

BAB III

ANALISIS DAN DESAIN SISTEM

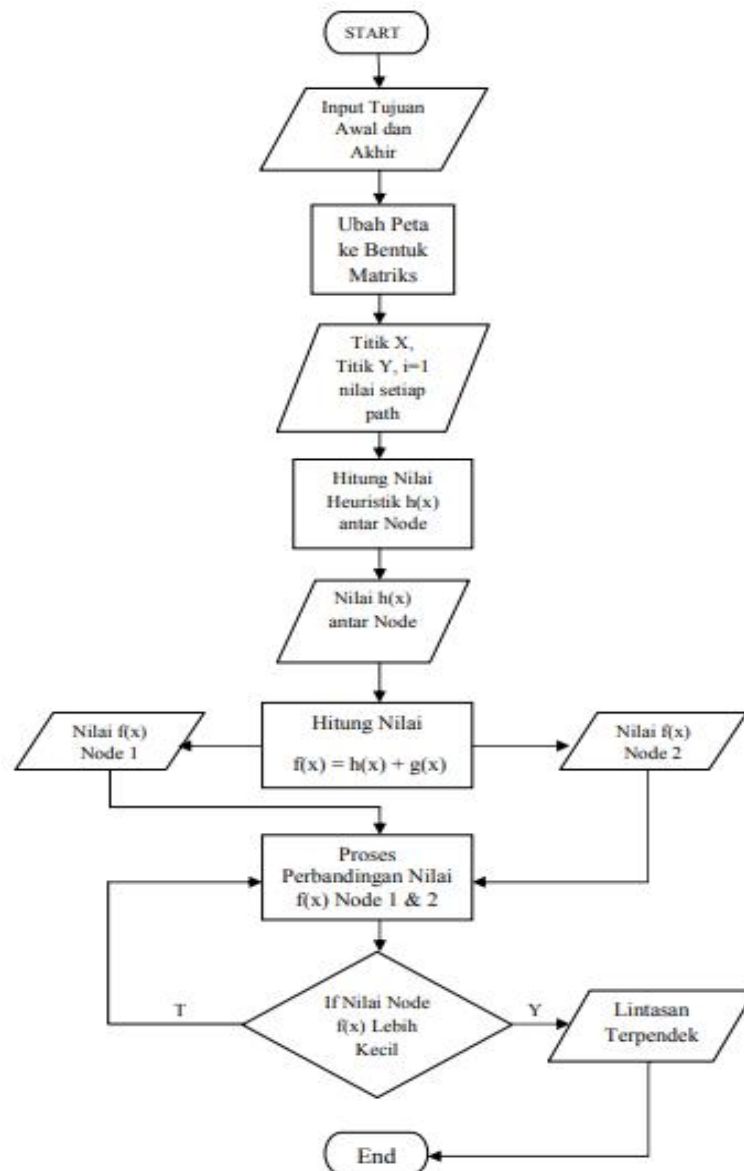
III.1. Analisis Masalah

Permasalahan yang ditemui oleh penulis yaitu dalam proses pencarian rute ke rumah sakit saat ini masih manual membutuhkan banyak waktu untuk proses pencarian rute yang pendek dan ditambah dengan masalah seperti rumah sakit yang sulit dijangkau oleh sinyal atau koneksi internet yang menghambat proses pencarian yang menyebabkan kurang efisien dan akurat.

III.2. Penerapan Algoritma A*

III.2.1. Studi Kasus

Penulis melakukan studi kasus dengan menerapkan Algoritma A* untuk Pencarian Rute Terdekat Rumah Sakit Menggunakan Metode Algoritma A –Star Berbasis *Android*. Adapun langkah-langkah penelitian pada studi kasus ini dapat di lihat pada gambar flowchart III.1.



Gambar III.1. Flowchart Langkah-Langkah Penelitian Algoritma A*

Penulis mengambil studi kasus jarak antara Lokasi A ke Lokasi B. Lokasi A yaitu lokasi titik awal yang merupakan rumah pengguna menuju Lokasi B yaitu Rumah Sakit Bhayangkara. Berikut adalah gambaran Lokasi A ke Lokasi B yang telah di proyeksikan ke dalam bentuk peta yang menghasilkan 6 *node*, dimana pengambilan *nodenya* berdasarkan

Keterangan gambar peta dalam bentuk matriks dapat dilihat pada tabel sebagai berikut :

Tabel III.1. Titik dan Koordinat Peta

TITIK	KOORDINAT
A	(0,0)
B	(40,60)
C	(20,50)
D	(30,50)
E	(40,50)
F	(30,60)

III.2.2. Menghitung Fungsi Heuristik

Fungsi Heuristik yang akan digunakan penulis adalah “*Euclidean Distance*” yaitu perhitungan jarak dari 2 buah titik dalam *Euclidean Space*, alasan penulis menggunakan fungsi heuristik ini karena fungsi ini memberikan hasil yang lebih baik (mendekati jarak sebenarnya) dibandingkan dengan fungsi heuristik yang lain, rumus *Euclidean Distance* yaitu :

$$d(x + y) = |\sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

Perhitungan dari semua titik dapat dilihat sebagai berikut :

$$\mathbf{A \ ke \ B = (0,0), (40,60) = 7211,10}$$

$$d = \sqrt{(40-0)^2} + \sqrt{(60-0)^2}$$

$$d = \sqrt{(40)^2} + \sqrt{(60)^2}$$

$$d = \sqrt{1600} + \sqrt{3600}$$

$$d = \sqrt{5200}$$

$$d = 7211,10$$

$$\mathbf{A \text{ ke } C = (0,0), (20,50) = 5385,16}$$

$$d = \sqrt{(20-0)^2} + \sqrt{(50-0)^2}$$

$$d = \sqrt{(20)^2} + \sqrt{(50)^2}$$

$$d = \sqrt{400} + \sqrt{2500}$$

$$d = \sqrt{2900}$$

$$d = 5385,16$$

$$\mathbf{A \text{ ke } D = (0,0), (30,50) = 5830,95}$$

$$d = \sqrt{(30-0)^2} + \sqrt{(50-0)^2}$$

$$d = \sqrt{(30)^2} + \sqrt{(50)^2}$$

$$d = \sqrt{900} + \sqrt{2500}$$

$$d = \sqrt{3400}$$

$$d = 5030,95$$

$$\mathbf{A \text{ ke } E = (0,0), (40,50) = 6403,12}$$

$$d = \sqrt{(40-0)^2} + \sqrt{(50-0)^2}$$

$$d = \sqrt{(40)^2} + \sqrt{(50)^2}$$

$$d = \sqrt{1600} + \sqrt{2500}$$

$$d = \sqrt{4100}$$

$$d = 6403,12$$

$$A \text{ ke } F = (0,0), (30,60) = 6708,20$$

$$d = \sqrt{(30-0)^2} + \sqrt{(60-0)^2}$$

$$d = \sqrt{(30)^2} + \sqrt{(60)^2}$$

$$d = \sqrt{900} + \sqrt{3600}$$

$$d = \sqrt{4500}$$

$$d = 6708,20$$

III.2.3. Langkah-langkah pencarian dalam Algoritma A-Star

Setelah nilai heuristik dari masing-masing *node* didapat maka kita akan mencari $f(n)$ menggunakan algoritma A-Star dengan rumus :

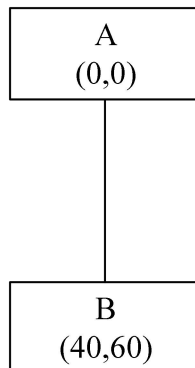
$$f(n) = h(n) + g(n)$$

$h(n)$ = Nilai heuristik antar Koordinat.

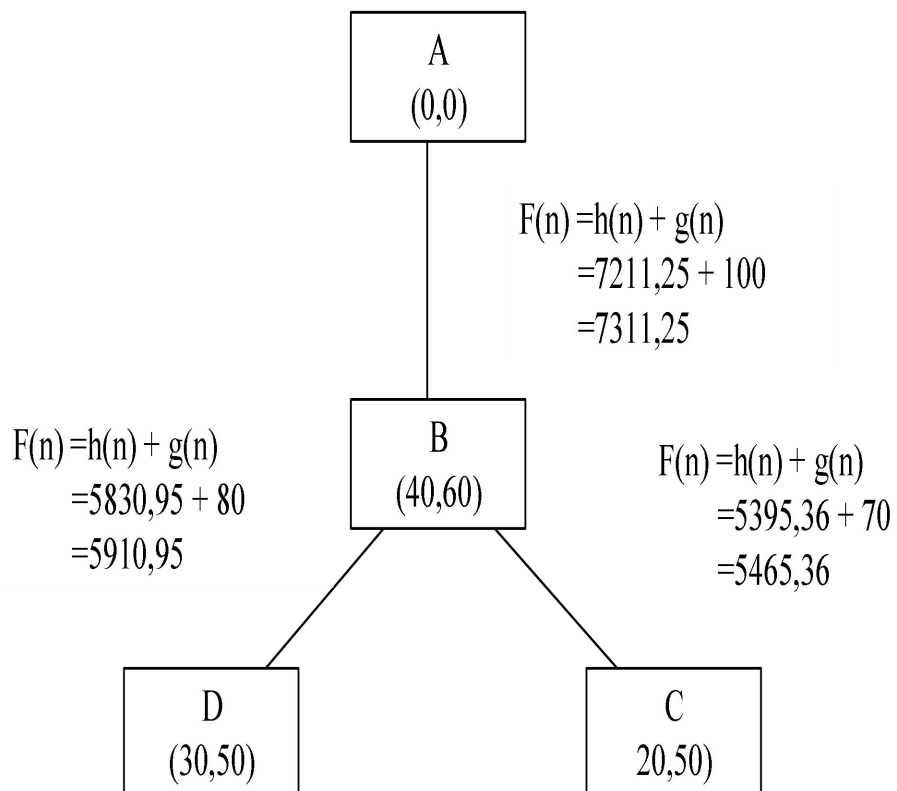
$g(n)$ = Jarak Koordinat ke titik tujuan.

Langkah 1

A (0,0)

Langkah 2

$$\begin{aligned}
 F(n) &= h(n) + g(n) \\
 &= 7211,25 + 100 \\
 &= 7311,25
 \end{aligned}$$

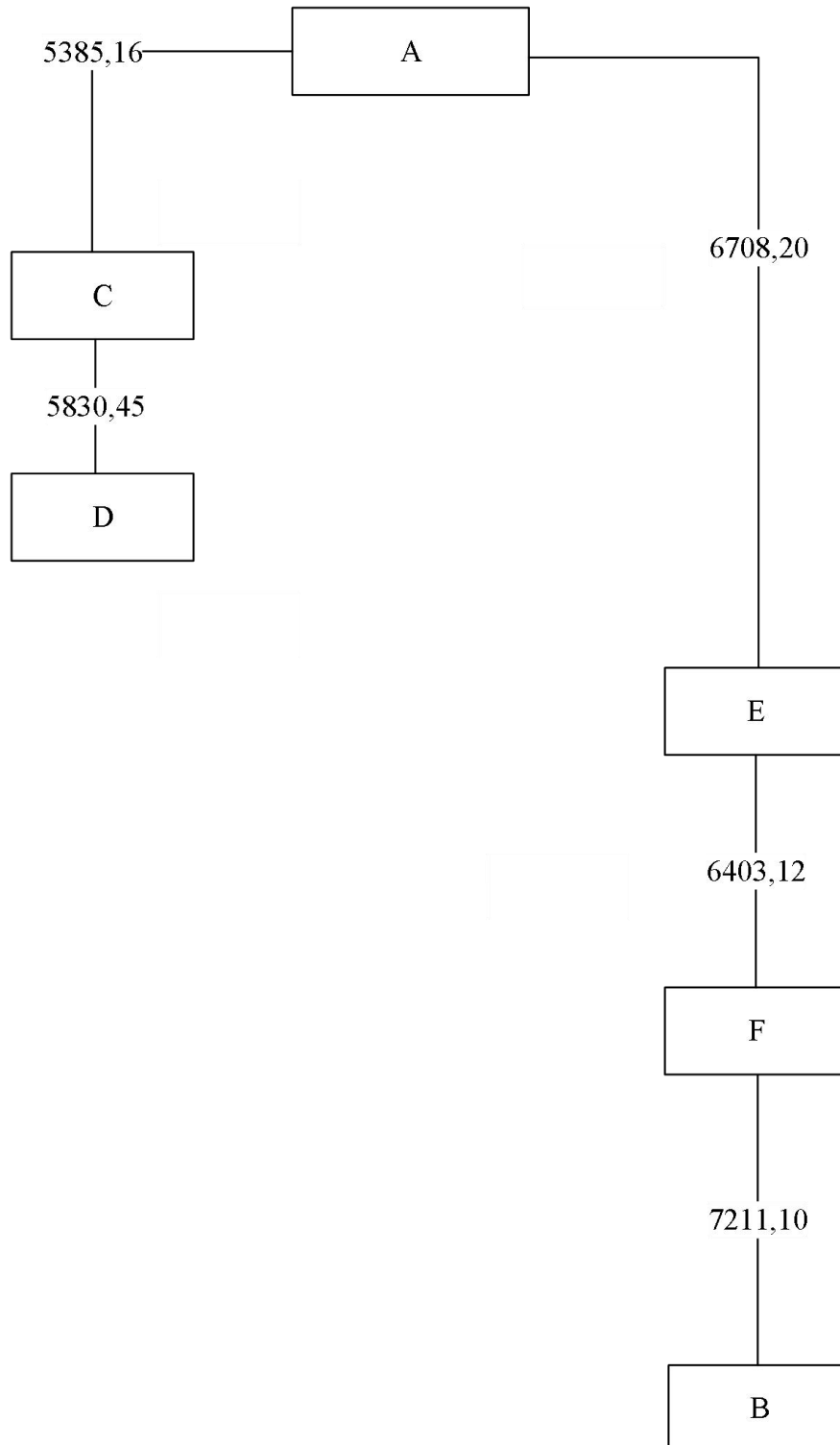
Langkah 3

$$\begin{aligned}
 F(n) &= h(n) + g(n) \\
 &= 7211,25 + 100 \\
 &= 7311,25
 \end{aligned}$$

$$\begin{aligned}
 F(n) &= h(n) + g(n) \\
 &= 5830,95 + 80 \\
 &= 5910,95
 \end{aligned}$$

$$\begin{aligned}
 F(n) &= h(n) + g(n) \\
 &= 5395,36 + 70 \\
 &= 5465,36
 \end{aligned}$$

Titik B memiliki 2 percabangan yaitu titik C dan titik D, maka $f(n)$ yang harus dipilih adalah $f(n)$ yang menghasilkan *cost* paling kecil, yaitu titik C.



Titik A memiliki 2 percabangan yaitu titik C dan titik E, maka $f(n)$ yang harus dipilih adalah $f(n)$ yang menghasilkan *cost* paling kecil dan menjadi titik terakhir yaitu titik D.

Maka $f(n)$ total yang didapat adalah 5830,45, karena satu indeks kordinat mewakili 100 meter maka jaraknya sebenarnya (dalam meter) adalah : $5830,45 \times 100 = 583045$ meter , dalam Kilometer = 5,83045 km, jalur yang dilalui : A - C - D.

Jadi Kesimpulan pada algoritma A-Star adalah untuk mempercepat atau memperpendek jarak untuk dalam pencarian rute terpendek, misalnya, pencarian pada kasus ini saya membuat aplikasi pencarian rumah sakit terdekat, yang ada di Kota Medan.

Masalah tersebut dapat berupa apa saja, dengan syarat untuk setiap permasalahan memiliki kriteria kondisi awal yang harus dipenuhi sebelum menjalankan sebuah algoritma. Algoritma juga memiliki pengulangan proses (iterasi), dan juga memiliki keputusan hingga keputusan selesai.

Prinsip dari algoritma A-Star adalah melakukan traversal satu per satu pada tiap simpul untuk memperoleh lintasan terpendek. Algoritma ini akan menghitung jarak salah satu lintasan, lalu menyimpannya dan kemudian menghitung jarak lintasan lainnya. Ketika seluruh lintasan telah selesai dihitung, algoritma A-Star akan memilih lintasan yang paling pendek.

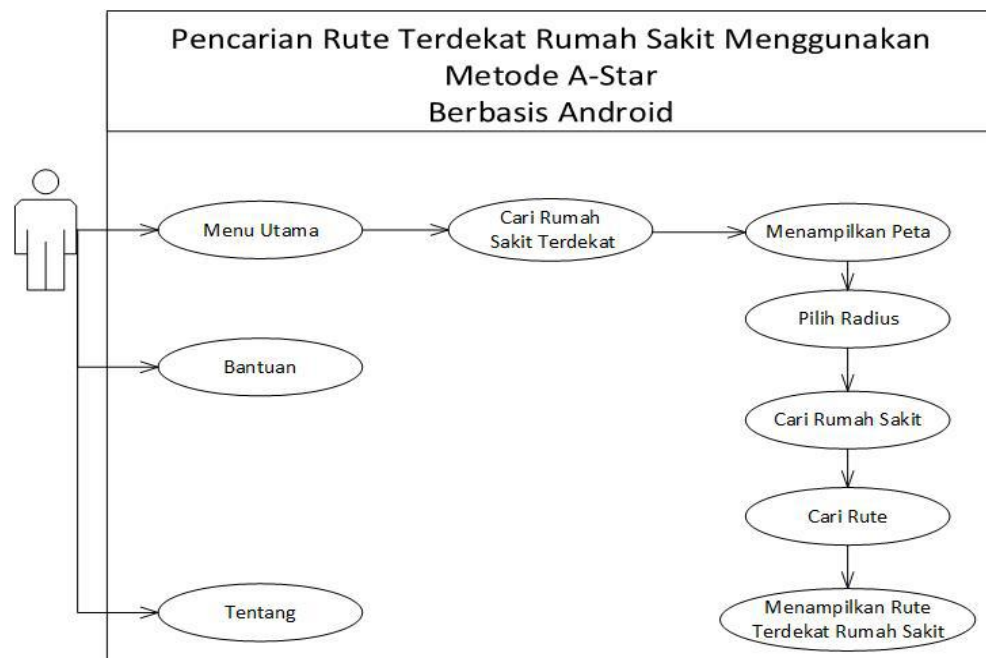
III.3. Desain Sistem

Adapun bentuk rancangan dalam sistem yang penulis akan buat menggunakan beberapa bentuk diagram pemodelan UML (*Unified*

Modeling Language) yang terdiri dari *Use Case Diagram*, *Activity Diagram* & *Sequence Diagram*.

III.3.1. Use Case Diagram

Use Case Diagram menjelaskan urutan kegiatan yang dilakukan *actor* dan *system* untuk mencapai tujuan tertentu. Sebuah *Use Case* mempresentasikan sebuah interaksi antara *actor* dengan *system* dan menggambarkan fungsionalitas yang diharapkan dari sebuah *system*. Berikut ini adalah *Use Case Diagram* untuk Pencarian Rute Terdekat Rumah Sakit Menggunakan Metode Algoritma A-Star Berbasis *Android* dapat dilihat pada gambar III.4



Gambar III.4. Use Case Diagram Pencarian Rute Terdekat Rumah Sakit A-Star

III.3.2. Activity Diagram

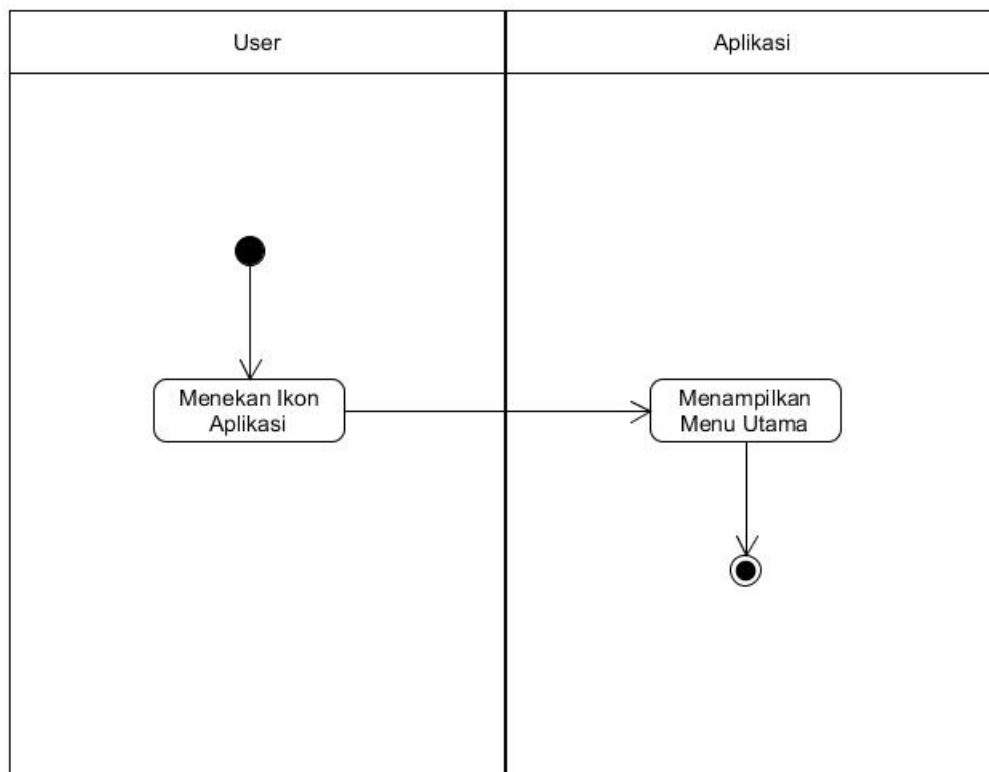
Activity Diagram merupakan proses sistem dalam komponen-komponen beserta seluruh penghubung antar suatu alur data

digambarkan dengan anak panah, yang menunjukkan arah menuju ke dalam dan ke luar dari suatu proses *Activity Diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.

Adapun pemecahan dari *Activity Diagram* berdasarkan *Usecase Diagram* mengenai Pencarian Rute Terdekat Rumah Sakit Menggunakan Metode Algoritma A-Star Berbasis *Android* dapat dilihat pada gambar III.2.

III.3.2.1. *Activity Diagram* “Menu Utama”

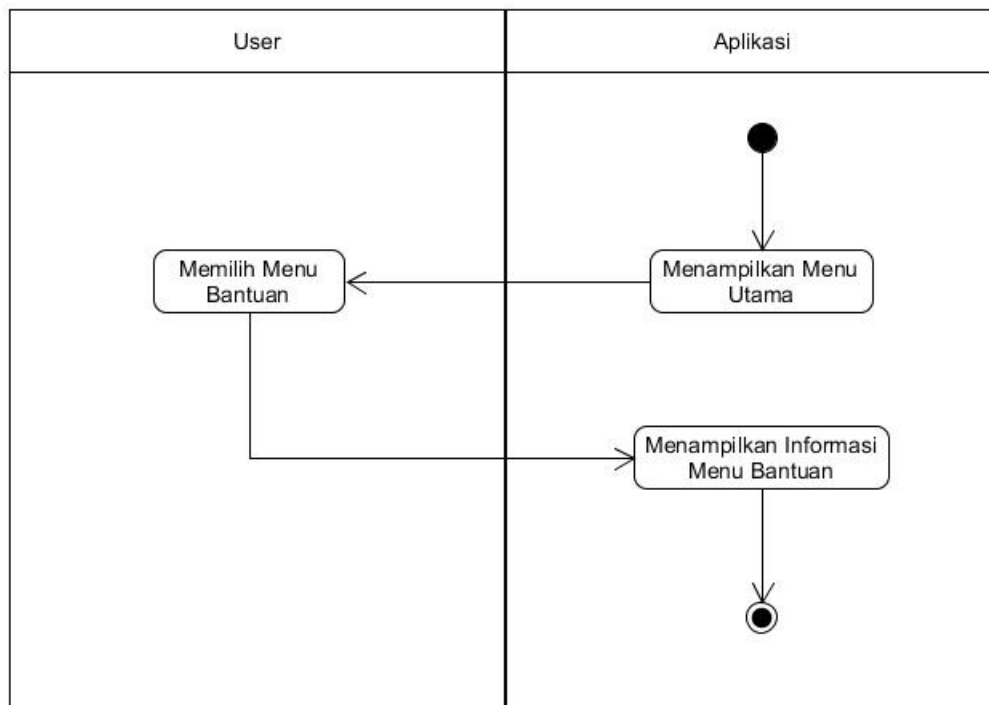
Menggambarkan aktivitas ketika pengguna pertama kali membuka aplikasi, dapat dilihat pada gambar III.5.



Gambar III.5. Activity Diagram Menu Utama

III.3.2.2. Activity Diagram “Bantuan”

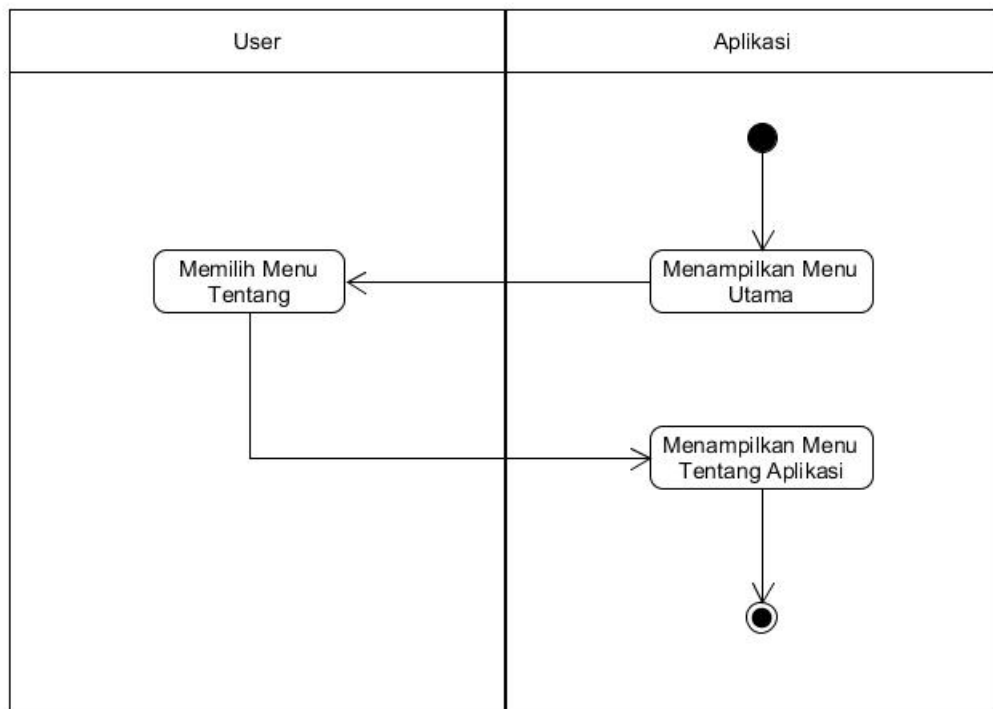
Menggambarkan aktivitas ketika pengguna memilih menu bantuan. Pengguna menekan tombol bantuan, selanjutnya sistem akan menampilkan menu bantuan yang berisi informasi bantuan cara menggunakan aplikasi, dapat dilihat pada gambar III.6.



Gambar III.6. Activity Diagram Bantuan

III.3.2.3. Activity Diagram “Tentang”

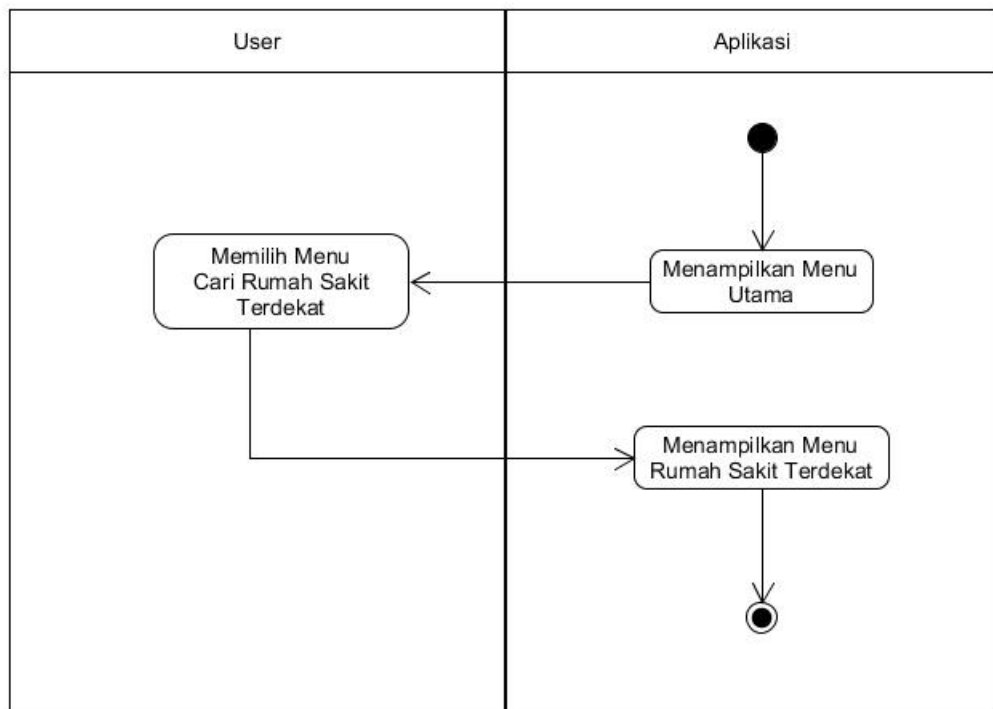
Menggambarkan aktivitas ketika pengguna memilih menu tentang. Pengguna menekan tombol tentang, selanjutnya sistem akan menampilkan menu tentang aplikasi Pencarian Rute Terdekat Rumah Sakit Menggunakan Metode Algoritma A -Star, yang dapat dilihat pada gambar III.7.



Gambar III.7. Activity Diagram Tentang

III.3.2.4. Activity Diagram “Cari Rumah Sakit Terdekat”

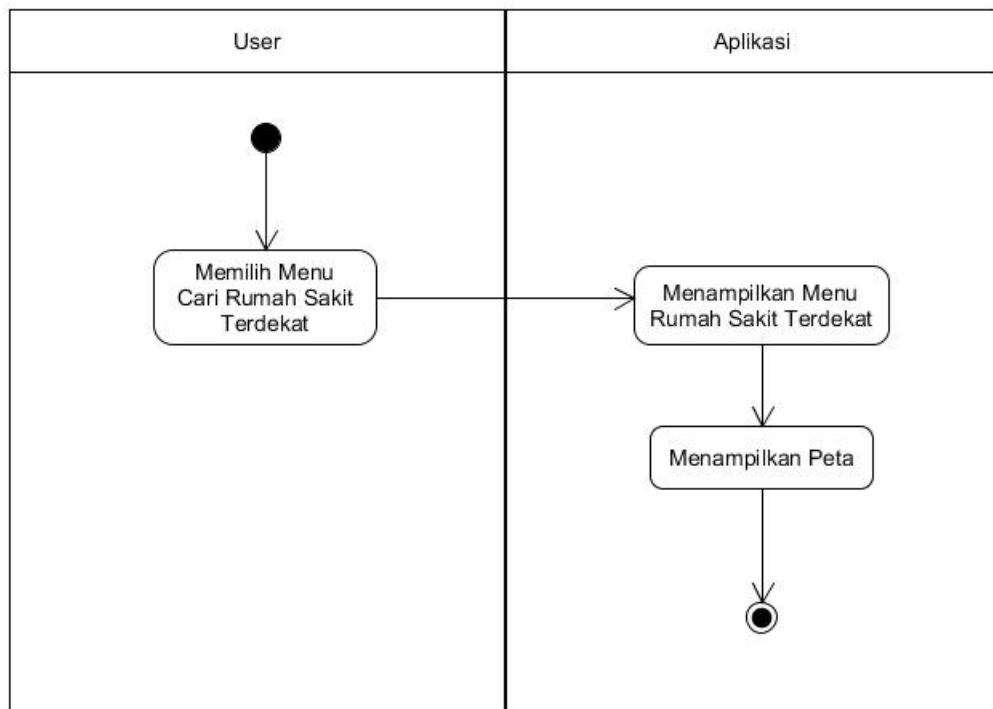
Menggambarkan aktivitas ketika pengguna memilih menu cari rumah sakit terdekat. Pengguna menekan tombol cari rumah sakit terdekat, selanjutnya sistem akan menampilkan menu mencari rute rumah sakit, yang dapat dilihat pada gambar III.8.



Gambar III.8. Activity Diagram Cari Rumah Sakit Terdekat

III.3.2.5. ActivityDiagram “Menampilkan Peta”

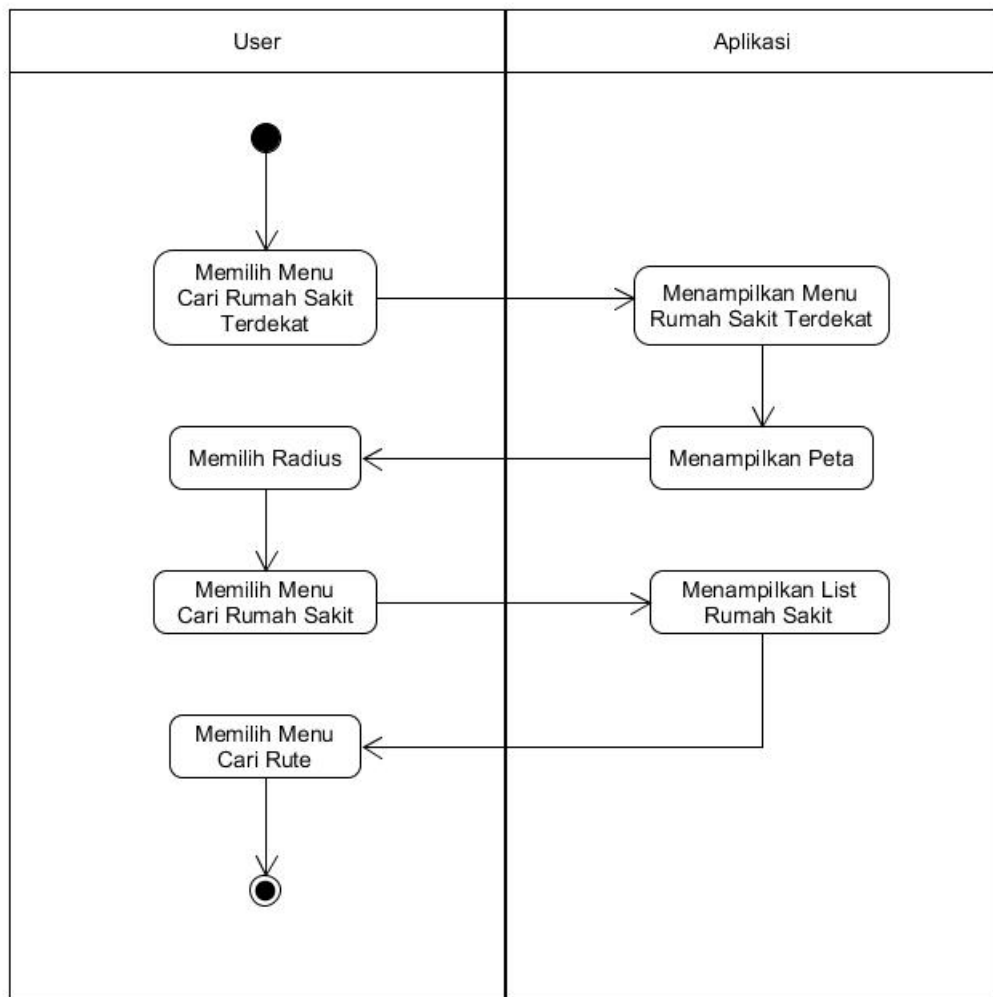
Menggambarkan aktivitas yang terjadi pada saat masuk pada menu cari rumah sakit terdekat, selanjutnya sistem akan menampilkan peta atau map,dapat dilihat pada gambar III.9.



Gambar III.9. Activity Diagram Menampilkan Peta

III.3.2.6. Activity Diagram “Cari Rute”

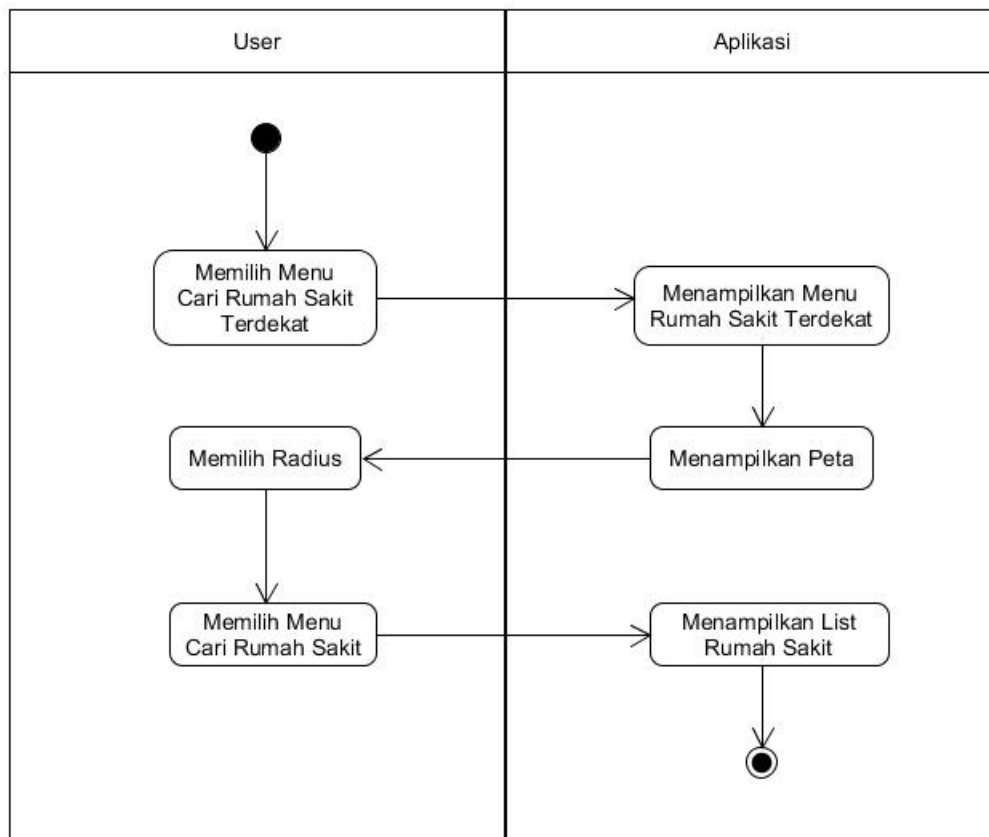
Menggambarkan Aktivitas yang terjadi pada aplikasi saat memilih menu cari rute, yang berfungsi mencari rute terdekat ke rumah sakit yang sudah di pilih, dapat dilihat pada gambar III.10.



III.10. Activity Diagram Cari Rute

III.3.2.7. ActivityDiagram “Cari Rumah Sakit”

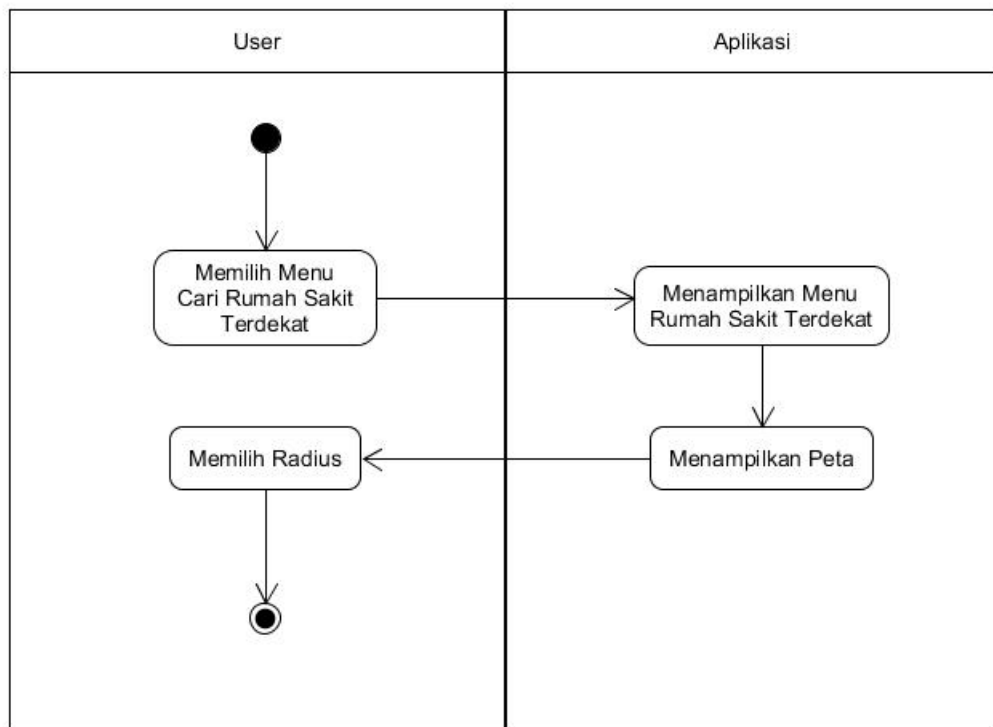
Menggambarkan Aktivitas yang terjadi pada aplikasi saat memilih menu cari rumah sakit, yang berfungsi mencari rumah sakit di radius yang sudah di pilih, dapat dilihat pada gambar III.11.



Gambar III.11. Activity Diagram Cari Rumah Sakit

III.3.2.8. Activity Diagram “Pilih Radius”

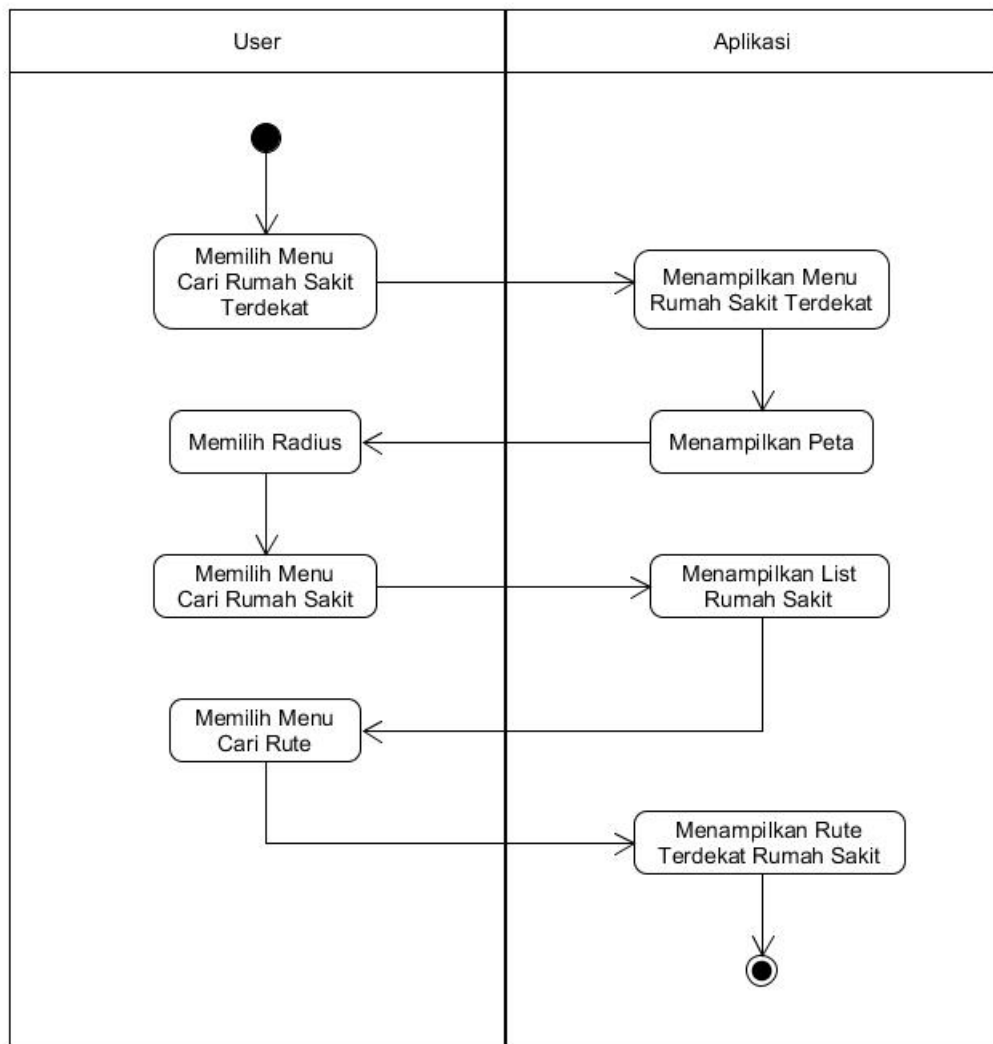
Menggambarkan aktivitas yang terjadi pada aplikasi saat memilih radius, yang berfungsi mencari radius rumah sakit yang dipilih, dapat dilihat pada gambar III.12.



Gambar III.12. Activity Diagram Pilih Radius

III.3.2.9. ActivityDiagram “Menampilkan Rute Terdekat Rumah Sakit”

Menggambarkan Aktivitas yang terjadi pada aplikasi saat menampilkan rute terdekat rumah sakit yang telah dipilih, dapat dilihat pada gambar III.13.



Gambar III.13. Activity Diagram Menampilkan Rute Terdekat Rumah Sakit

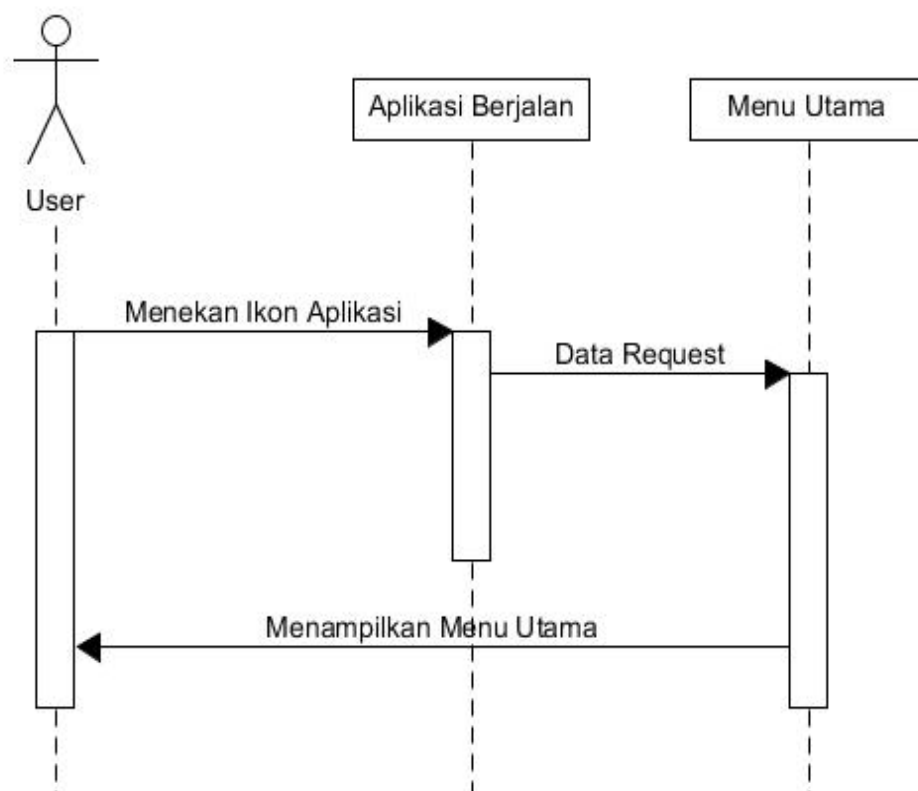
III.3.3. Sequence Diagram

Sequence Diagram (diagram urutan) akan menunjukkan operasi yang dilakukan secara detail dengan menggambarkan interaksi objek dan memberi petunjuk yang disusun dalam suatu urutan waktu. Berikut ini adalah *Sequence Diagram* untuk Pencarian Rute Terdekat Rumah Sakit

Menggunakan Metode Algoritma A –Star Berbasis *Android* dapat dilihat pada gambar III.11.

III.3.3.1. *Sequence Diagram “Menu Utama”*

Aktivitas ketika pengguna pertama kali membuka aplikasi, dapat dilihat pada gambar III.14.

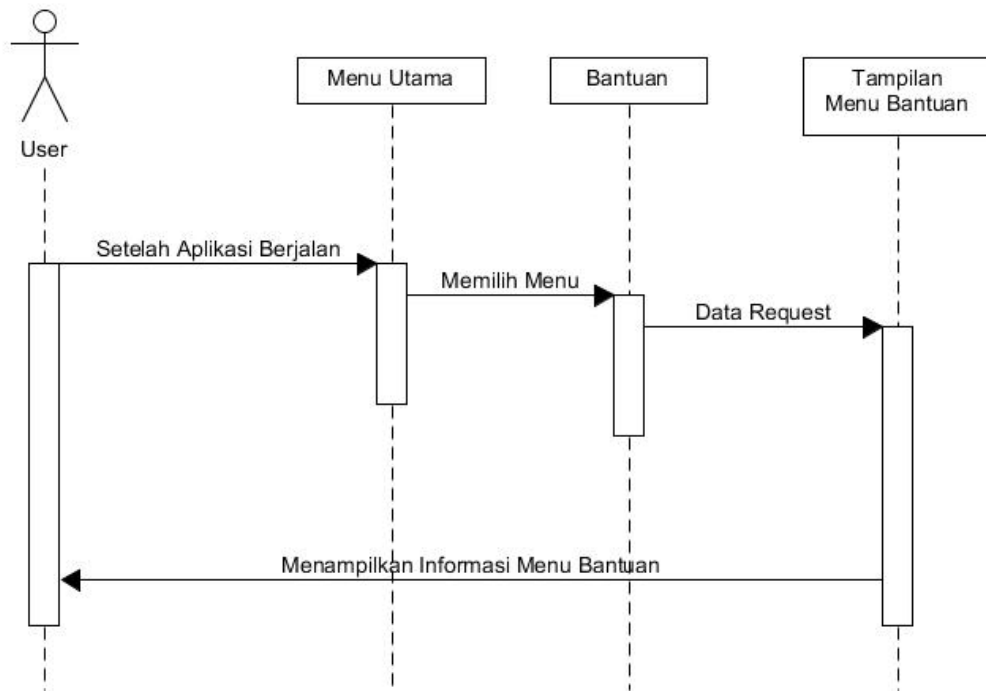


Gambar III.14. *Sequence Diagram Menu Utama*

III.3.3.2. *Sequence Diagram “Bantuan”*

Aktivitas ketika pengguna memilih menu bantuan. Pengguna menekan tombol bantuan, selanjutnya sistem akan menampilkan menu

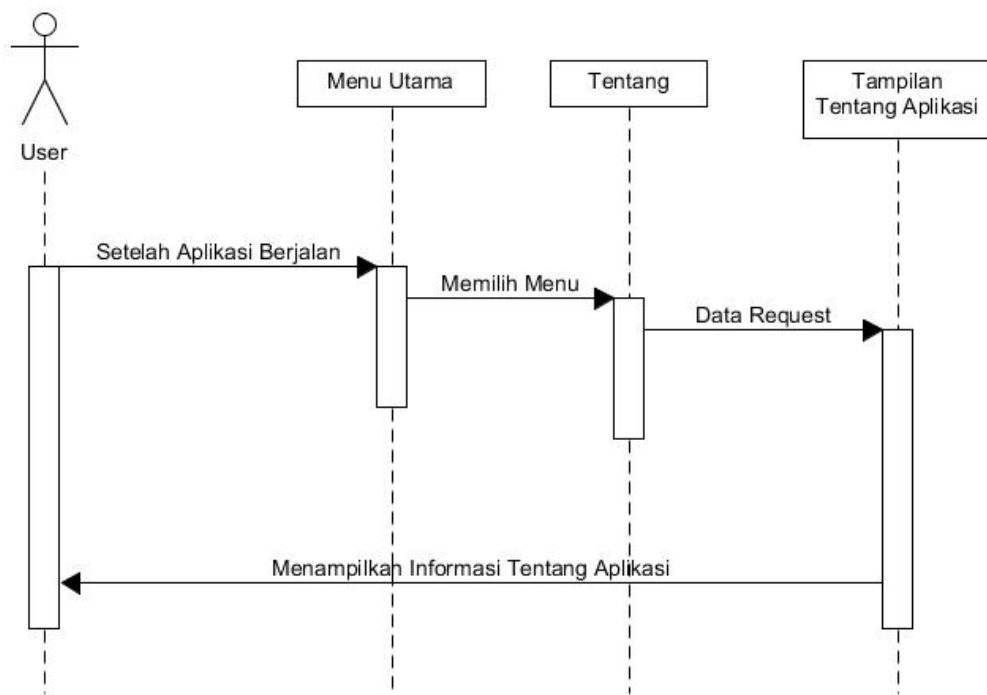
bantuan yang berisi informasi bantuan cara menggunakan aplikasi, dapat dilihat pada gambar III.15.



Gambar III.15. Sequence Diagram Bantuan

III.3.3.3. Sequence Diagram “Tentang”

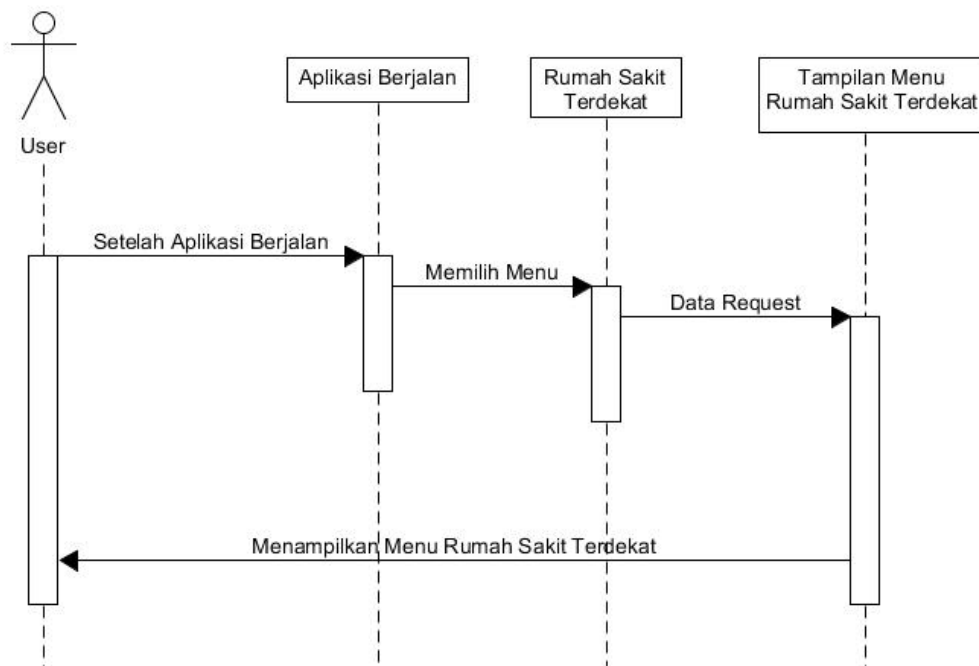
Menggambarkan aktivitas ketika pengguna memilih menu tentang. Pengguna menekan tombol tentang, selanjutnya sistem akan menampilkan menu tentang aplikasi Pencarian Rute Terdekat Rumah Sakit Menggunakan Metode Algoritma A -Star, yang dapat dilihat pada gambar III.16.



Gambar III.16. *Sequence Diagram* Tentang

III.3.3.4. *Sequence Diagram* “Cari Rumah Sakit”

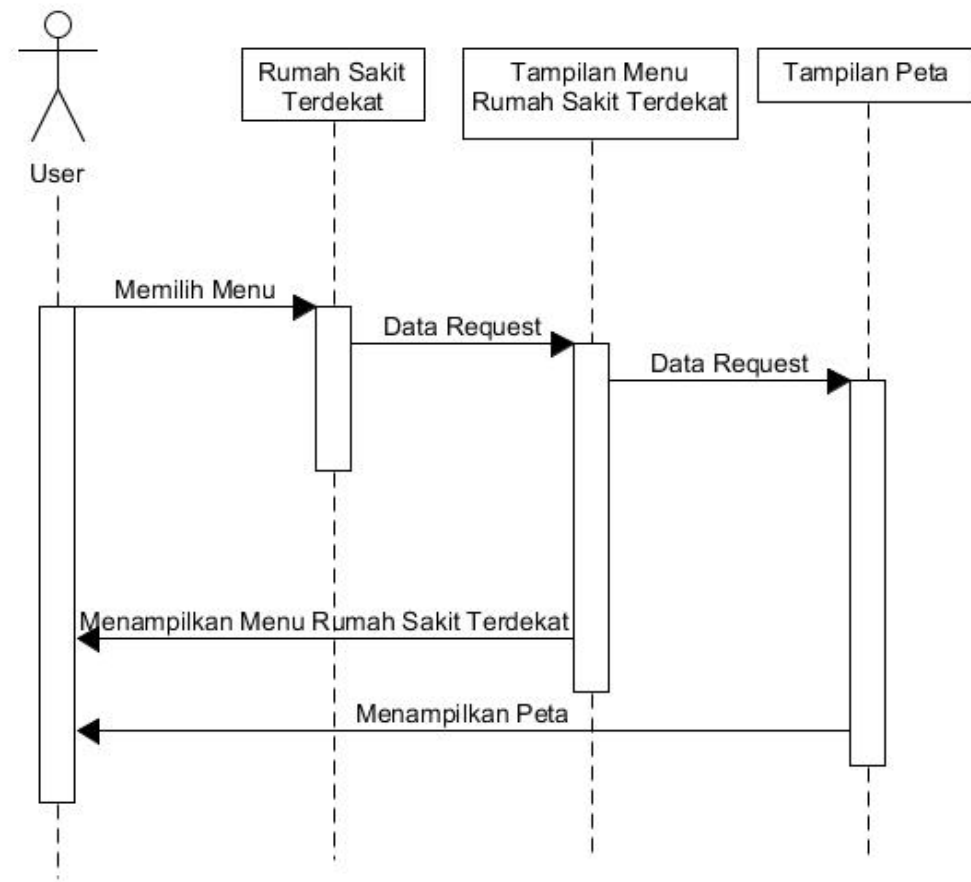
Aktivitas ketika pengguna memilih menu cari rumah sakit terdekat. Pengguna menekan tombol cari rumah sakit terdekat, selanjutnya sistem akan menampilkan menu mencari rute rumah sakit, yang dapat dilihat pada gambar III.17.



Gambar III.17. *Sequence Diagram* Cari Rumah Sakit

III.3.3.5. *Sequence Diagram* “Menampilkan Peta”

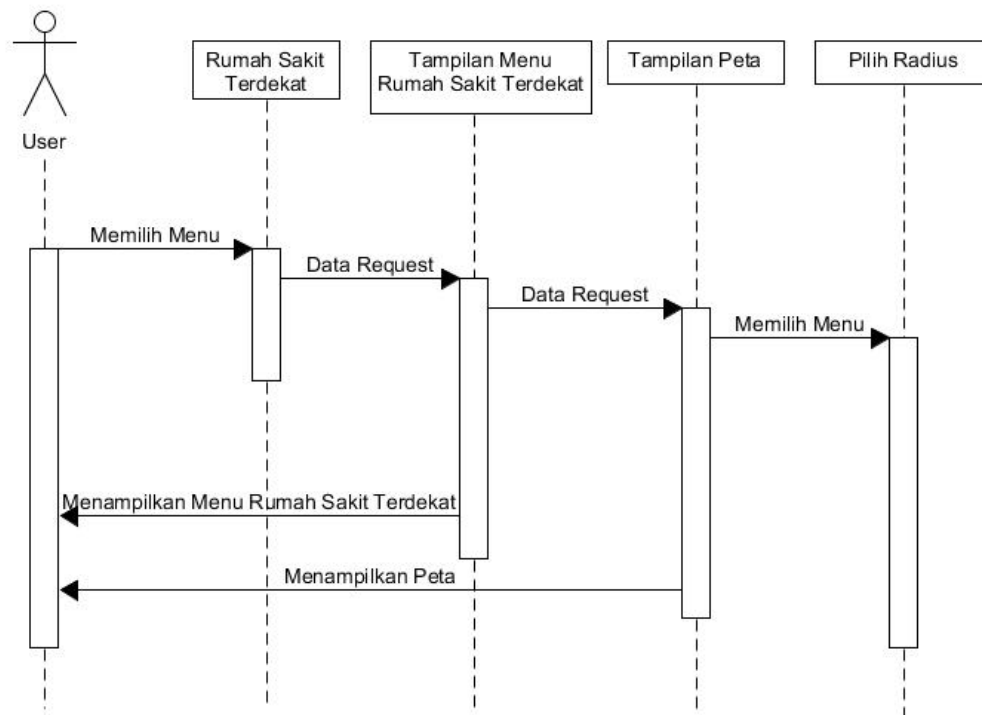
Aktivitas yang terjadi pada saat masuk pada menu cari rumah sakit terdekat, selanjutnya sistem akan menampilkan peta atau map, dapat dilihat pada gambar III.18.



Gambar III.18. *Sequence Diagram* Menampilkan Peta

III.3.3.6. *Sequence Diagram* “Pilih Radius”

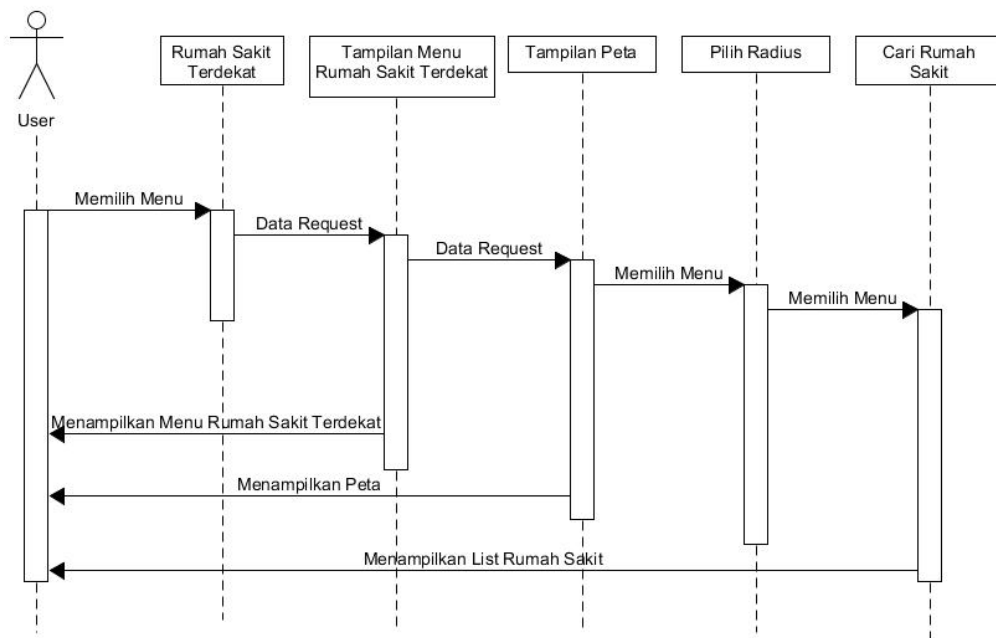
Mengambarkan aktivitas yang terjadi pada aplikasi saat memilih radius, yang berfungsi mencari radius rumah sakit yang dipilih, dapat dilihat pada gambar III.19.



Gambar III.19. Sequence Diagram Pilih Radius

III.3.3.7. Sequence Diagram “Cari Rumah Sakit”

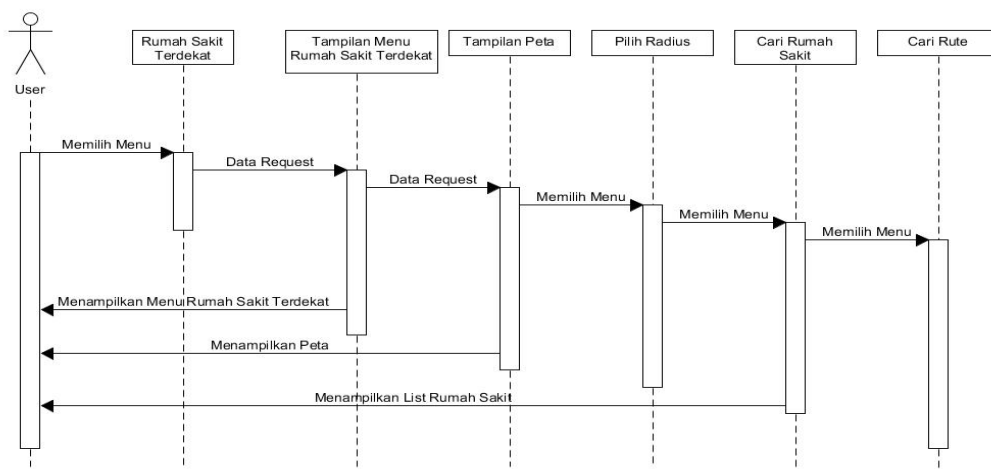
Menggambarkan Aktivitas yang terjadi pada aplikasi saat memilih menu cari rumah sakit, yang berfungsi mencari rumah sakit di radius yang sudah di pilih, dapat dilihat pada gambar III.20.



Gambar III.20. Sequence Diagram Cari Rumah Sakit

III.3.3.8. Sequence Diagram “Cari Rute”

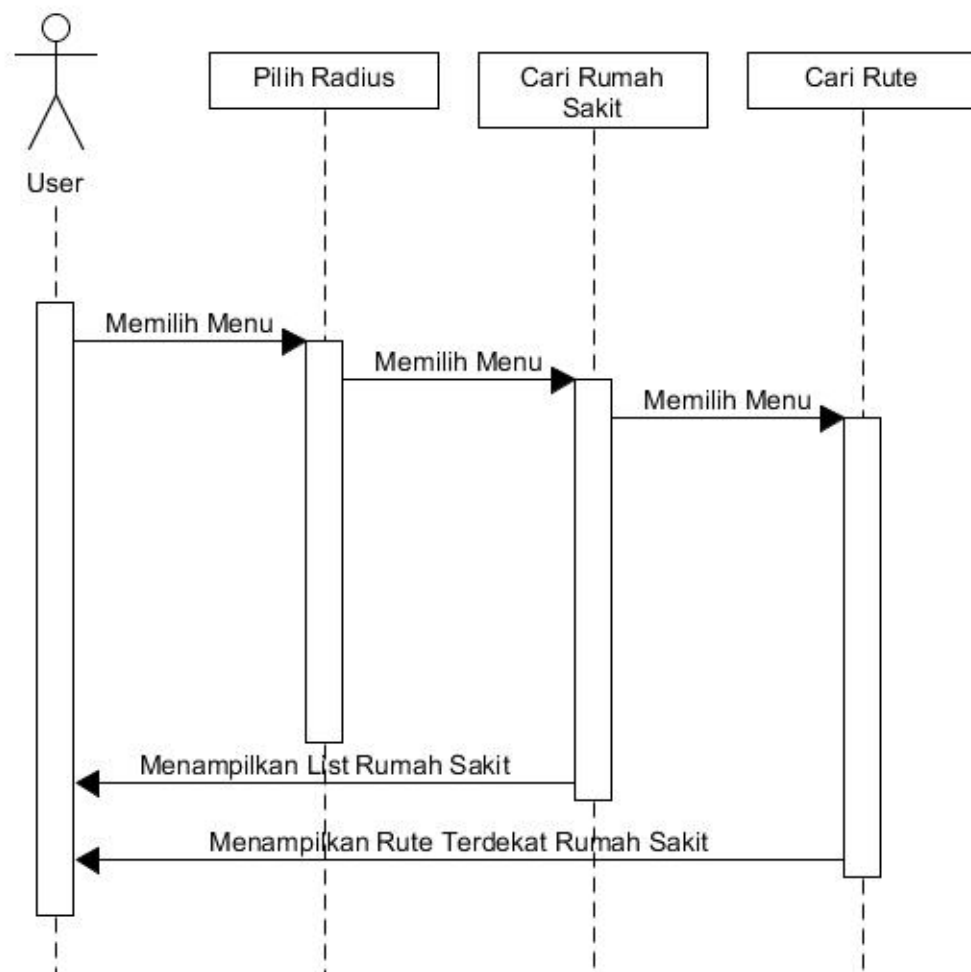
Menggambarkan Aktivitas yang terjadi pada aplikasi saat memilih menu cari rute, yang berfungsi mencari rute terdekat ke rumah sakit yang sudah di pilih, dapat dilihat pada gambar III.21.



Gambar III.21. Sequence Diagram Cari Rute

III.3.3.9. *Sequence Diagram* “Menampilkan Rute Terdekat Rumah Sakit”

Menggambarkan Aktivitas yang terjadi pada aplikasi saat menampilkan rute terdekat rumah sakit yang telah dipilih, dapat dilihat pada gambar III.22.



Gambar III.22. *Sequence Diagram* Menampilkan Rute Terdekat Rumah Sakit

III.4. Desain *User Interface*

Perancangan antarmuka merupakan tampilan dari aplikasi.

Perancangan antarmuka dapat dilihat sebagai berikut :

1. Tampilan Menu Utama

Tampilan menu utama dapat dilihat pada Gambar III.23.



Gambar III.23. Tampilan Menu Utama

2. Tampilan Sub Menu Utama

Tampilan sub menu terdiri dari menu cari rumah sakit terdekat, menu bantuan dan menu tentang, dapat dilihat pada Gambar III.24.



Gambar III.24. Tampilan Sub Menu Utama

3. Tampilan Menu Bantuan

Tampilan menu bantuan berisi informasi tentang bantuan menggunakan aplikasi, dapat dilihat pada Gambar III.25.



Gambar III.25. Tampilan Menu Bantuan

4. Tampilan Menu Tentang

Tampilan menu tentang berisi informasi tentang aplikasi, dapat dilihat pada Gambar III.26.



Gambar III.26. Tampilan Menu Tentang

5. Tampilan Menu Cari Rumah Sakit Terdekat

Tampilan menu cari rumah sakit terdekat memiliki beberapa menu yaitu:

- a. Menu Radius. yang berfungsi mencari radius dalam stuan meter (m) pada rumah sakit yang akan dicari.
- b. Menu Cari Rumah Sakit, berfungsi mencari rumah sakit dalam radius yang telah dipilih.
- c. Menu Rumah Sakit, yang berfungsi memilih rumah sakit yang tertera dalam radius yang telah dipilih.
- d. Menu Cari Rute,yang berfungsi mencari rute terpendek pada rumah sakit yang dipilih.

Tampilan menu cari rumah sakit terdekat dapat dilihat pada Gambar III.27.



Gambar III.27. Tampilan Menu Cari Rumah Sakit Terdekat