simpan pada *database* android_api tabel perbandingan_hasil _vertex *field* jarak.

Di bawah ini adalah perhitungan algoritma dijkstra akan ditunjukkan pada gambar 4.4.

LANGKAH	PILIH	MIN	A	B	C	D	E	F	G	H	I	J	K	L
1	A	100		200		100								
		D												
					100+18						100+46			
2	D	118	100,A		118						146			
		C												
												118+200		
3	C	318			118,D							318		
		J												
													318+100	
4	J	418			318,C								418	
		K												
						418+200								418+85
5	K	503				618							418,J	503
		L												

Gambar 4.4 Perhitungan Algoritma *Dijkstra*

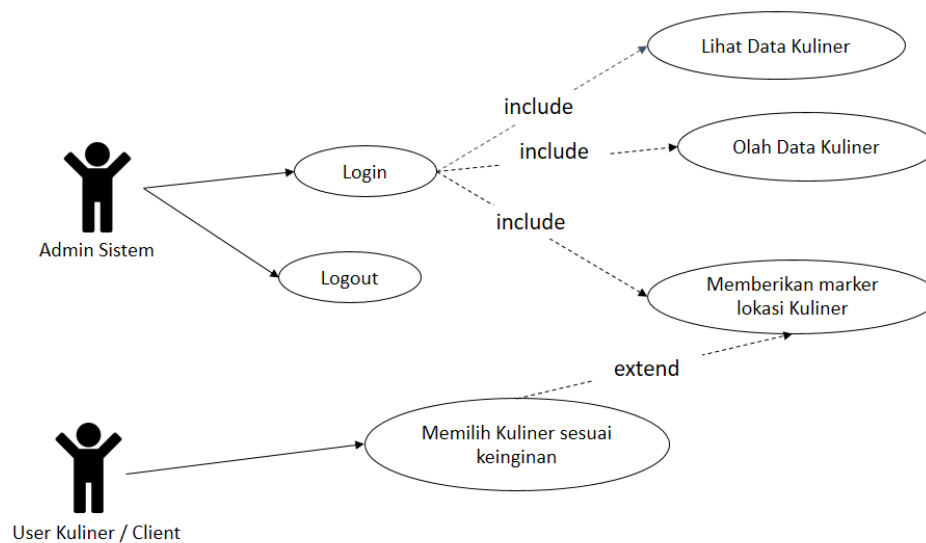
Di bawah ini adalah hasil perhitungan algoritma dijkstra akan ditunjukkan pada gambar 4.5.



Gambar 4.5 Hasil Perhitungan Algoritma *Dijkstra*

Berdasarkan data pada gambar 4.11 dan perhitungan yang telah dilakukan pada gambar 4.12 di dapatkan hasil perhitungan metode *dijkstra* pada gambar 4.13 maka di dapat jalur terpendek yaitu pada jalur A – D – C – J – K – L dengan jarak 503 meter. Panjang jalan di dapat dari *database database android_api* tabel *perbandingan_hasil _vertex field* jarak.

4.2.4. Desain Use Case



Gambar 4.6 Desain use case

Deskripsi pendefinisian aktor pada sistem pencarian jalur terpendek menuju lokasi kuliner kota Malang yang di sajikan pada tabel 4.1

Tabel 4.3 Tabel Fungsi aktor use case

No	Aktor	Kebutuhan
1	Admin	Orang yang bertugas mengolah data Kuliner
2	Client	Pengguna yang melakukan pencarian Kuliner berdasarkan kriteria -kriteria yg diinginkan

Selanjutnya, pendefinisian *use case* dijelaskan secara detail dan terperinci sebagaimana yang disajikan pada tabel 4.2 Pada tabel tersebut menjelaskan fitur – fitur dan menu yang ada pada aplikasi yang dibangun.

Tabel 4.4 Definisi use case

No	Nama Use Case	Deskripsi	Pengguna
1	<i>Login</i>	Berisi form yang digunakan untuk masuk kedalam sistem	<i>Admin</i>

2	<i>Logout</i>	Berisi form yang digunakan untuk keluar dari sistem	<i>Admin</i>
3	Lihat Data Kuliner	Berisi data – data kuliner yang sudah tersimpan sebelumnya di basis data	<i>Admin</i>
4	Olah Data Kuliner	Berisi data – data kuliner yaitu data kuliner, jenis kuliner, alamat, latitude, longitude kuliner malang.	<i>Admin</i>
5	Memberikan <i>marker</i> lokasi kuliner	Berisi data kuliner yang sudah selesai diolah dan siap untuk ditampilkan ke <i>Client</i>	<i>Admin</i>
6	Memilih kuliner sesuai keinginan	Berisi data kuliner yang lengkap, pengguna bisa memilih lokasi kuliner	<i>Client</i>

Selanjutnya, untuk pendeskripsian setiap *use case* akan disajikan pada tabel-tabel berikut ini yang ditampilkan pada Tabel 4.3, Tabel 4.4, Tabel 4.5, Tabel 4.6, Tabel 4.7, Tabel 4.8 dan Tabel 4.9.

Tabel 4.5 Deskripsi Use Case "Login"

Use Case Name: <i>Login</i>	ID: UC.01 Siklus1	<i>Importance Level:</i> <i>High</i>
<i>Primary Actor: Admin</i>	<i>Use Case Type:</i>	
<i>Stakeholder and Interest:</i> <i>Admin</i> masuk kedalam sistem untuk mengolah data yang ada.		
<i>Brief Description:</i> Menjelaskan proses inisialisasi pengguna		
<i>Trigger: Admin</i> masuk kedalam sistem untuk mengelola data kuliner yang ada. <i>Type: internal</i>		
<i>Relationship:</i> <i>Association: Admin</i> <i>Include:</i>		

<p><i>Extend:</i></p> <p><i>Generalization:</i></p>
<p><i>Normal flow of event:</i></p> <ol style="list-style-type: none"> 1. <i>Admin</i> memasukkan <i>username</i> dan <i>password</i> 2. <i>Admin</i> menekan tombol <i>enter</i> untuk <i>login</i> 3. Sistem memeriksa kebenaran data yang diberikan 4. Jika data yang dimasukkan sesuai, maka <i>login user</i> diterima. <p>Sistem menampilkan halaman yang sesuai dengan hak akses yang diberikan.</p>
<p><i>Subflows:</i></p> <ol style="list-style-type: none"> 1a. Jika <i>username</i> dan <i>password</i> yang diinputkan kosong, maka akan muncul alert "<i>Username atau Password salah</i>" 1b. Jika <i>username</i> dan <i>password</i> yang diinputkan salah, maka akan muncul alert "<i>Username atau Password salah</i>"

Tabel 4.6 Deskripsi Use Case "Logout"

<i>Use Case Name:</i> <i>Logout</i>	ID: UC.02 Siklus2	<i>Importance Level:</i> <i>High</i>
<i>Primary Actor: Admin</i>	<i>Use Case Type:</i>	
<i>Stakeholder and Interest:</i> <i>Admin</i> keluar dari sistem setelah selesai melakukan pengolahan data atau aktivitas yang diperlukan sudah selesai.		
<i>Brief Description:</i> Menjelaskan proses inisialisasi pengguna		
<i>Trigger: Admin</i> masuk kedalam sistem untuk mengelola data kuliner yang ada. <i>Type: internal</i>		

<i>Relationship:</i> <i>Association: Admin</i> <i>Include: Login</i>
<i>Extend:</i> <i>Generalization:</i>
<i>Normal flow of event:</i> 1. <i>Admin</i> menekan tombol <i>Logout</i> untuk melakukan <i>logout</i> . 2. Sistem akan menampilkan <i>form Login</i> .
<i>Subflows:</i> 2a. Setelah menekan tombol atau button <i>Logout</i> , maka akan muncul halaman <i>Logout</i> .

Tabel 4.7 Deskripsi Use Case "Lihat data Kuliner"

<i>Use Case Name:</i> Lihat data Kuliner	ID: UC.03 Siklus3	<i>Importance Level:</i> High
<i>Primary Actor: Admin</i>	<i>Use Case Type:</i>	
<i>Stakeholder and Interest:</i> Untuk melihat data yang sudah tersimpan di basis data.		
<i>Brief Description:</i> Menjelaskan proses <i>admin</i> melihat data yang ada		
<i>Trigger: Admin</i> melihat data yang ada. <i>Type: internal</i>		
<i>Relationship:</i> <i>Association: Admin</i> <i>Include: Login</i>		
<i>Extend:</i> <i>Generalization:</i>		

<i>Normal flow of event:</i> 1. Jika <i>user</i> menekan tombol <i>login</i> , maka otomatis akan tampil <i>form</i> yang berisi data2 yang ada di dalam basis data
<i>Subflows:</i>

Tabel 4.8 Deskripsi Use Case "Olah data Kuliner"

<i>Use Case Name:</i> Olah data Kuliner	ID: UC.04 Siklus4	<i>Importance Level:</i> High
<i>Primary Actor: Admin</i>	<i>Use Case Type:</i>	
<i>Stakeholder and Interest:</i> Admin memasukkan data dan mengolah data yang diperlukan dalam basis data.		
<i>Brief Description:</i> Menjelaskan proses <i>admin</i> mengelola data agar siap ditampilkan dalam sistem		
<i>Trigger: Admin</i> mengelola data yang ada. <i>Type: internal</i>		
<i>Relationship:</i> <i>Association: Admin</i> <i>Include: Login</i>		
<i>Extend:</i> <i>Generalization:</i>		
<i>Normal flow of event:</i> 1. Jika <i>user</i> menekan tombol <i>login</i> , maka otomatis akan tampil <i>form</i> yang berisi data2 yang ada di dalam basis data dan <i>admin</i> bisa melakukan pengolahan data seperti CRUD		
<i>Subflows:</i> 1a. <i>Admin</i> menekan tombol atau <i>button</i> simpan, apabila data berhasil tersimpan di <i>database</i> akan tampil pesan “data berhasil disimpan” dan jika data tidak berhasil disimpan di <i>database</i> maka akan tampil pesan “		

data tidak berhasil di simpan”.

4b. Jika *Admin* di *form* ubah menekan *button* simpan setelah melakukan beberapa data yang di ubah, apabila data tersimpan ke *database* sukses akan tampil pesan “berhasil diubah” jika data tidak tersimpan ke *database* maka akan tampil pesan “data gagal diubah”.

4c. Jika *Admin* menekan *button* hapus maka akan tampil pesan “data berhasil dihapus” ketika data yang di inginkannya berhasil dihapus dari *database* jika tidak berhasil dihapus maka akan tampil pesan “data gagal dihapus”.

Tabel 4.9 Deskripsi Use Case " Marker Lokasi Kuliner"

<i>Use Case Name:</i> Memberikan <i>marker</i> Lokasi Kuliner	ID: UC.05 Siklus5	<i>Importance Level:</i> <i>High</i>
<i>Primary Actor: Admin</i>	<i>Use Case Type:</i>	
<i>Stakeholder and Interest:</i> Untuk memberikan penanda semua lokasi Kuliner di peta.		
<i>Brief Description:</i> Menjelaskan proses <i>Admin</i> memberikan penanda lokasi semua Kuliner pada peta secara otomatis ketika ada data yang masuk di basis data		
<i>Trigger: Admin</i> menampilkan lokasi semua Kuliner <i>Type: internal</i>		
<i>Relationship:</i> <i>Association: Admin</i> <i>Include: Login</i>		
<i>Extend:</i> <i>Generalization:</i>		
<i>Normal flow of event :</i> 1. <i>Admin</i> menekan tombol atau <i>button</i> simpan, data yang dimasukkan secara otomatis akan tampil di peta		

Subflows:

Tabel 4. 10 Deskripsi Use Case "Memilih Kuliner "

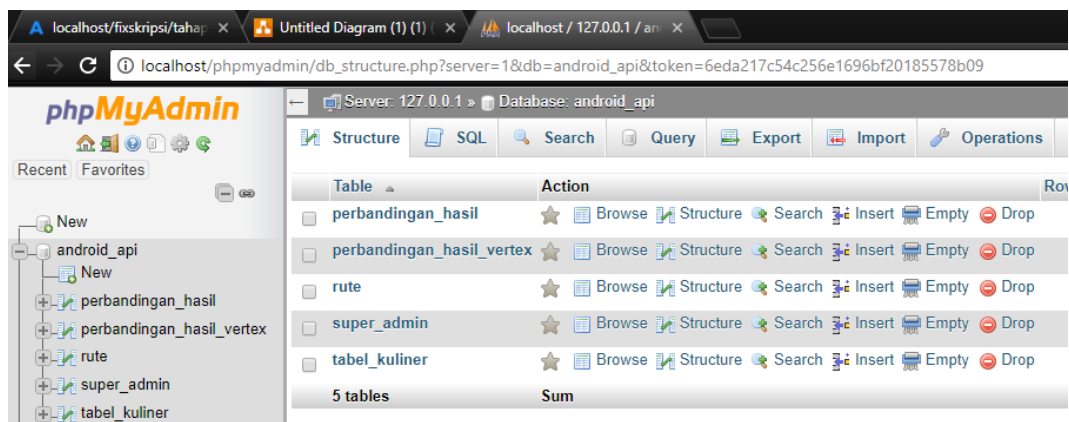
<i>Use Case Name:</i> Memilih Kuliner sesuai kriteria	ID: UC.06 Siklus6	<i>Importance Level:</i> High
<i>Primary Actor: Client</i>	<i>Use Case Type:</i>	
<i>Stakeholder and Interest:</i> Pengguna memilih Kuliner yang diinginkan.		
<i>Brief Description:</i> Menjelaskan proses pengguna memilih Kuliner yang diinginkan		
<i>Trigger:</i> Pengguna memilih Kuliner <i>Type: internal</i>		
<i>Extend:</i> <i>Generalization:</i>		
<i>Normal flow of event :</i> 1. <i>Client</i> menekan navigation maps kemudian memilih kuliner yang diinginkan		
<i>Subflows:</i>		

BAB V. IMPLEMENTASI

Setelah melakukan tahapan perancangan aplikasi, maka pada bab ini akan dibahas tentang proses implementasi, yaitu realisasi perancangan menjadi nyata

5.1 Implementasi Basis Data

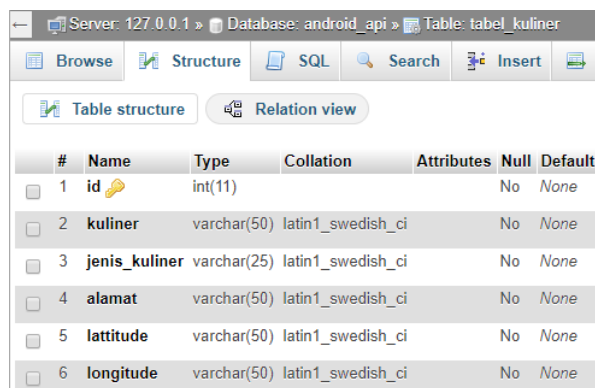
Implementasi basis data merupakan hasil implementasi dari perancangan basis data yang sebelumnya telah dilakukan, terdapat 5 tabel yang dibuat menggunakan MySQL seperti yang ditunjukkan pada Gambar 5.1



Gambar 5.1 Basis data

5.1.1 Tabel Kuliner

Pada Gambar 5.2 adalah implementasi dari rancangan tabel kuliner, berfungsi menyimpan data kuliner, memiliki field antara yaitu id, kuliner, jenis_kuliner, alamat, latitude, longitude.

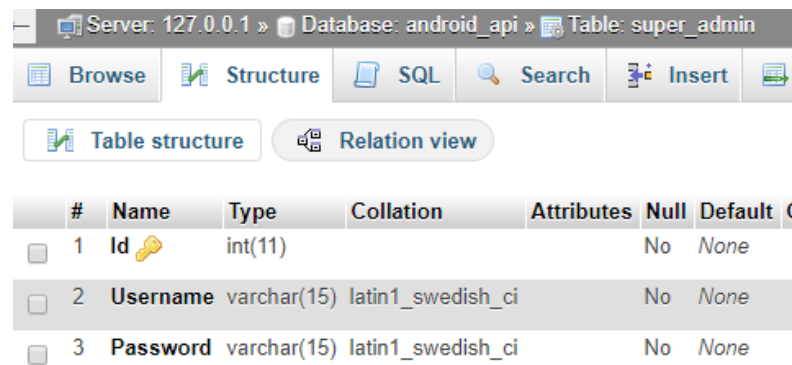
The image shows a screenshot of the phpMyAdmin web interface, specifically the 'Table structure' view for the 'tabel_kuliner' table. The table structure is displayed in a table format with columns: #, Name, Type, Collation, Attributes, Null, and Default. The table has 6 columns: id (int(11)), kuliner (varchar(50)), jenis_kuliner (varchar(25)), alamat (varchar(50)), latitude (varchar(50)), and longitude (varchar(50)). All columns are set to 'No' for Null and 'None' for Default.

#	Name	Type	Collation	Attributes	Null	Default
1	id	int(11)			No	None
2	kuliner	varchar(50)	latin1_swedish_ci		No	None
3	jenis_kuliner	varchar(25)	latin1_swedish_ci		No	None
4	alamat	varchar(50)	latin1_swedish_ci		No	None
5	latitude	varchar(50)	latin1_swedish_ci		No	None
6	longitude	varchar(50)	latin1_swedish_ci		No	None

Gambar 5.2 Tabel kuliner

5.1.2 Tabel Super Admin Kuliner

Pada Gambar 5.3 adalah implementasi dari rancangan tabel super admin kuliner, berfungsi untuk menyimpan data super admin kuliner, memiliki field antara lain yaitu id, Username, Password.

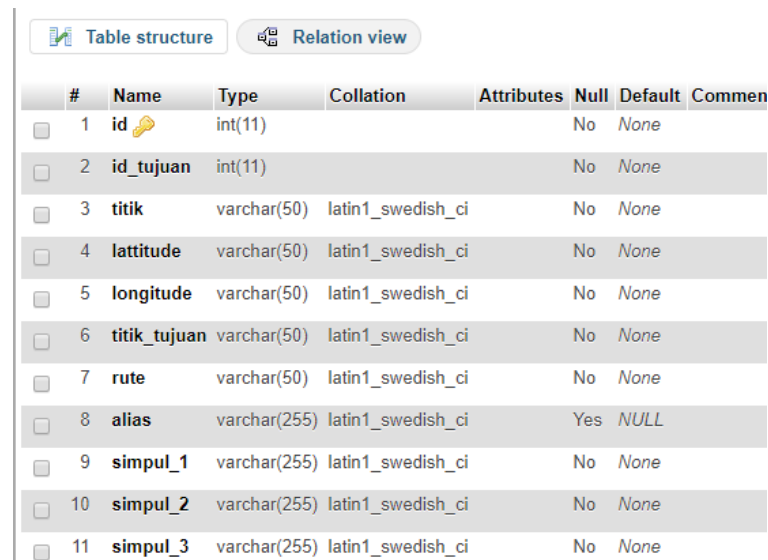


#	Name	Type	Collation	Attributes	Null	Default
1	Id	int(11)			No	None
2	Username	varchar(15)	latin1_swedish_ci		No	None
3	Password	varchar(15)	latin1_swedish_ci		No	None

Gambar 5.3 Tabel super admin kuliner

5.1.3 Tabel Rute

Pada Gambar 5.4 adalah implementasi dari rancangan tabel rute, berfungsi untuk menyimpan data rute, memiliki field antara lain yaitu id, id_tujuan, titik, latitude, longitude, titik_tujuan, rute, alias, simpul_1, simpul_2, simpul_3.

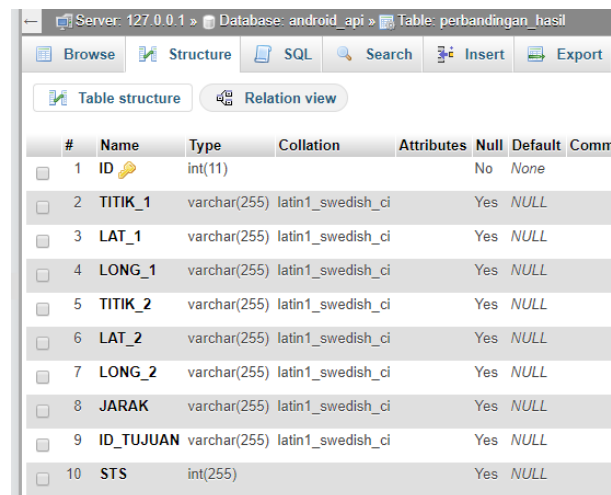


#	Name	Type	Collation	Attributes	Null	Default	Comment
1	id	int(11)			No	None	
2	id_tujuan	int(11)			No	None	
3	titik	varchar(50)	latin1_swedish_ci		No	None	
4	latitude	varchar(50)	latin1_swedish_ci		No	None	
5	longitude	varchar(50)	latin1_swedish_ci		No	None	
6	titik_tujuan	varchar(50)	latin1_swedish_ci		No	None	
7	rute	varchar(50)	latin1_swedish_ci		No	None	
8	alias	varchar(255)	latin1_swedish_ci		Yes	NULL	
9	simpul_1	varchar(255)	latin1_swedish_ci		No	None	
10	simpul_2	varchar(255)	latin1_swedish_ci		No	None	
11	simpul_3	varchar(255)	latin1_swedish_ci		No	None	

Gambar 5.4 Tabel rute

5.1.4 Tabel Perbandingan Hasil

Pada Gambar 5.5 adalah implementasi dari rancangan tabel perbandingan_hasil, berfungsi untuk menyimpan data perbandingan hasil, memiliki field antara lain yaitu id, titik_1, lat_1,long_1, titik_2, lat_2,long_2, jarak,id_tujuan,sts.



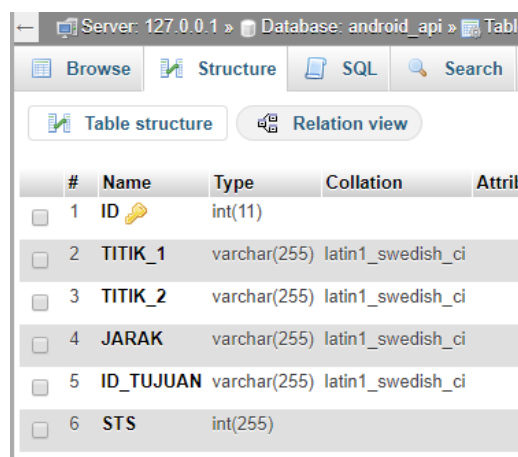
The screenshot shows a database management interface with the following table structure for 'perbandingan_hasil':

#	Name	Type	Collation	Attributes	Null	Default	Comm
1	ID	int(11)			No	None	
2	TITIK_1	varchar(255)	latin1_swedish_ci		Yes	NULL	
3	LAT_1	varchar(255)	latin1_swedish_ci		Yes	NULL	
4	LONG_1	varchar(255)	latin1_swedish_ci		Yes	NULL	
5	TITIK_2	varchar(255)	latin1_swedish_ci		Yes	NULL	
6	LAT_2	varchar(255)	latin1_swedish_ci		Yes	NULL	
7	LONG_2	varchar(255)	latin1_swedish_ci		Yes	NULL	
8	JARAK	varchar(255)	latin1_swedish_ci		Yes	NULL	
9	ID_TUJUAN	varchar(255)	latin1_swedish_ci		Yes	NULL	
10	STS	int(255)			Yes	NULL	

Gambar 5.5 Tabel Perbandingan Hasil

5.1.5 Tabel Perbandingan Hasil Vertex

Pada Gambar 5.6 adalah implementasi dari rancangan tabel perbandingan_hasil, berfungsi untuk menyimpan data perbandingan hasil, memiliki field antara lain yaitu id, titik_1, titik_2, jarak, id_tujuan, sts.



The screenshot shows a database management interface with the following table structure for 'perbandingan_hasil':

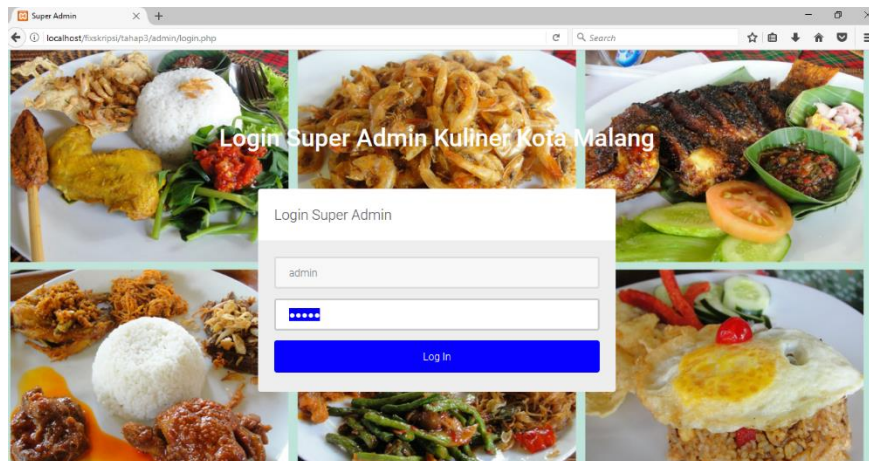
#	Name	Type	Collation	Attrib
1	ID	int(11)		
2	TITIK_1	varchar(255)	latin1_swedish_ci	
3	TITIK_2	varchar(255)	latin1_swedish_ci	
4	JARAK	varchar(255)	latin1_swedish_ci	
5	ID_TUJUAN	varchar(255)	latin1_swedish_ci	
6	STS	int(255)		

Gambar 5.6 Tabel Perbandingan Hasil Vertex

5.2 Implementasi Antar Muka Super Admin Sistem

5.2.1 Halaman Login

Halaman yang pertama kali di akses admin sistem sebelum menuju halaman admin.



Gambar 5.7 Halaman login

5.2.1 Halaman Utama Admin

Halaman Utama Super Admin setelah melakukan login, menampilkan sekilas data kuliner yang sudah diinputkan.

Super Admin Kulliner					
Log Out					
Data Kuliner					
Id	Kuliner	Jenis	Alamat	Latitude	Longitude
17	Warung Mbak Aml	nasi	Jl. Semanggi Bar. No.21A	-7.945745656612625	112.61682629585266
23	Ayam Goreng Nelongso	nasi	Jl. Soekarno Hatta No.37	-7.9465238981351645	112.61765241622925
24	Kober Mie Setan	mie	Jl. Soekarno Hatta	-7.948497638109266	112.61694833636284
25	McDonald's Watugong	nasi	Jl. M.T. Haryono No. 115,	-7.9469170530189555	112.61267423629761
26	Deo Bakso Beranak	bakso	Jl. Semanggi Timur No.4 Jatimulyo	-7.947520066872427	112.61947631835938
27	Mie Jogging	mie	Jl. Soekarno Hatta No.24	-7.945724304946267	112.61875748634338
28	Bakso Pakdhe Remujung	bakso	Jl. Remujung No.26 Jatimulyo	-7.9441583197537256	112.61590629816055
29	Warung Sambel Uleg	nasi	Jl. Pisang Kipas Bar	-7.9412348657659875	112.6134467124939
30	Warung Prasmanan Andong	nasi	Jl. Bunga Andong No.13	-7.949005017914714	112.6187413930893
31	Warung Bu Lasminah	nasi	Jl. Kembang Turi Jatimulyo	-7.942804846823671	112.61451423168182
32	Sop Ayam Pak Min Klaten	nasi	Jl. M.T. Haryono No.27 Dinoyo	-7.944635155572071	112.61120975017548
33	politeknik negeri malang	kampus	Jl. Suhat Malang No.9, Jatimulyo	-7.946553	112.615211
36	RM Barelang	nasi	Jl. Pisang Kipas No.26 Jatimulyo	-7.942666709983261	112.6168692111969

Gambar 5.8 Halaman utama admin

5.2.2 Halaman Input Data Kuliner

Halaman Super Admin Sistem ketika ingin melakukan penambahan data Kuliner.

Super Admin Kulinier

Log Out

Lihat Data

Input Kuliner

Input Data Kuliner [Semua Data](#)

Kuliner

Kuliner

Jenis Kuliner

Jenis Kuliner

Alamat

Alamat

Coordinate

Latitude Longitude [Check Coordinate](#)

[Simpan](#) [Reset](#)

Map Satellite

Surabaya

Pulau Mado

Pamekasan

Mojoagung

Sidoarjo

EAST JAVA

Kediri

Malang

Probolinggo

Tulungagung

Nggalek

Lumajang

Google

Map data ©2017 Google

Terms of Use

Report a map error

Gambar 5.9 Halaman input data kuliner

5.2.3 Halaman Data Kuliner

Halaman Super Admin Sistem dalam melihat data kuliner yang telah ditambahkan sebelumnya.

Super Admin Kulinier

Log Out

Lihat Data

Input Kuliner

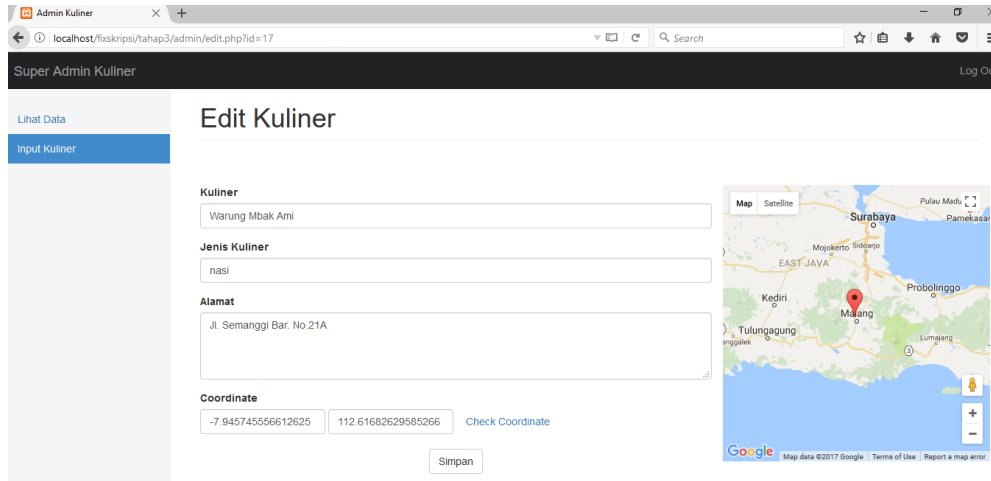
[Input Data Kuliner](#) [Semua Data](#)

ID	Kuliner	Jenis	Alamat	Latitude	Longitude	Edit	Delete
17	Warung Mbak Aml	nasi	Jl. Semanggi Bar. No.21A	-7.945745556612625	112.61682629585266	Edit	Delete
23	Ayam Goreng Nelongso	nasi	Jl. Soekarno Hatta No.37	-7.9465238981351645	112.61765241622925	Edit	Delete
24	Kober Mie Setan	mie	Jl. Soekarno Hatta	-7.948497638109266	112.61694833636284	Edit	Delete
25	McDonald's Watugong	nasi	Jl. MT. Haryono No. 115,	-7.9469170530189555	112.61267423629761	Edit	Delete
26	Deo Bakso Beranak	bakso	Jl. Semanggi Timur No.4 Jatimulyo	-7.947520066872427	112.61947631835938	Edit	Delete
27	Mie Jogging	mie	Jl. Soekarno Hatta No.24	-7.945724304946267	112.61875748634338	Edit	Delete
28	Bakso Pakdhe Remujung	bakso	Jl. Remujung No.26 Jatimulyo	-7.9441583197537256	112.61590629816055	Edit	Delete
29	Warung Sambel Uleg	nasi	Jl. Pisang Kipas Bar	-7.9412348657659875	112.6134467124939	Edit	Delete
30	Warung Prasmanan Andong	nasi	Jl. Bunga Andong No.13	-7.949005017914714	112.6187413930893	Edit	Delete
31	Warung Bu Lasminah	nasi	Jl. Kembang Turi Jatimulyo	-7.942804846823671	112.61451423168182	Edit	Delete
32	Sop Ayam Pak Min Klaten	nasi	Jl. M.T. Haryono No.27 Dinoyo	-7.944635155572071	112.61120975017548	Edit	Delete

Gambar 5.10 Halaman data kuliner

5.2.4 Halaman Edit Data Kuliner

Halaman Admin Sistem ketika ingin mengedit suatu data yang telah dipilih

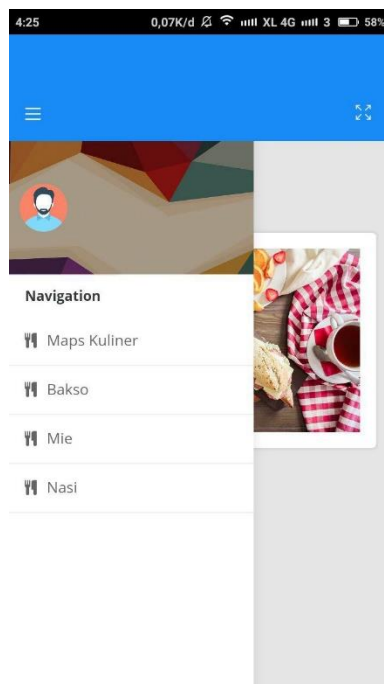


Gambar 5.11 Halaman edit data kuliner

5.3 Implementasi Antar Muka User Kuliner

5.3.1 Halaman Utama

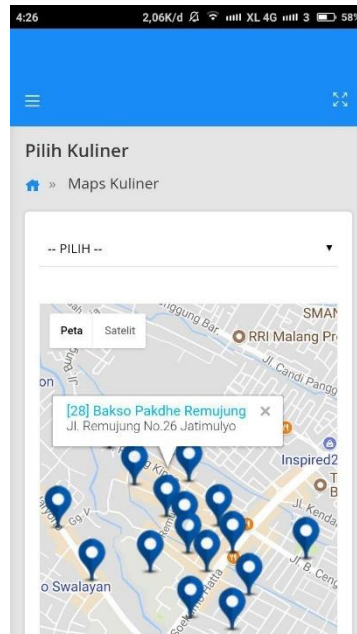
halaman utama akan tampil navigation maps , seperti pada Gambar 5.11



Gambar 5.12 Halaman utama

5.3.2 Halaman Menu Map

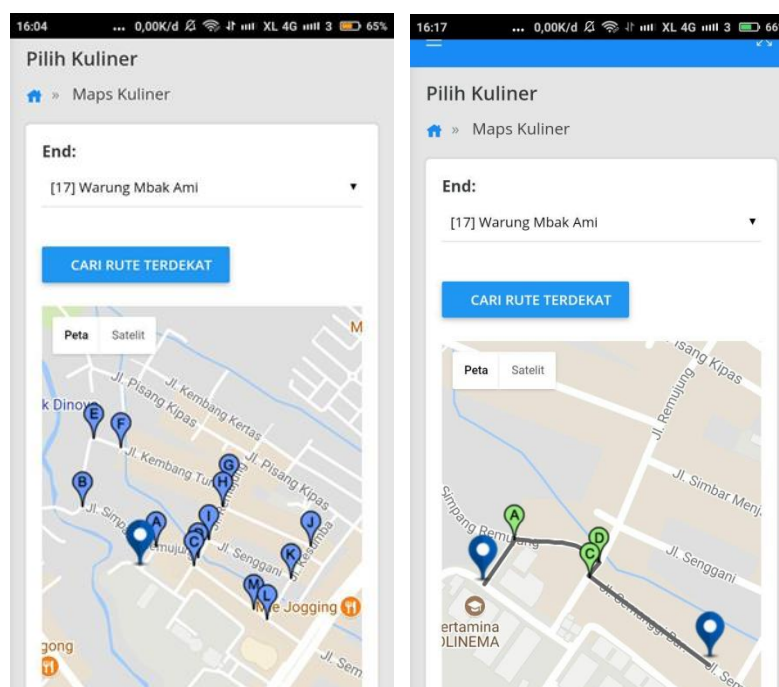
Halaman menu map, berfungsi untuk melihat lokasi kuliner dan mencari lokasi kuliner.



Gambar 5.13 Halaman menu map

5.3.3 Halaman Menu Perhitungan Algoritma Dijkstra

Halaman menu map, berfungsi untuk melihat lokasi kuliner dan mencari lokasi kuliner.



Gambar 5.14 Halaman menu perhitungan algoritma dijkstra

BAB VI. PENGUJIAN DAN PEMBAHASAN

Bab ini berisi hasil-hasil pengujian yang dilakukan terhadap Pengembangan Aplikasi *Mobile* Pencarian rute terpendek menuju lokasi Kuliner Kota Malang menggunakan algoritma Dijkstra.

6.1 Uji Coba

Proses uji coba dilakukan dengan sub sub bab uji coba fungsional dan sub sub bab uji coba akurasi. Uji coba fungsional dilakukan untuk mengetahui apakah sistem yang dibangun sudah menyediakan fungsi - fungsi yang sesuai dengan yang dibutuhkan. Pengujian aplikasi dilakukan dengan cara memverifikasi data yang ditampilkan aplikasi dengan kondisi riil lapangan.

6.1.1 Pengujian Spesifikasi Perangkat Uji Coba

Proses uji coba sistem ini dilakukan pada perangkat *laptop* yang ditunjukkan pada Tabel 6.1 dan Tabel 6.3 dan perangkat *smartphone* dengan spesifikasi yang diuraikan pada Tabel 6.2 dan Tabel 6.4 sebagai berikut:

Tabel 6.1 Spesifikasi perangkat keras PC / Laptop

Perangkat Keras	Keterangan
<i>Processor</i>	Core I7
<i>RAM</i>	4 GB
<i>Harddisk</i>	750 GB
<i>VGA</i>	NVIDIA Geforce 720M

Tabel 6.2 Spesifikasi Perangkat Keras Smartphone

Perangkat Keras	Keterangan
<i>CPU</i>	Snapdragon 425 Quad-core 1.4GHz
<i>Internal</i>	32 GB
<i>RAM</i>	2 GB
<i>Network</i>	3G, HSPA, EDGE, 4G LTE, 4G LTE FDD

Tabel 6.3 Spesifikasi perangkat lunak PC / Laptop

Perangkat Lunak	Keterangan
Windows 10	Sistem Operasi yang digunakan untuk menjalankan program
Android Studio	<i>Software</i> untuk menulis program
Java dan PHP	Bahasa pemograman
MySQL	<i>Software</i> untuk pembuatan <i>database</i> sistem

Tabel 6.4 Spesifikasi Perangkat Lunak Smartphone

Perangkat Lunak	Keterangan
Android OS, 6.0 Marshmallow	Sistem Operasi yang digunakan untuk menjalankan program

6.1.2 Pengujian Performa fungsionalitas Sistem

Pengujian Performa fungsionalitas sistem ini dilakukan dengan cara menjalankan setiap fitur dalam aplikasi dan melihat apakah hasilnya sudah berjalan sesuai dengan yang diharapkan. Hasil pengujiannya dapat dilihat dalam Tabel 6.5.

Tabel 6.5 Fungsionalitas Sistem

No	Pola Pengujian	Hasil Pengujian	Status Pengujian
<i>Website Admin</i>			
1	Proses <i>login</i>	<i>Login</i> berhasil	Sesuai
2	Proses <i>logout</i>	<i>Logout</i> berhasil	Sesuai
3	Kelola data Kuliner		
	a. Menampilkan data kuliner	Data kuliner berhasil ditampilkan	Sesuai
	b. Mengubah data kuliner	Data kuliner berhasil diubah	Sesuai
	c. Menghapus data kuliner	Data kuliner berhasil dihapus	Sesuai
	d. Menampilkan data	Data kuliner berhasil	Sesuai

	kuliner	ditampilkan	
Aplikasi Pengguna Konsumen			
4	Kriteria Pencarian		
	a. Cari berdasarkan pengguna	Tampil <i>marker</i> pada peta	Sesuai
5	Hasil Pencarian		
	a. Menampilkan jalur terpendek	Tampil jalur terpendek	Sesuai
6	Informasi		
	a. Menampilkan informasi kuliner	Tampil alamat kuliner	Sesuai

6.2. Analisis Hasil Uji Coba Sistem

Analisis hasil uji coba yang didapat dari setelah melakukan uji coba sistem yaitu:

- Pada sistem, *Admin* mampu melakukan proses *login* sesuai dengan rancangan yang telah dibuat.
- Pada sistem, *admin* mampu melakukan kelola data kuliner.
- Pada sistem, Pengguna dapat melakukan pencarian berdasarkan menu aplikasi. Lalu setelah memilih Kuliner tujuan proses perhitungan Algoritma Dijkstra berjalan kemudian akan menghasilkan jalur terpendek menuju Kuliner yang dipilih sebelumnya.

6.3. Hasil Uji Coba dengan Responden

Untuk menguji sistem dari segi tampilan, *friendly user*, kehandalan atau keakurasian serta manfaat dari aplikasi yang dikembangkan, telah dilakukan penyebaran kuesioner kepada 10 responden. Responden yang dimaksud adalah para warga Malang dan juga pendatang. Pendapat dari responden tersebut ditampilkan pada Tabel 6.6

Tabel 6.6 Data Hasil Kuesioner

No	Pernyataan	S	R	TS	Jumlah
1	Aplikasi memiliki tampilan menarik	7	3	0	10
2	Aplikasi mudah digunakan	9	1	0	10
3	Proses yang cepat saat mencari rute terpendek	7	3	0	10
4	Aplikasi membantu pengguna menemukan rute terdekat	8	2	0	10

Keterangan :

S = Setuju

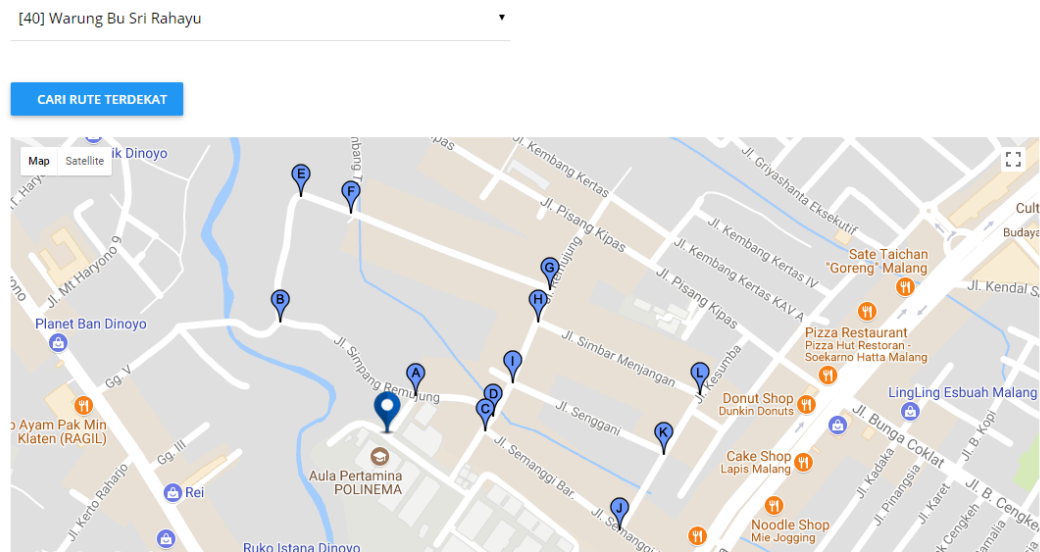
R = Ragu-Ragu

TS = Tidak Setuju

Secara grafik ditunjukkan pada Grafik 6.4.2 Grafik ini menampilkan hasil kuesioner dalam bentuk presentase (%). Nomor 1, 2, 3, 4, 5 pada grafik merupakan nomor dari pertanyaan dalam kuesioner.

6.4. Pengujian Algoritma Dijkstra

Untuk menguji apakah algoritma Dijkstra ini berjalan sesuai dengan hasil yang diharapkan, pengujian algoritma Dijkstra dilakukan dengan melakukan pengecekan secara manual dengan menggunakan perhitungan untuk menentukan apakah rute yang diambil sudah optimal atau tidak.



Gambar 5.15 Pengujian Algoritma Dijkstra

Tabel 6.7 Tabel jarak

Dari		Jarak
A	B	200m
A	D	100m
B	A	200m
B	E	200m
C	D	18m
C	J	200m
D	A	100m
D	C	18m
D	I	46m
E	B	200m
E	F	65m

Tabel 6.8 Tabel jarak 2

Dari		Jarak
F	E	65m
F	G	300m
G	F	300m
G	H	41m
H	G	41m
H	I	79m
H	L	200m
I	D	46m
I	H	79m
I	K	200m
J	C	200m
J	K	100m
K	I	200m
K	J	100m
K	L	85m
L	H	200m
L	K	85m

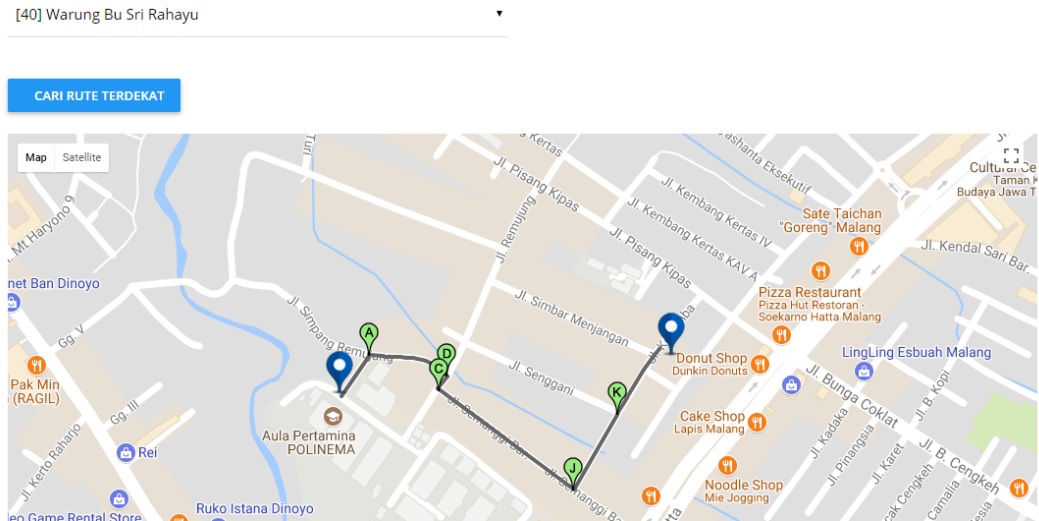
Tabel 6.9 Pengujian algoritma dijkstra

LANGKAH	PILIH	MIN	A	B	C	D	E	F	G	H	I	J	K	L
1	A	100		200		100								
		D												
					100+18						100+46			
2	D	118	100,A		118						146			
		C												
3	C	318			118,D						118+200			
		J										318		
													318+100	
4	J	418			318,C								418	
		K												
						418+200								418+85
5	K	503				618							418,J	503
		L												

Pada gambar diatas untuk mencari rute terpendek dengan cara membandingkan perhitungan tabel diatas, menggunakan rumus minimum formula untuk mencari bobot terkecil atau rute terpendek.

Min (DestValue, MarkedValue+EdgeWeight) :

Berdasarkan hasil dari perhitungan yang telah dilakukan dapat disimpulkan rute terpendek menuju lokasi kuliner dari titik A awal menuju ke titik akhir L yaitu pada rute A-D-C-J-K-L dengan jarak 503meter



Gambar 5.16 Halaman jalur A-D-C-J-K-L

BAB VII. KESIMPULAN

7.1. Kesimpulan

Berdasarkan pengujian yang telah dilakukan dapat diambil beberapa kesimpulan sebagai berikut:

1. Aplikasi dapat menampilkan lokasi kuliner terdekat dan sekitar titik awal kampus Politeknik Negeri Malang.
2. Aplikasi dapat menampilkan informasi seperti tipe kuliner, nama dan alamat lokasi kuliner.
3. Aplikasi dapat menampilkan rute terdekat menuju lokasi kuliner menggunakan perhitungan algoritma dijkstra.

7.2. Saran

Dalam menguji aplikasi ini dapat digunakan metode lain seperti metode genetika dan metode A-star untuk dijadikan perbandingan keakurasian dalam penelitian lebih lanjut.

DAFTAR PUSTAKA

- [1] M. Rizki Kurniawan, Oky Dwi Nurhayati, Kurniawan Teguh Martono., 2015 “Sistem Informasi Geografis pencarian lokasi agen Bus dan Travel terdekat di Kota Semarang berbasis *mobile* dengan metode Dijkstra”. *Jurnal Sistem Komputer, Fakultas Teknik, Universitas Diponegoro Semarang. (e-ISSN: 2338-0403)*
- [2] Pressman. 2005. “Software Engineering, a Practitioner’s Approach”. McGraw-Hill Edisi ke-6
- [3] Roger, Rick, John Lombardo, et al. 2009. Android Application Development. O’Reilly Media: USA.
- [4] Tommy Purwantoro Nugroho., 2016 “Aplikasi Pencarian jalur terpendek untuk menemukan lokasi ATM (Anjungan Tunai Mandiri) terdekat di kota Malang berbasis Android”. *Skripsi Teknologi informasi, Program Studi Teknik Informatika, Politeknik Negeri Malang.*
- [5] Ramadhani, Syaifudin, Anis, Urifatun, Tazkiyatul Masruro, Siti. 2013. “Rancang Bangun Sistem Informasi Geografis Layanan Kesehatan di Kecamatan Lamongan dengan PHP MySQL”. *Jurnal Teknika.*
- [6] Martin Nugroho Parapat., 2016 “Rancang Bangun Aplikasi pencarian rute terpendek jasa kiriman barang berbasis *mobile* dengan metode algoritma Dijkstra”. *Skripsi Teknologi Informasi, Program Studi Teknik Informatika, Politeknik Negeri Malang.*
- [7] Fitriah Apri Triansyah., 2013 “Implementasi Algoritma Dijkstra Dalam Aplikasi Untuk Menentukan Lintasan Terpendek Jalan Darat Antar Kota Di Sumatera Bagian Selatan”. *Jurnal Teknologi Informasi, Program Studi Teknik Informatika, Darma jaya bandar lampung.*
- [8] Naufal Tawang ZA., 2012 “Aplikasi Layanan Pencarian Lokasi Kuliner Terdekat di Yogyakarta berbasis Android”. *Jurnal Jurusan Teknik Informatika, Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM Yogyakarta.*
- [9] Nazruddin Safaat H, 2011, Android “Pemograman Aplikasi Mobile smartphone”. *Program Studi Teknik Informatika, Bandung*
- [10] Akhmad Agus Wijayanto., 2012 “Aplikasi Mobile Lokasi Objek Wisata Kota dan Kabupaten Tegal berbasis Android”. *Jurnal Jurusan Teknik Informatika, Universitas Dian Nuswantoro Semarang.*

LAMPIRAN-LAMPIRAN

Lampiran 1. Data Lokasi Kuliner

No	Nama Kuliner	Alamat	Lattitude	Longitude
1	Warung Mbak Ami	Jl. Semanggi Bar. No.21A	-7.945745556612625	112.61682629585266
2	Ayam Goreng Nelongso	Jl. Soekarno Hatta No.37	-7.9465238981351645	112.61765241622925
3	Kober Mie Setan	Jl. Soekarno Hatta	-7.948497638109266	112.61694833636284
4	McDonald's Watugong	Jl. MT. Haryono No. 115,	-7.9469170530189555	112.61267423629761
5	Deo Bakso Beranak	Jl. Semanggi Timur No.4	-7.947520066872427	112.61947631835938
6	Mie Jogging	Jl. Soekarno Hatta No.24	-7.945724304946267	112.61875748634338
7	Bakso Pakdhe Remujung	Jl. Remujung No.26 Jatimulyo	-7.9441583197537256	112.61590629816055
8	Warung Sambel Uleg	Jl. Pisang Kipas Bar	-7.9412348657659875	112.6134467124939
9	Warung Prasmanan Andong	Jl. Bunga Andong No.13	-7.949005017914714	112.6187413930893
10	Sop Ayam Pak Min Klaten	Jl. M.T. Haryono No.27 Dinoyo	-7.944635155572071	112.61120975017548
11	Warung Bu Lasminah	Jl. Kembang Turi Jatimulyo	-7.942804846823671	112.61451423168182
12	RM Barelang	Jl. Pisang Kipas No.26 Jatimulyo	-7.942666709983261	112.6168692111969
13	Mie Ayam Solo Pak Doel	Jl. semanggi timur Jatimulyo	-7.946919709469586	112.61904716491699
14	Warung Prasmanan Mama	Jl. Senggani No.26 Jatimulyo	-7.944871580925668	112.61668682098389
15	Warung Bu Sri Rahayu	Jl. Kesumba No.15 Jatimulyo	-7.94457405686306	112.61815935373306
16	Bakso Urat Cak Nono	Jl. Gajayana Dinoyo	-7.944852985678071	112.60956823825836

17	Lalapan bu Sripiiiiitt	Jl. Kembang Kertas No.12B Jatimulyo	-7.941505827392268	112.61600285768509
18	Sate Taichan "Goreng"	Jl. Soekarno Hatta No.A8 Jatimulyo	-7.943350752536392	112.62019917368889
19	Bakso Bakar Pak Ahmad	Jl. M.T. Haryono Ketawanggede	- 7.9474961588510595	112.6136264204979
20	Warung Uyun	Jl. Simpang Remujung No.1 Jatimulyo	-7.944730788315773	112.61559784412384

Lampiran 2. Script Metode Algoritma *Dijkstra*

```
<?PHP

ini_set('max_execution_time', 300); //300 seconds = 5 minutes

$id_tujuan = $_POST['id_tujuan'];

$link = mysqli_connect('localhost','root','', 'android_api');

// mysqli_select_db($link, '');

mysqli_query($link, "TRUNCATE TABLE perbandingan_hasil");
mysqli_query($link, "TRUNCATE TABLE perbandingan_hasil_vertex");
$sql = 'SELECT * from rute WHERE id_tujuan = "'.$id_tujuan.'" ';
$data = mysqli_query($link, $sql);

while($row = mysqli_fetch_array($data)){

    $sql_2 = 'SELECT * from rute WHERE id_tujuan = "'.$id_tujuan.'" ';
    $data_2 = mysqli_query($link, $sql_2);

    while($row_2 = mysqli_fetch_array($data_2)){

        $titik_1 = $row['titik']; // titik 1
        $titik_2 = $row_2['titik']; // titik 2

        $lat_1 = trim($row['latitude']);
        $lng_1 = trim($row['longitude']);

        $lat_2 = trim($row_2['latitude']);
        $lng_2 = trim($row_2['longitude']);

        $sql_jarak = "
        SELECT a.* ,
        (6371 * acos(cos(radians('".$lat_2."'))
```

```

        * cos(radians(".$lat_1.")) * cos(radians(".$lng_1."))
        - radians(".$lng_2.")) + sin(radians(".$lat_2."))
        * sin(radians(".$lat_1."))))
    AS JARAK
FROM rute a
WHERE a.titik = ".$titik_1." AND a.id_tujuan = ".$id_tujuan."
";

$jrak = GetDrivingDistance($lat_1, $lat_2, $lng_1, $lng_2);
$jrak = str_replace(',', '.', $jrak);
$jrak = explode(' ', $jrak);

$jarak = $jrak[0];
if($jrak[1] == 'km'){
    $jarak = $jarak*1000;
}
$data_jarak = mysqli_query($link, $sql_jarak);
while($row_jarak = mysqli_fetch_array($data_jarak)){
    mysqli_query($link, "
        INSERT INTO perbandingan_hasil
        (TITIK_1, LAT_1, LONG_1, TITIK_2, LAT_2, LONG_2,
JARAK, ID_TUJUAN)
        VALUES
        (".$titik_1.", ".$lat_1.", ".$lng_1.", ".$titik_2.",
        ".$lat_2.", ".$lng_2.", ".$jarak.", ".$id_tujuan.")
    ");
}
}
}

```

```

        $sql_cek_tujuan = "SELECT * FROM rute WHERE id_tujuan = '". $id_tujuan.'" AND
titik_tujuan = 'tujuan' ";

        $data_cek_tujuan =      mysqli_query($link, $sql_cek_tujuan);
        while($row_cek_tujuan = mysqli_fetch_array($data_cek_tujuan)){

            $tujuan = $row_cek_tujuan['titik'];

        }

        $data_array = array();
        $titik = "A";
        $titik_tujuan = "";
        array_push($data_array, $titik);
        $jml_data = mysqli_num_rows($data);
        for ($i=0; $i < $jml_data; $i++) {

            $sql_simpul = "SELECT * FROM rute WHERE id_tujuan = '". $id_tujuan.'"
AND titik = '". $titik.'" ";

            $data_simpul = mysqli_query($link, $sql_simpul);
            while($row_simpul = mysqli_fetch_array($data_simpul)){

                $simpul_1 = $row_simpul['simpul_1'];
                $simpul_2 = $row_simpul['simpul_2'];

            }

            if($simpul_2 == $tujuan || $simpul_1 == $tujuan){
                break;
            }

            $sql_titik = "
SELECT a.*, b.titik_tujuan FROM perbandingan_hasil a
LEFT JOIN rute b ON a.TITIK_2 = b.titik

```

```

        WHERE a.TITIK_1 = ".$titik." AND a.TITIK_2 != ".$titik." AND
        b.id_tujuan = ".$id_tujuan." AND a.STS IS NULL AND (a.TITIK_2 = ".$simpul_1." OR
        a.TITIK_2 = ".$simpul_2.")

        ORDER BY a.JARAK ASC LIMIT 1

";

$data_titik = mysqli_query($link, $sql_titik);
while($row_titik = mysqli_fetch_array($data_titik)){
    $titik_awal = $row_titik['TITIK_1'];
    $titik = $row_titik['TITIK_2'];
    $titik_tujuan = $row_titik['titik_tujuan'];

    mysqli_query($link, "UPDATE perbandingan_hasil SET STS = 1 WHERE TITIK_2 =
    ".$titik_awal." ");
}

array_push($data_array, $titik);
if($titik_tujuan == 'tujuan'){
    break;
}

}

$sql_vertex = "SELECT * FROM rute WHERE id_tujuan = ".$id_tujuan."";
$data_vertex = mysqli_query($link, $sql_vertex);
while($row_vertex = mysqli_fetch_array($data_vertex)){
    $titik_1 = $row_vertex['titik'];
    $simpul_vertex_1 = $row_vertex['simpul_1'];
    $simpul_vertex_2 = $row_vertex['simpul_2'];
    $simpul_vertex_3 = $row_vertex['simpul_3'];

```

```

        $sql_vertex_1 = "SELECT * FROM perbandingan_hasil WHERE TITIK_1 =
        """.$titik_1.""" AND (TITIK_2 = """.$simpul_vertex_1.""" OR TITIK_2 = """.$simpul_vertex_2."""
        OR TITIK_2 = """.$simpul_vertex_3.""") ";

        $data_vertex_1 = mysqli_query($link, $sql_vertex_1);

        while($row_vertex_1 = mysqli_fetch_array($data_vertex_1)){

            $titik_2 = $row_vertex_1['TITIK_2'];

            $jarak_vertex = $row_vertex_1['JARAK'];

            $sts_vertex = $row_vertex_1['STS'];

            mysqli_query($link, "

                INSERT INTO perbandingan_hasil_vertex

                (TITIK_1, TITIK_2, JARAK, ID_TUJUAN, STS)

                VALUES

                (""".$titik_1.", """.$titik_2.", """.$jarak_vertex.",

                """.$id_tujuan.", """.$sts_vertex.""")

                ");

        }

    }

    $data_hasil = array();

    for ($i=0; $i < count($data_array); $i++) {

        $sql_3 = 'SELECT * from rute WHERE id_tujuan = '.$id_tujuan.' AND
        titik = '.$data_array[$i].'';

        $data_3 = mysqli_query($link, $sql_3);

        while($row_hasil_metode = mysqli_fetch_array($data_3)){

            $row_data = array(

                'titik' => $row_hasil_metode['titik'],

                'latitude' => $row_hasil_metode['latitude'],

                'longitude' => $row_hasil_metode['longitude'],

                'alias' => $row_hasil_metode['alias'],

```

```

        );
        array_push($data_hasil, $row_data);
    }
}

echo json_encode($data_hasil);

function GetDrivingDistance($lat1, $lat2, $long1, $long2){
    $url =
    "https://maps.googleapis.com/maps/api/distancematrix/json?origins=".$lat1.", ".$long1.
    "&destinations=".$lat2.", ".$long2."&mode=driving&language=pl-PL";

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_PROXYPORT, 3128);
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
    $response = curl_exec($ch);
    curl_close($ch);
    $response_a = json_decode($response, true);
    $dist = $response_a['rows'][0]['elements'][0]['distance']['text'];
    $time = $response_a['rows'][0]['elements'][0]['duration']['text'];

    return $dist;
}

?>

```


Lampiran 8. Biodata Penulis



1. DATA PRIBADI

Nama : Michael Eduardo Karambut
Tempat, Tanggal Lahir : Pasuruan 22 Januari 1995
Jenis Kelamin : Laki-laki
Agama : Kristen Protestan
Alamat : Perum Pondok Surya Kencana II blok A no 5
Kota Pasuruan
No. Telp : 081232492402
Email : meduardo295@gmail.com

2. RIWAYAT PENDIDIKAN

- Tahun 2001 s/d 2007 : SD Widya Merti Surabaya
- Tahun 2007 s/d 2010 : SMP Widya Merti Surabaya
- Tahun 2010 s/d 2013 : SMA Negeri 4 Pasuruan
- Tahun 2013 s/d 2017 : D4 Politeknik Negeri Malang
(Teknik Informatika)

