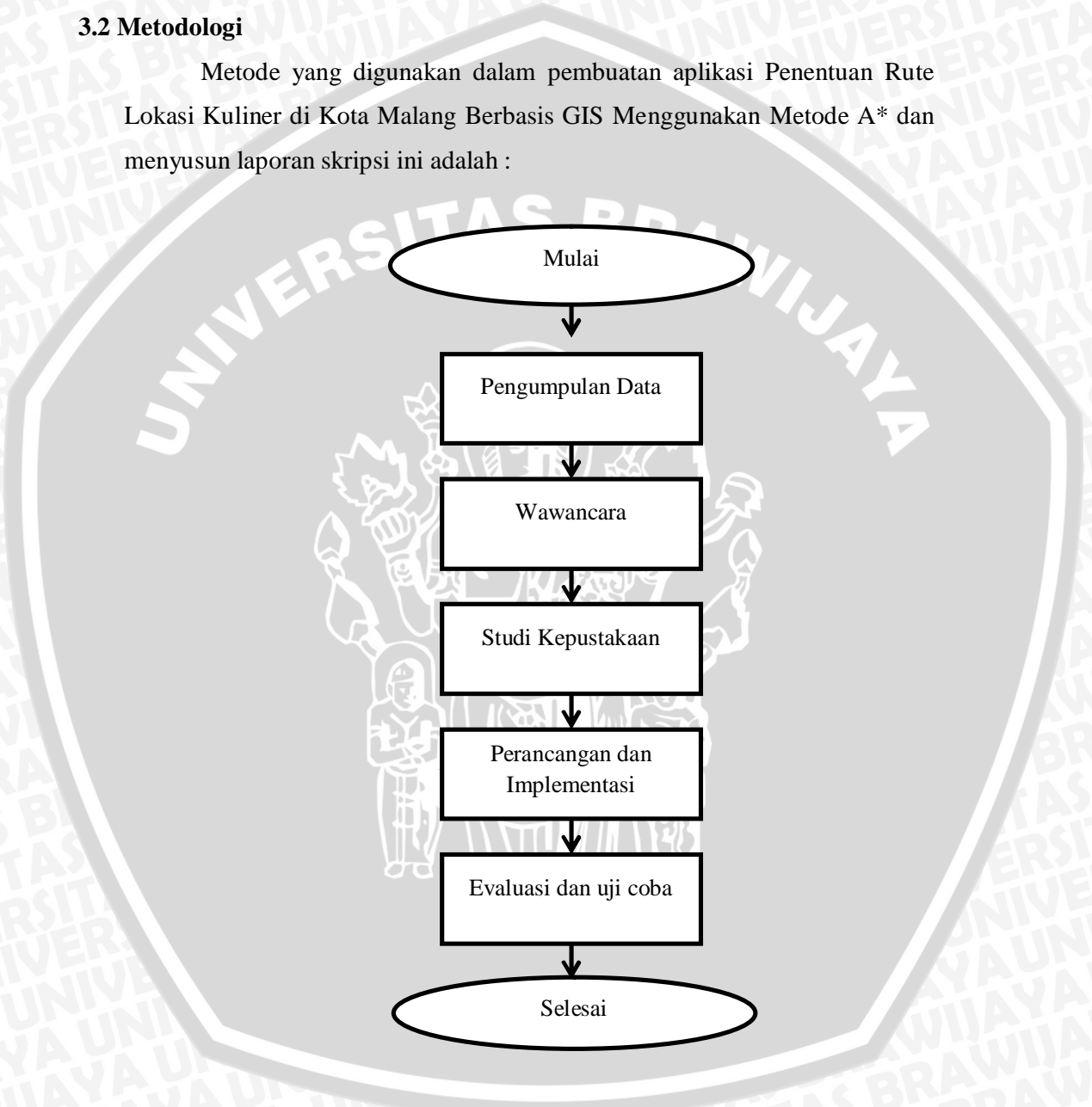


BAB III
METODOLOGI DAN PERANCANGAN SISTEM

3.2 Metodologi

Metode yang digunakan dalam pembuatan aplikasi Penentuan Rute Lokasi Kuliner di Kota Malang Berbasis GIS Menggunakan Metode A* dan menyusun laporan skripsi ini adalah :



Gambar 3.1 Diagram alir metodologi

1. Metode Pengumpulan Data

Mencari dan mengumpulkan data – data yang dibutuhkan dan berkaitan dengan pembuatan aplikasi rute lokasi kuliner di kota Malang.

2. Metode Wawancara

Dilakukan wawancara dengan kantor pariwisata kota malang untuk mendapatkan informasi objek wisata yang ada di Kota Malang.

3. Studi Kepustakaan

Studi kepustakaan seperti mempelajari, dan membaca buku – buku pengembangan aplikasi yang berbasis Android dan perancangan aplikasi mobile GIS pada platform Android untuk membantu dalam pembuatan Aplikasi GIS pada platform Android. Selain itu juga mempelajari web – web referensi di internet tentang pengembangan aplikasi android dan implementasi GIS pada sistem mobile yang berbasis sistem operasi Android.

4. Merancang dan Mengimplementasikan

Menganalisa kebutuhan sistem dan mengidentifikasi kebutuhan informasi berdasarkan hasil pengamatan dan wawancara serta studi pustaka yang telah dilakukan. Alat bantu analisis sistem yang digunakan pada sistem ini adalah use case diagram dan class diagram.

Merancang dan mengimplementasikan “Aplikasi Penentuan Rute Lokasi Kuliner di Kota Malang Berbasis GIS Menggunakan Metode A*” dengan menggunakan bahasa pemrograman Java, Eclipse, SDK Version 8.

5. Evaluasi dan uji coba

Evaluasi dan uji coba yaitu melakukan pengujian terhadap aplikasi yang telah dibuat dengan menggunakan telepon seluler yang berbasis Android.

3.2 Analisis Sistem

Analisis sistem bertujuan untuk mengidentifikasi permasalahan permasalahan yang ada pada sistem. Analisis ini diperlukan sebagai dasar untuk tahapan perancangan sistem. Analisis sistem meliputi identifikasi masalah, spesifikasi aplikasi, spesifikasi pengguna dan lingkungan operasi.

3.2.1 Analisis Permasalahan

Masalah yang dihadapi dalam membangun aplikasi ini adalah sebagai berikut:

- a. Bagaimana membangun GIS berdasarkan data spasial dan data non – spasial yang diperoleh.
- b. Bagaimana cara menentukan titik koordinat lokasi wisata Kuliner secara akurat.
- c. Bagaimana supaya sistem yang akan di bangun dapat dimengerti oleh pengguna, sehingga informasi yang diberikan memberikan hasil yang optimal.

3.2.2 Ruang Lingkup Masalah

Ruang lingkup masalah yang dibahas dalam membangun aplikasi ini yaitu memberikan layanan informasi kepada pengguna yang membutuhkan informasi lokasi dan informasi penting lainnya mengenai tempat wisata Kuliner di kota Malang. Fitur-fitur utama aplikasi ini adalah sebagai berikut:

- a. Pencarian rute terdekat wisata Kuliner.
- b. Menampilkan informasi-informasi penting suatu tempat wisata Kuliner.
- c. Memberikan informasi rute kendaraan menuju objek wisata.
- d. Memberikan jalur alternatif atau jalur terdekat menuju objek wisata.
- e. Pengolahan data wisata Kuliner.

3.2.3 Analisis Data

Data yang digunakan dalam GIS berupa data spasial dan data atribut. Oleh karena itu penulis mengelompokkannya berdasarkan kategori tersebut

3.2.3.1 Data Spasial

3.2.3.1.1 Analis Google Maps API

Google Maps adalah layanan Peta gratis dari perusahaan Google Inc, Google memngizinkan siapapun yang dapat menambahkan fitur Google Maps dalam web mereka sendiri dengan Google Maps API. Dalam hal

kecepatan pencarian data, google sangat diunggulkan oleh semua pemakai internet, maka google maps sangat optimal dan handal dalam hal pencarian data.

Elemen – elemen penting pada Google Map API adalah :

1. Loading Google Maps API

Koneksi script yang kita buat ke server Google Map API dengan menggunakan key yang anda dapatkan pada saat anda mendaftar ke Google Maps API.

2. Map DOM Elements

Untuk menampilkan map pada halaman web, umumnya menggunakan div tag untuk menampungnya. Diharuskan membuat sebuah div tag dengan nama "map_canvas", dan mereferensikan element ini ke Document Object Model (DOM).

3. GMap2 - Elementary Object

Class javascript yang membuat peta itu ada adalah class GMap2, Object dari class ini akan menyediakan sebuah peta di halaman web, Variable map akan berikan nilai sebuah object dari class GMap2. Fungsi GMap2 adalah sebagai constructor dan definisinya.

4. Inisialisasi Map

Inisialisasi di proses dengan menggunakan method setCenter(). Method setCenter() membutuhkan GetLatLng koordinat dan zoom level, dan method ini harus segera dikirim sebelum ada pengoperasian lain pada peta, termasuk seting atribut peta itu sendiri.

5. Loading Map

Ketika halaman HTML di render, document object model (DOM) sudah bisa digunakan, dan semua gambar external dan script diterima oleh object document. Untuk memastikan peta kita di muat sesudah halaman selesai dimuat oleh browser.

6. Fungsi Gunload

adalah utility untuk menangani memory leak.

7. Fungsi Gdirections

Adalah utility yang menangani masalah pencarian rute

8. Gmarker

Marker adalah simbol yang menandakan suatu lokasi wisata pada peta yang ditampilkan Google Maps.

3.2.3.2 Data Atribut/ Non Spasial

Data atribut merupakan data pendukung, dalam hal ini data wisata kuliner, seperti alamat, kode pos, gambar, keterangan dan sebagainya.

3.2.4 Analisa Kebutuhan Sistem

1. Data Input

1. Data tempat wisata kuliner yang akan dicari hasil masukan oleh pengguna.
2. Data posisi koordinat dari hasil masukan (inputan) pengguna guna memperoleh posisi *longitude* dan *latitude* tempat asal.

2. Data Proses

Data proses tempat wisata kuliner di kota Malang, yaitu data posisi (koordinat) setiap tempat Kuliner.

3. Data Output

Data lokasi tempat wisata kuliner terdekat, yaitu solusi hasil akhir pencarian jarak terdekat

3.2.5 Spesifikasi Aplikasi

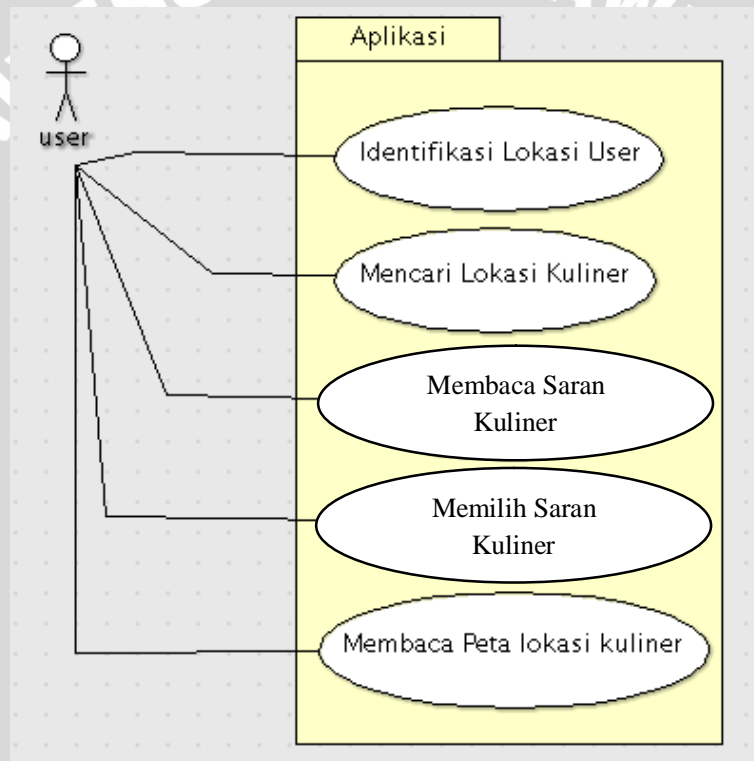
Sistem yang akan dirancang memberikan data keluaran dan memiliki kemampuan sebagai berikut :

1. Memberikan informasi posisi wisata kuliner di kota Malang.
2. Memberikan informasi dan solusi jarak terdekat dari satu titik asal ke lokasi tempat Kuliner.
3. Memiliki kemampuan navigasi peta seperti zoom in, zoom out, scalebar dan lain sebagainya.

3.3 Analisa Dan Perancangan Aplikasi

3.3.1 Use Case Diagram

Untuk mengenal proses dari suatu sistem digunakan *use case* diagram. *Use case* diagram merupakan diagram *high level* yang digunakan untuk menganalisa *requirement & behavior* sebuah sistem dan memodelkan fungsionalitas sistem dan apa yang dibuat oleh aktor. Dengan *use case* diagram ini dapat diketahui proses yang terjadi pada suatu aplikasi. Gambar *use case* dalam aplikasi ini dapat dilihat dibawah ini:



Gambar 3.2 Use Case Diagram Aplikasi

Use case diagram diatas dapat dijelaskan pada tabel dibawah ini :

Tabel 3.1 Definisi Aktor

No	Aktor	Deskripsi
1	User	Merupakan user yang melakukan pencarian lokasi kuliner dan melihat hasil pencarian data berdasarkan keyword yang diinputkan di kolom pencarian aplikasi.
2	Admin	Aktor ini memiliki wewenang yang sama dengan user, akan tetapi dapat menginputkan data yang dibutuhkan oleh user.

Tabel 3.2 Definisi *Use Case*

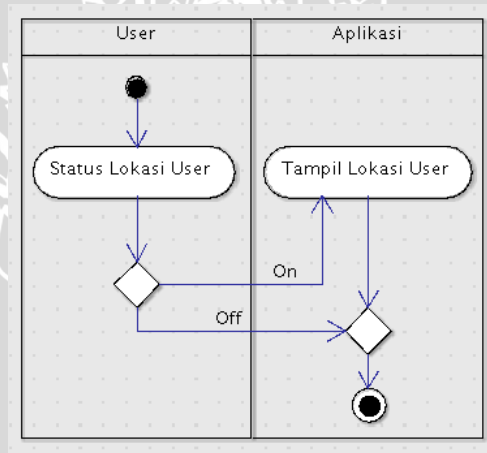
No.	Use Case	Deskripsi
1	Identifikasi Lokasi User	User mengijinkan aplikasi untuk melakukan identifikasi terhadap lokasi dirinya sendiri
2	Mencari Lokasi Kuliner	User dapat melakukan search dengan cara memasukkan keyword kuliner yang dicari kekolom pencarian yang terdapat dalam aplikasi.
3	Membaca Saran Kuliner	User dapat membaca saran kuliner yang ditampilkan aplikasi saat mengetik keyword kuliner. Saran dikeluarkan oleh aplikasi untuk memudahkan pencarian kuliner yang diperlukan.
4	Memilih Saran Kuliner	User dapat memilih dan mengakses saran yang di keluarkan oleh aplikasi untuk memudahkan pencarian kuliner.
5	Membaca Peta Lokasi Kuliner	User dapat membaca hasil pencarian berupa peta yang telah dicari berdasarkan keyword yang telah dimasukkan dalam kolom pencarian pada aplikasi.

3.3.2 Activity Diagram

Activity diagram Menggambarkan bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram digunakan untuk mengilustrasikan proses-proses parallel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis *horizontall* atau *vertical*. Berikut ini merupakan Activity diagram yang terdapat dalam aplikasi pencarian jarak terdekat rute lokasi kuliner di kota Malang.

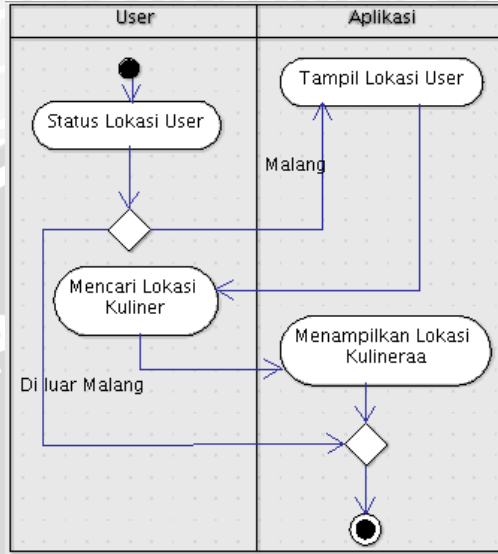
Activity diagram ini menunjukkan sebuah alur yang dilakukan oleh sistem terhadap pemberlakuan *user*. Dimana didalamnya terdapat *input* data pencarian dan mendapatkan info lokasi.

1. Identifikasi lokasi user



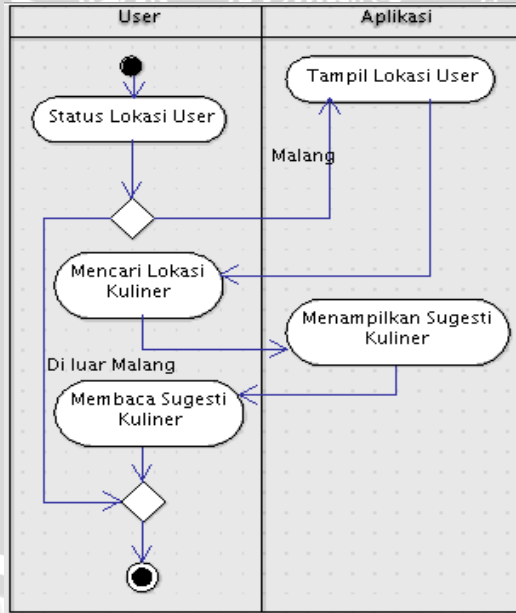
Gambar 3.3 Activity Diagram Identifikasi lokasi user

2. Mencari lokasi kuliner



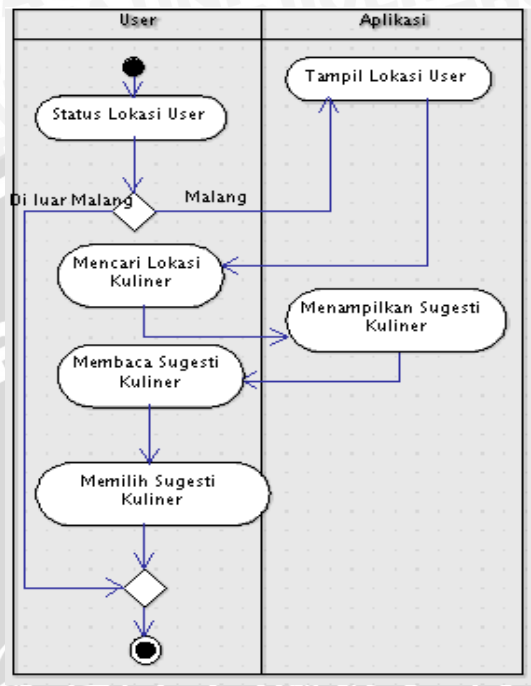
Gambar 3.4 Activity Diagram Mencari lokasi kuliner

3. Membaca saran kuliner



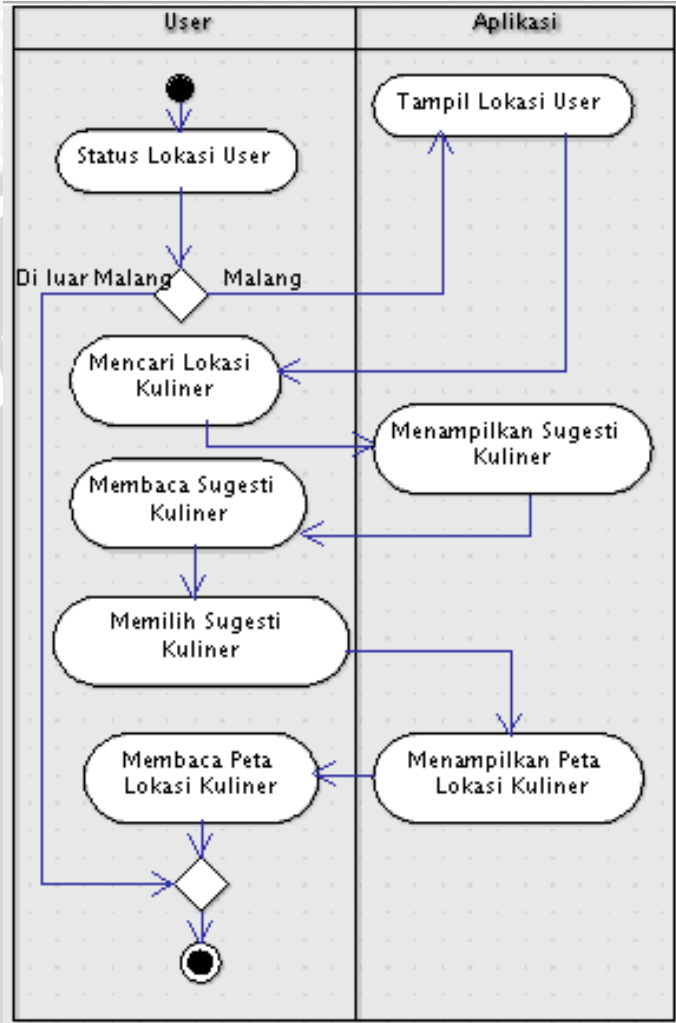
Gambar 3.5 Activity Diagram Membaca saran kuliner

4. Memilih saran kuliner



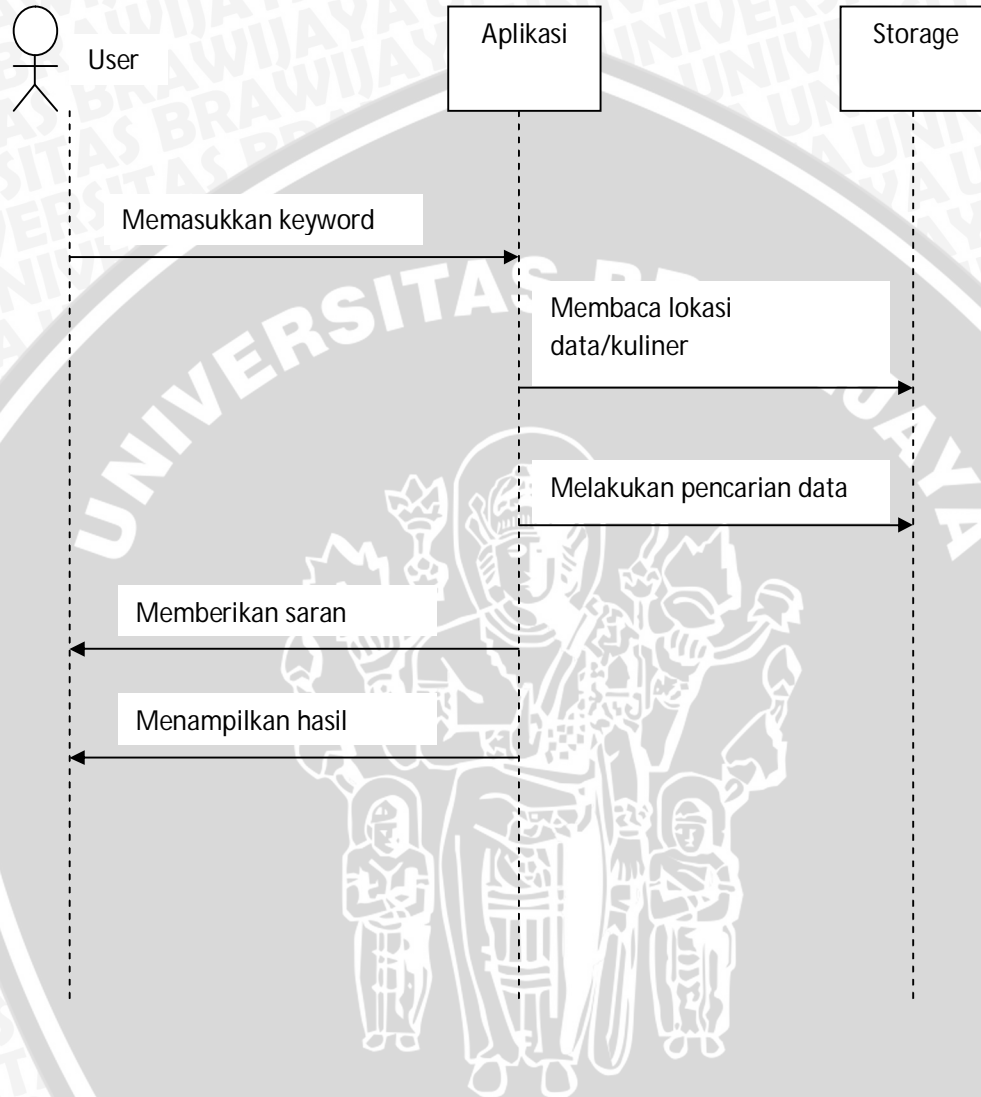
Gambar 3.6 Activity Diagram Memilih saran kuliner

5. Membaca peta lokasi kuliner



Gambar 3.7 Activity Diagram Membaca peta lokasi kuliner

3.3.3 Sequence Diagram



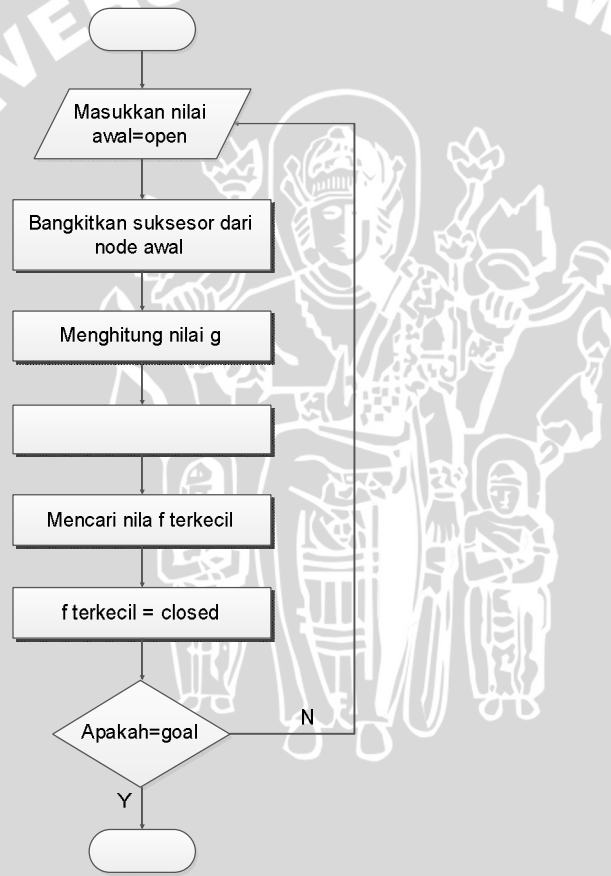
3.4 Contoh Kasus Pencarian Rute Lokasi kuliner Terdekat

3.4.1 Alur Algoritma A*

Dalam perancangan aplikasi pencarian rute lokasi kuliner terdekat di kota Malang ini menggunakan algoritma A*. Karena algoritma A* menerapkan teknik heuristic dalam membantu penyelesaian persoalan. Heuristik adalah penilai yang memberi harga pada tiap simpul yang

memandu A* mendapatkan solusi yang diinginkan. Dengan heuristik yang benar, maka A* pasti akan mendapatkan solusi (jika memang ada solusinya) yang dicari.

Cara kerja aplikasi pencarian rute lokasi kuliner terdekat di kota Malang menggunakan algoritma A* akan di gambarkan dalam bentuk *flowchart*. *Flowchart* merupakan alur proses dari sistem mulai dari *start* sampai dengan *finish*. Adapun gambaran carakerja aplikasi ini adalah sebagai berikut :



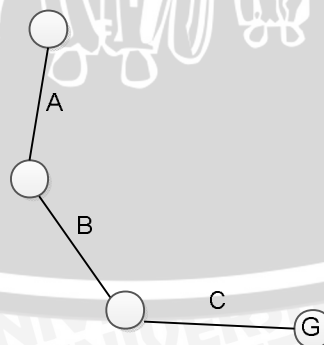
Gambar 3.8 Flowchart Penerapan Algoritma A*

Keterangan *flowchart* penerapan algoritma A.* adalah sebagai berikut :

1. Menentukan posisi awal dan masukkan ke dalam open.
2. Bangkitkan suksesor dari node awal untuk mendapatkan nilai g.
3. Dilakukan perhitungan nilai g.
4. Mencari nilai heuristik dari *current node* menuju *node goal*.
5. Dilakukan pencarian nilai f terkecil.
6. Kemudian nilai f terkecil dimasukkan ke dalam *closed*.
7. Dilakukan pengecekan apakah sudah menuju *node goal*. Apabila belum, maka dilakukan perulangan dari tahap awal sampai didapatkan *node goal*.

3.4.2 Fungsi Heuristik

Pada proses penerapan algoritma A* menggunakan heuristic manhattan *distance* yang mana perhitungannya dengan menjumlahkan setiap rute dari *current node* menuju *goal state*. Perhitungannya dapat di ilustrasikan pada graf dibawah ini.



Gambar 3.9 Graf Heuristik

Pada gambar 3.2 dapat digambarkan dimana *node S* sebagai *current node* dan *node G* sebagai *goal state*. Dari gambar diatas ditentukan bahwa untuk mendapatkan nilai heuristik di dapat dari penjumlahan nilai A, B dan C. Berikut perumusannya :

$$H(n) = A + B + C$$

3.4.3 Jarak

Untuk penentuan jarak pada aplikasi ini menggunakan nilai latitude dan longitude yang diperoleh dari google maps. Perhitungannya dapat dirumuskan sebagai berikut.

$$\text{Jarak} = \sqrt{(\text{Lat1} - \text{Lat2})^2 + (\text{Long1} - \text{Long2})^2}$$

Keterangan :

Lat1 = Nilai latitude titik awal.

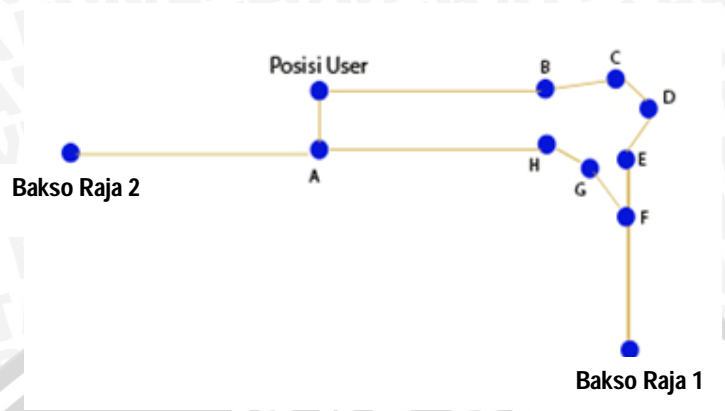
Lat2 = Nilai latitude titik tujuan.

Long1 = Nilai longitude titik awal.

Long2 = Nilai longitude titik tujuan.

3.5 Implementasi pada Node Bakso Raja

Pada implementasi kali ini dimana posisi user berada pada posisi - 7.938398, 112.625414 di jalan Sukarno Hatta, dalam pencarian lokasi posisi Bakso Raja



Gambar 3.10 Posisi User dan Titik Yang Dilalui

Keterangan :

Posisi User = Pertigaan Sukarno Hatta.

BaksoRaja 1 = Jalan Sukarno Hatta.

BaksoRaja 2 = Jalan Sukarno Hatta (Pom Bensin).

A = Pertigaan Sukarno Hatta

B = Pertigaan Sukarno Hatta (Bundaran Pesaawat)

C = Belokan Sukarno Hatta (Bundaran Pesaawat)

D = Pertigaan Bundaran Pesaawat (Arah Sudimoro)

E = Belokan Sukarno Hatta (Bundaran Pesaawat)

F = Pertigaan Sukarno Hatta (Bundaran Pesaawat)

G = Belokan Sukarno Hatta (Bundaran Pesaawat)

H = Pertigaan Sukarno Hatta (Bundaran Pesaawat)

Untuk keterangan letak posisi Bakso Raja dan titik – titik lintasan yang dilalui akan dijelaskan pada tabel dibawah ini.

Tabel 3.3 Posisi Kuliner

N a m a	Latitude	Longitude	Jalan
Bakso Raja – 1	-7.937595	112.627544	Sukarno Hatta
Bakso Raja – 2	-7.939143	112.625201	Sukarno Hatta

Tabel 3.4 Posisi Titik Lintasan

Titik	Latitude	Longitude	Jalan
A	-7.938456	112.625532	Sukarno Hatta
B	-7.936974	112.626181	Sukarno Hatta
C	-7.936814	112.626229	Pertigaan-pesawat
D	-7.936804	112.626438	Pertigaan-pesawat
E	-7.93691	112.626503	Pertigaan-pesawat
F	-7.937133	112.626857	Sukarno Hatta
G	-7.937064	112.626455	Pertigaan-pesawat
H	-7.937107	112.626261	Sukarno Hatta

Metode penyelesaian :

Langkah 1 :

Tentukan posisi awal ke dalam *open list*. Posisi *user* ditetapkan sebagai posisi awal, kemudian letakkan pada *closed list*. Membangkitkan suksesor dari posisi *user*, yaitu *node A* dan *B*. Kedua *node* tersebut ditempatkan pada *open list*. Berikut perhitungan selanjutnya yang nantinya akan di tempatkan pada *closed list*.

Tabel 3.5 Perhitungan Nilai G

G n	(x ₁ - x ₂)	(y ₁ - y ₂)	$\sqrt{(x_1-x_2)^2 + (y_1-y_2)^2}$
Gn(A)	0,000000003	0,000000013	0,000131484
Gn(B)	0,000002027	0,000001857	0,001971179

Untuk perhitungan nilai heuristik yang melalui rute node B menuju *goal state* Bakso Raja 1 dijelaskan pada tabel dibawah ini.

Tabel 3.6 Perhitungan Nilai H(B)

<i>Node</i>	$abs(x_1 - x_2)$	$abs(y_1 - y_2)$	$H(n)$
Hn(C)	0,00016	0,000607	0,000939
Hn(D)	0,00001	0,000316	0,000940
Hn(E)	0,000106	0,000110	0,000168
Hn(F)	0,000223	0,000282	0,000952
Hn(Bakso Raja 1)	0,000462	0,000687	0,00115
Total = Hn(C) + Hn(D) + Hn(E) + Hn(F) + Hn(BaksoRaja 1)			0,00232

Untuk perhitungan nilai heuristik yang melalui rute node A menuju *goal state* Bakso Raja 1 dijelaskan pada tabel dibawah ini.

Tabel 3.7 Perhitungan Nilai H(A)

<i>Node</i>	$abs(x_1 - x_2)$	$abs(y_1 - y_2)$	$H(n)$
Hn(H)	0,001349	0,000729	0,002078

Tabel 3.8 Lanjutan

<i>N o d e</i>	a b s (x ₁ - x ₂)	a b s (y ₁ - y ₂)	H (n)
Hn (G)	0,000004	0,000194	0,000237
Hn (F)	0,000007	0,000402	0,000471
Hn(BaksoRaja1)	0,000462	0,000687	0,001149
Total = Hn (H) + Hn (G) + Hn (F) + Hn (Bakso Raja 1)			0,003935

Dari tabel diatas maka dapat dilakukan perhitungan :

$$\begin{aligned} F(A) &= G(A) + H(A) \\ &= 0,000131484 + 0,003935 = 0,004066484 \end{aligned}$$

$$\begin{aligned} F(B) &= G(B) + H(B) \\ &= 0,001971179 + 0,00232 = 0,00430 \end{aligned}$$

Dapat disimpulkan bahwa titik A yang menjadi prioritas, jadi node A diletakkan pada *closed list*.

Langkah ke 2 :

Posisi awal ada pada *node* A. Selanjutnya membangkitkan suksesor dari posisi awal yaitu *node* Bakso Raja 2 dan H. Kedua *node* ditempatkan pada *open list*. Berikut perhitungannya :

Tabel 3.9 Perhitungan Nilai G

G n	(x ₁ - x ₂)	(y ₁ - y ₂) ²	$\sqrt{(x_1-x_2)^2 + (y_1-y_2)^2}$
Gn(Bakso Raja 2)	0,00000047	0,00000011	0,000763
Gn(H)	0,0000018	0,00000053	0,001533

Untuk perhitungan nilai heuristik yang melalui rute node H menuju *goal state* Bakso Raja 1 dijelaskan pada tabel dibawah ini.

Tabel 3.10 Perhitungan Nilai H(H)

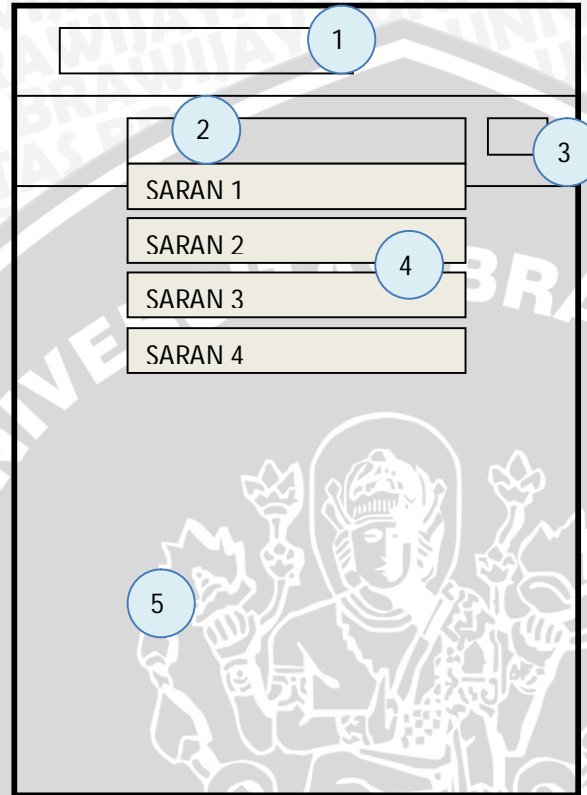
<i>N o d e</i>	$abs(x_1-x_2)$	$abs(y_1-y_2)$	H (n)
Hn (G)	0,000043	0,000194	0,000237
Hn (F)	0,000069	0,000402	0,000471
Hn(Bakso Raja 1)	0,000462	0,000687	0,001149
Total = Hn (G) + Hn (F) + Hn(Bakso Raja 1)			0,001857

Pada langkah ke 2 yang menjadi prioritas utama yaitu pada *node* BaksoRaja 2, sehingga diletakkan pada *closed list*. Dikarenakan *node* tujuan telah tercapai, maka perhitungan diberhentikan dengan mengabaikan *node* lain yang masih mempunyai *node - node* lagi setelahnya. Diperhitungan ini disarankan *user* untuk memilih Bakso Raja 2.



3.6 Rancangan User Interface

Desain *interface* pada sistem ini adalah sebagai berikut :



Gambar 3.11 Desain Interface

Desain dasar untuk halaman Aplikasi Penentuan Rute Lokasi Kuliner di Kota Malang Berbasis GIS Menggunakan Metode A* terdiri dari beberapa bagian sebagai berikut:

1. *Header*, berisi nama sistem
2. Kolom Pencarian, merupakan form untuk mengisi keyword kuliner yang akan dicari.
3. Tombol *Search*, merupakan tombol yang digunakan untuk mencari keyword kuliner setelah kolom pencarian diisi.
4. Saran, berisi beberapa saran yang ditawarkan aplikasi saat menginputkan keyword kuliner.
5. *Map*, berisi peta lokasi user dan lokasi kuliner yang dicari.