

BAB III

ANALISA DAN PERANCANGAN SOFTWARE

3.1 Analisa Kebutuhan Software

Penulis akan menjelaskan kebutuhan orang-orang untuk mencari tempat ibadah dengan cepat di tempat yang akan dikunjungi khususnya daerah Jakarta Timur, Sehingga untuk memudahkan pengguna untuk mencari tempat ibadah, maka penulis membuat aplikasi pencarian tempat-tempat ibadah tersebut.

3.1.1 Identifikasi masalah

Tempat ibadah merupakan salah satu lokasi yang banyak dicari oleh masyarakat untuk menjalankan ibadahnya masing-masing. Salah satu wilayah yang mempunyai banyak tempat ibadah yaitu di wilayah Jakarta timur, Dimana penduduknya menganut berbagai agama yang diakui oleh pemerintah Indonesia. Tidak semua orang mengetahui lokasi tempat ibadah tersebut, hal ini menjadi masalah jika masyarakat akan melaksanakan ibadah di tempat ibadah tersebut terutama masyarakat pendatang karena kesulitan itu maka penulis membuat aplikasi pencarian tempat ibadah ini.

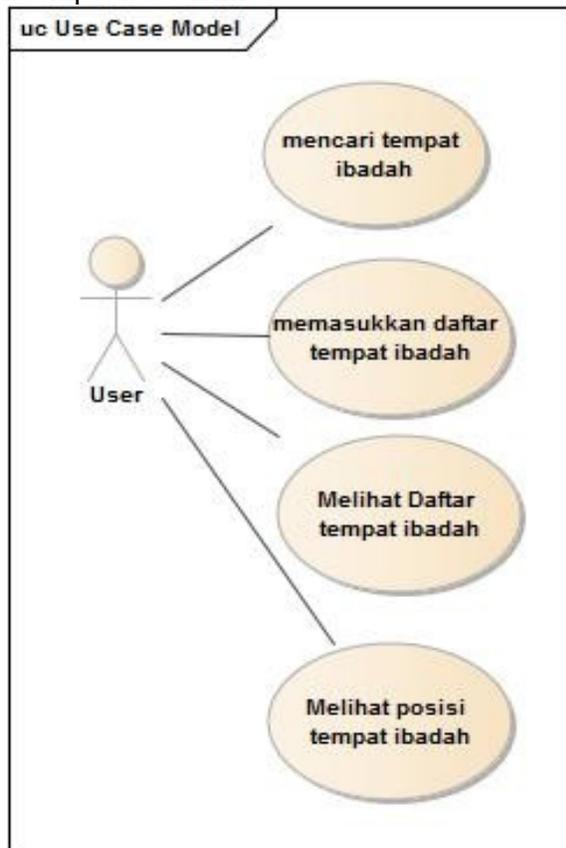
3.1.2 Analisa Kebutuhan

Setelah melakukan identifikasi masalah dan mendefinisikan masalah maka penulis merumuskan untuk kebutuhan permasalahan perangkat lunak yang dibutuhkan adalah aplikasi Eclipse sdk manager, UML(*Unified Modeling Language*),

dan smartphone android versi beta ke atas yang sudah memiliki paket data dan terhubung ke internet.

1. *Use Case Diagram*

Use case diagram di bawah ini menggambarkan hubungan antara User dan system, yang mana user dapat melihat secara garis besar fungsi dari ikon di aplikasi pencarian tempat ibadah ini.



Gambar III.1 *Use case diagram*

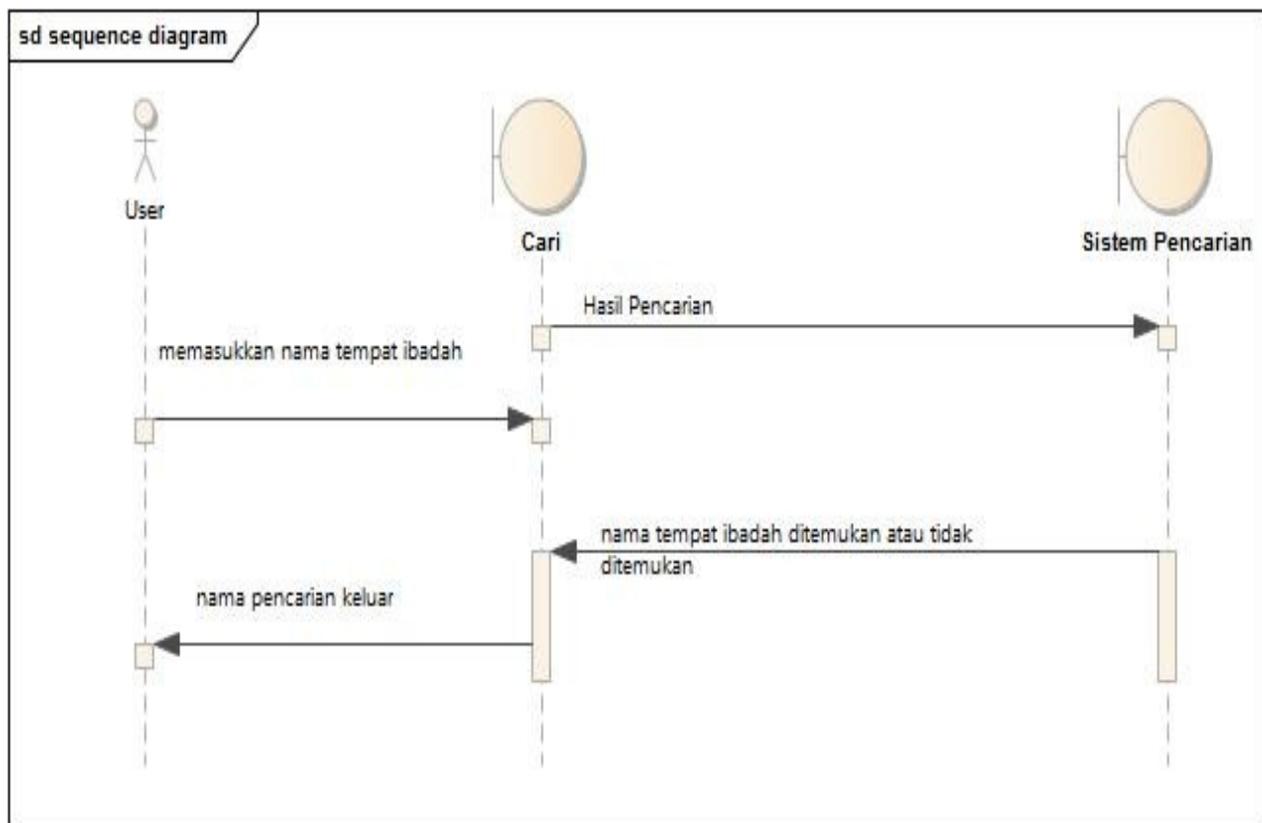
2. *Activity Diagram*

Dalam activity diagram ini penulis akan menjelaskan empat activity diagram yang ada di dalam aplikasi yang di buat.

Gambar III.2 Activity Diagram

3. Sequence Diagram

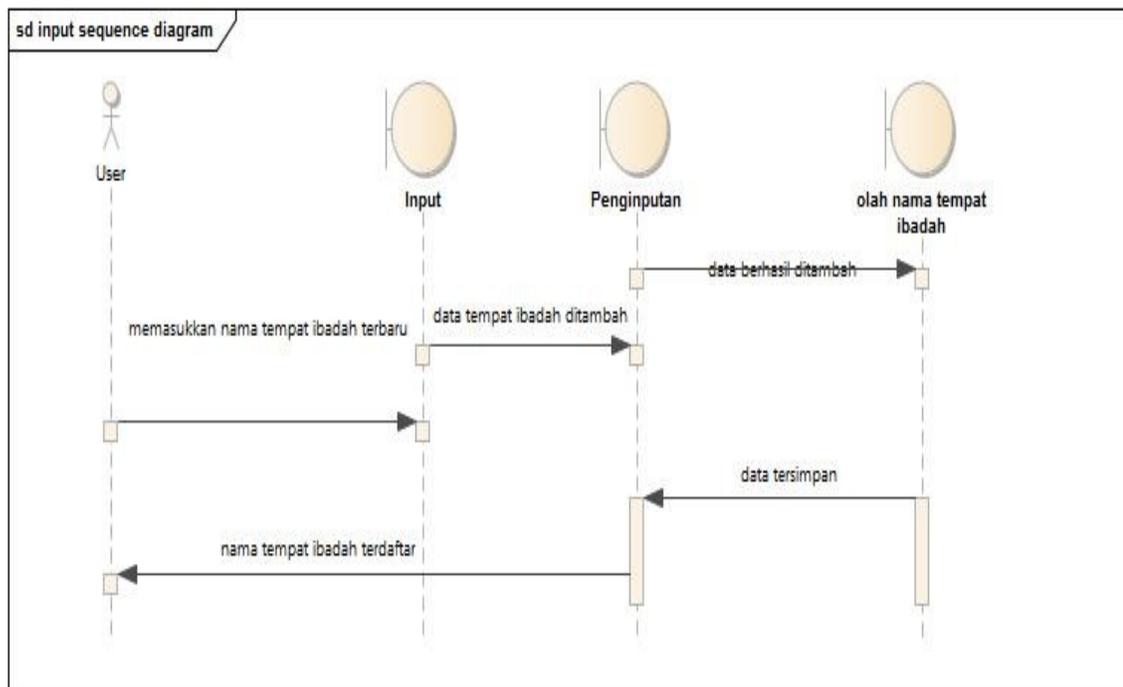
Sequence diagram menggambarkan interaksi antar objek di dalam suatu system. Untuk aplikasi pencarian tempat ibadah ini akan di buat empat



sequence diagram yaitu : Cari, Input, List, dan Peta.

3.1 Sequence Cari

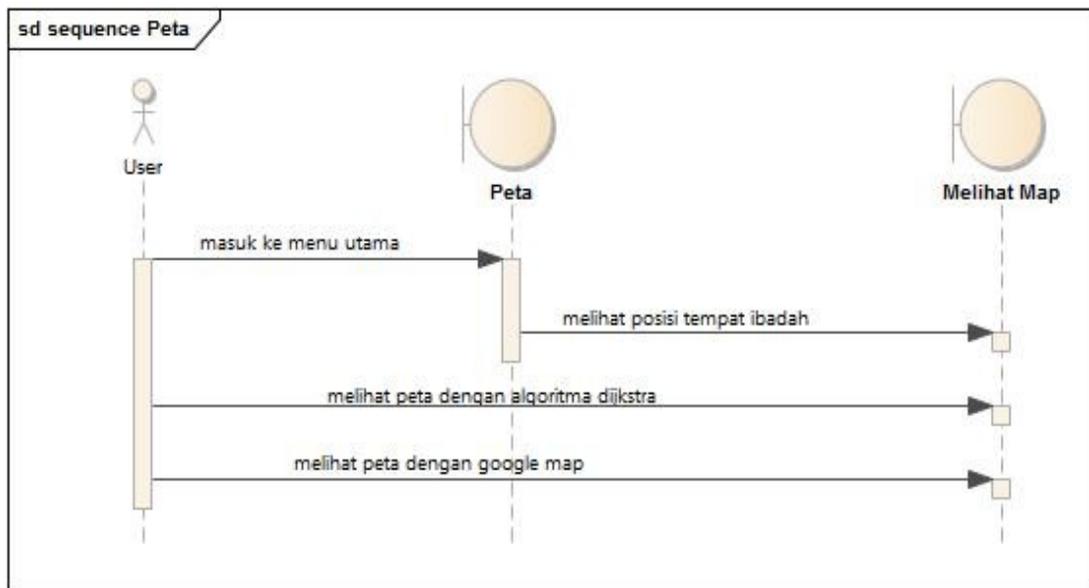
Gambar III.3 Sequence Cari



3.2 Sequence Input

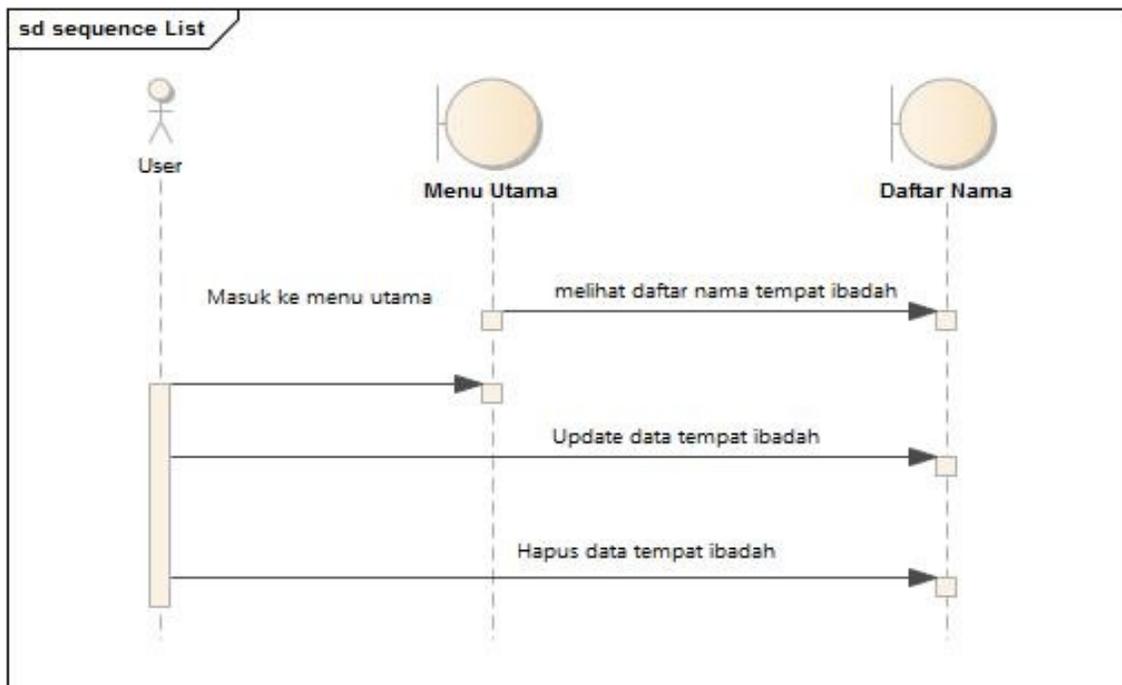
Gambar III.4 Sequence Input

3.3 Sequence Peta



Gambar III.5 Sequence Peta

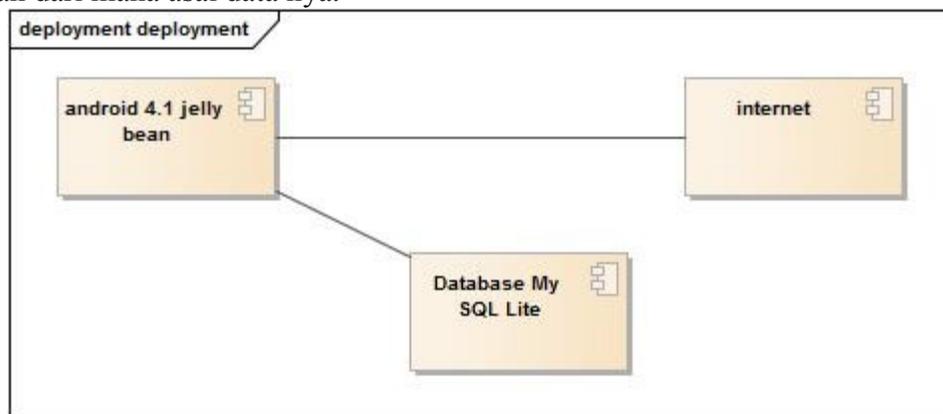
3.4 Sequence List



Gambar III.6 Sequence List

4. Deployment Diagram

Di sini penulis akan menjelaskan tentang dari mana User mengambil data base dan dari mana asal data nya.



Gambar III.7 Deployment Diagram

5. Class Diagram

3.2 Desain

3.2.1 Algoritma Pada Kasus

Algoritma yang digunakan untuk membuat aplikasi ini adalah algoritma dijkstra, yang mana algoritma ini digunakan untuk mencari jalur termurah dari satu vertex awal ke vertex akhir. Algoritma ini berbasis pada teknik *greedy*, di mana digunakan sebuah himpunan vertex S yang mula-mula diisi dengan vertex awal. Pada setiap langkah vertex v berikutnya memiliki bobot terkecil atau biaya termurah, ditambahkan. Sebuah Array D biasa digunakan untuk merekam panjang jalur terpendek dari vertex awal ke vertex akhir.

3.2.2 Database

3.2.3 Software Architecture

Pseudocode Algoritma Dijkstra

```

for (var i = 0; i <= numSteps; ++i) {
  nextPher[currPath[i]][currPath[i+1]] += (bestTrip - asymptoteFactor * bestTrip) /
  (numAnts * (currDist - asymptoteFactor * bestTrip));
}
}
for (var i = 0; i < numActive; ++i) {
  for (var j = 0; j < numActive; ++j) {
pher[i][j] = pher[i][j] * (1.0 - rho) + rho * nextPher[i][j];
  nextPher[i][j] = 0.0;
}
}
}
}

function nextSetOf(num) {
  var count = 0;
  var ret = 0;
  for (var i = 0; i < numActive; ++i) {
    count += nextSet[i];
  }
  if (count < num) {
    for (var i = 0; i < num; ++i) {
      nextSet[i] = 1;
    }
  }
}

```

```

    }
    for (var i = num; i < numActive; ++i) {
        nextSet[i] = 0;
    }
} else {
    // temukan satu yang pertama
    var firstOne = -1;
    for (var i = 0; i < numActive; ++i) {
        if (nextSet[i]) {
            firstOne = i;
            break;
        }
    }

    var firstZero = -1;
    for (var i = firstOne + 1; i < numActive; ++i) {
        if (!nextSet[i]) {
            firstZero = i;
            break;
        }
    }
    if (firstZero < 0) {
        return -1;
    }
    for (var i = 0; i < firstZero - firstOne - 1; ++i) {
        nextSet[i] = 1;
    }
    for (var i = firstZero - firstOne - 1; i < firstZero; ++i) {
        nextSet[i] = 0;
    }
}
for (var i = 0; i < numActive; ++i) {
    ret += (nextSet[i]<<i);
}
return ret;
}

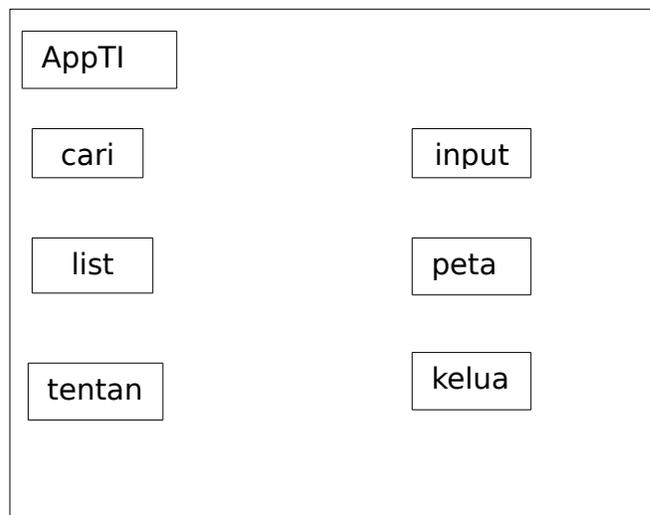
function makeLatLng(latLng) {
    return(latLng.toString().substr(1,latLng.toString().length-2));
}

function getWayStr(curr) {
    var nextAbove = -1;
    for (var i = curr + 1; i < waypoints.length; ++i) {

```

```
if (wpActive[i]) {  
  if (nextAbove == -1) {  
    nextAbove = i;  
  } else {  
    wayStr.push(makeLatLng(waypoints[i]));  
    wayStr.push(makeLatLng(waypoints[curr]));  
  }  
}  
}  
}  
if (nextAbove != -1) {  
  wayStr.push(makeLatLng(waypoints[nextAbove]));  
  getWayStr(nextAbove);  
  wayStr.push(makeLatLng(waypoints[curr]));  
}  
}
```

3.2.4 User Interface

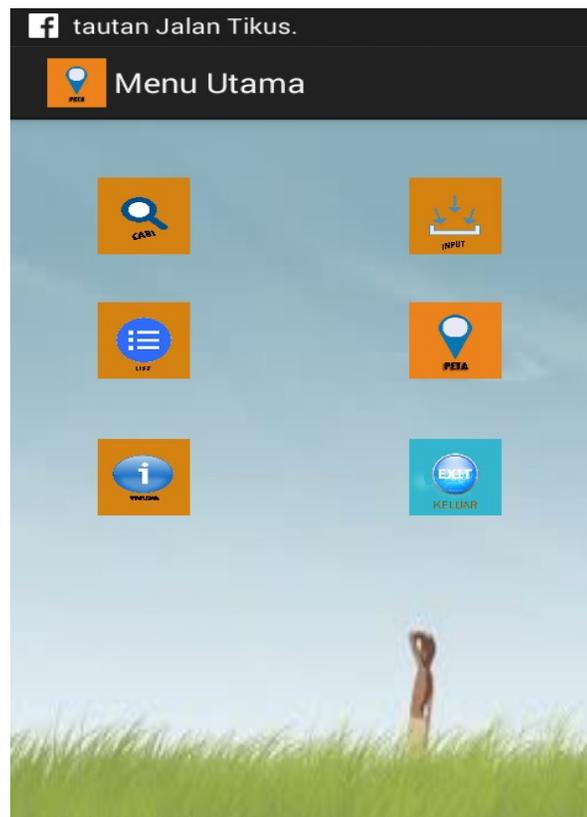


Gambar III.9 Tampilan Dasar Aplikasi

3.3 Implementasi

1. Tampilan Menu Utama

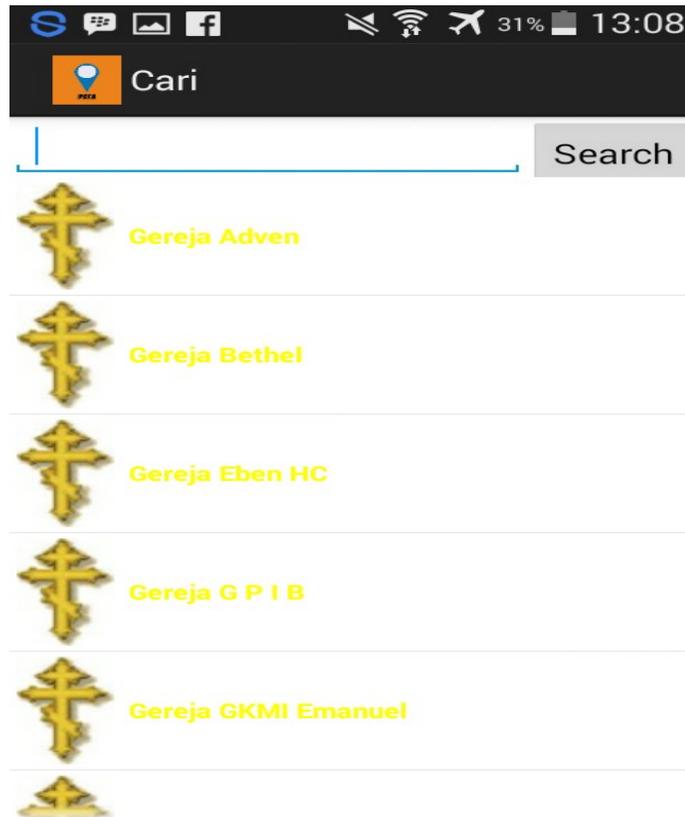
Di bawah ini adalah tampilan dairi menu utama aplikasi pencarian tempat ibadah yang penulis buat. Di sini penulis membuat beberpa ikon yang terdiri dari yaitu: Ikon Cari, Ikon Input, Ikon List, Ikon Peta, dan Ikon Tentang.berikut adalah tampilan nya.



Gambar III.10 Tampilan Menu Utama

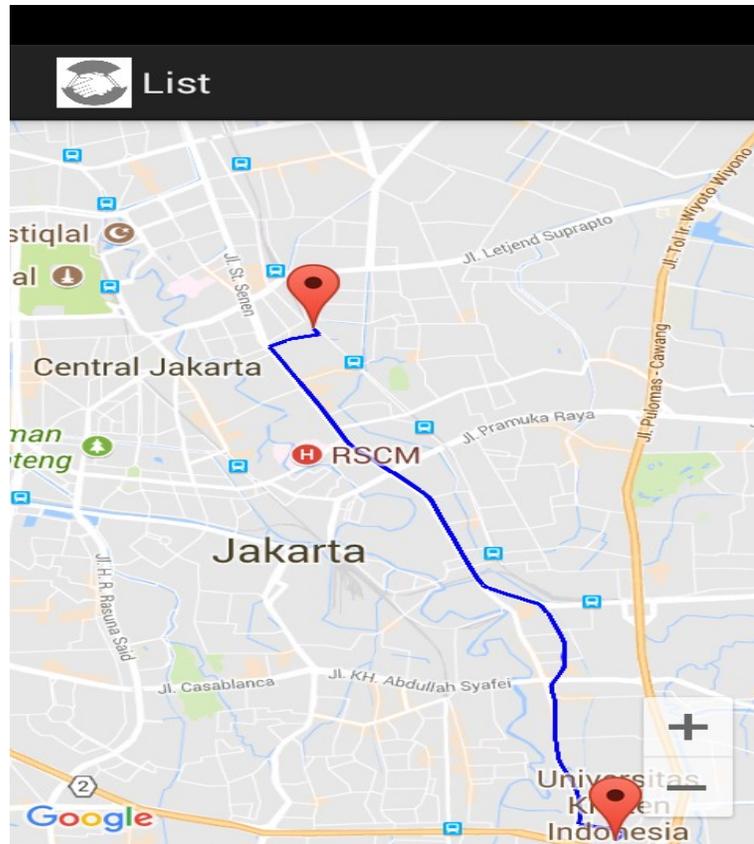
2. Tampilan Menu Cari

Dalam Ikon Cari ini User bisa melihat daftar cari dari nama tempat ibadah yang ingin di cari secara cepat. Berikut ini adalah tampilan dari Ikon Cari nya.



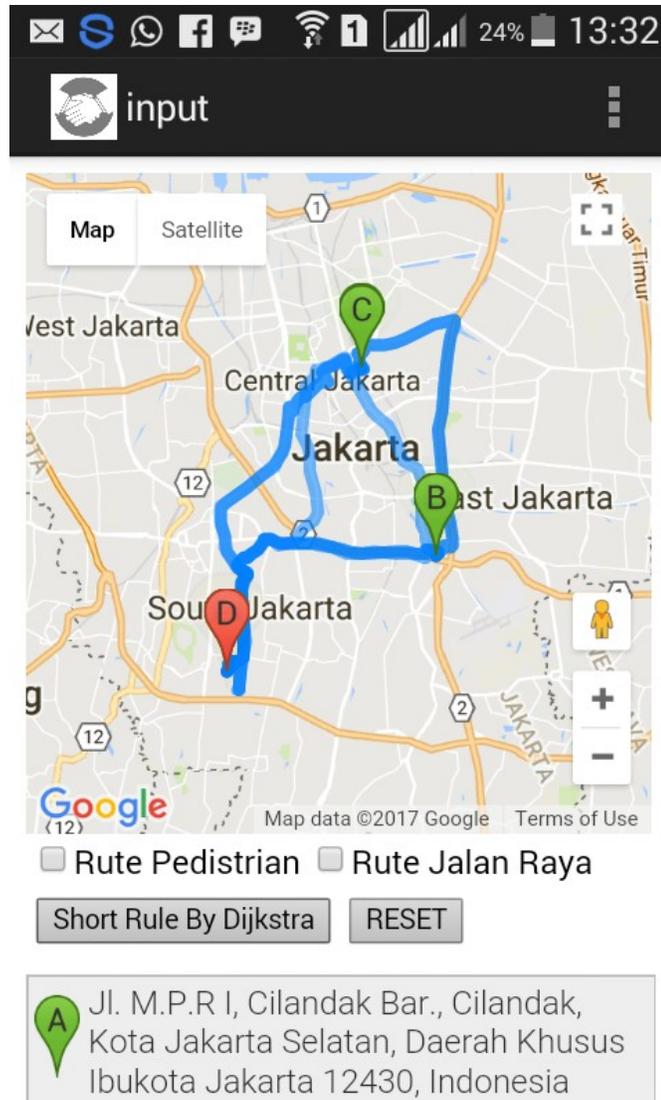
Gambar III.11 Tampilan Menu Cari

3. Tampilan menu direction



Gambar III.12 Tampilan direction

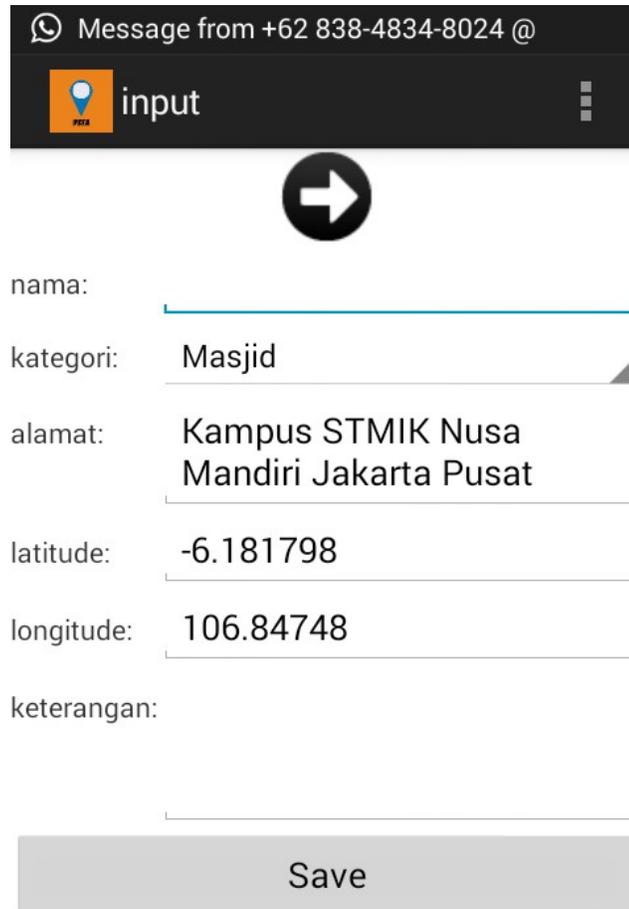
4. Tampilan Peta Dijkstra



Gambar III.13 Tampilan peta dijkstra

5. Tampilan Menu Input

Di dalam tampilan ikon input ini pengguna bisa menambahkan nama tempat ibadah yang baru dibangun. Jika pengguna sudah memasukkan data tempat ibadah secara benar maka nama tempat ibadah tersebut akan tersimpan secara otomatis.



Message from +62 838-4834-8024 @

input

nama: _____

kategori: Masjid

alamat: Kampus STMIK Nusa Mandiri Jakarta Pusat

latitude: -6.181798

longitude: 106.84748

keterangan: _____

Save

Gambar III.14 Tampilan Menu Input

6. Tampilan input tidak lengkap



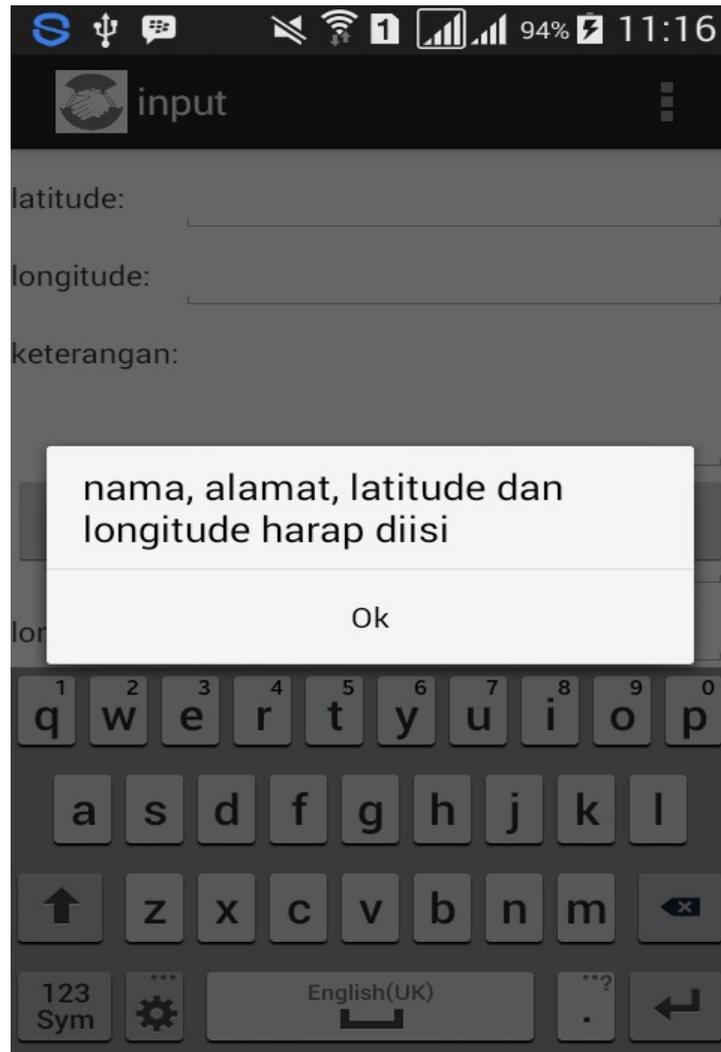
The screenshot shows an Android application interface with a dark header bar. The header bar contains a status bar at the top with various icons (Bluetooth, USB, camera, messages, mute, Wi-Fi, signal strength, battery) and the time 11:16. Below the status bar is a dark bar with a white icon of a hand holding a globe and the text 'input'. The main content area is white and contains several input fields:

- kategori:** A dropdown menu with the text 'Masjid'.
- alamat:** A text input field containing 'Kampus STMIK Nusa Mandiri Jakarta Pusat'.
- latitude:** An empty text input field.
- longitude:** An empty text input field with a blue border, indicating it is the active field.
- keterangan:** An empty text input field.
- longitude:** A text input field containing the value '106.84748'.

At the bottom of the screen, a virtual keyboard is visible, showing the letters 'q w e r t y u i o p' on the top row, 'a s d f g h j k l' on the second row, and '↑ z x c v b n m ↵' on the third row. The keyboard is set to 'English(UK)' and has a '123 Sym' button on the left and a '↵' button on the right.

Gambar III.15 Tampilan input tidak lengkap

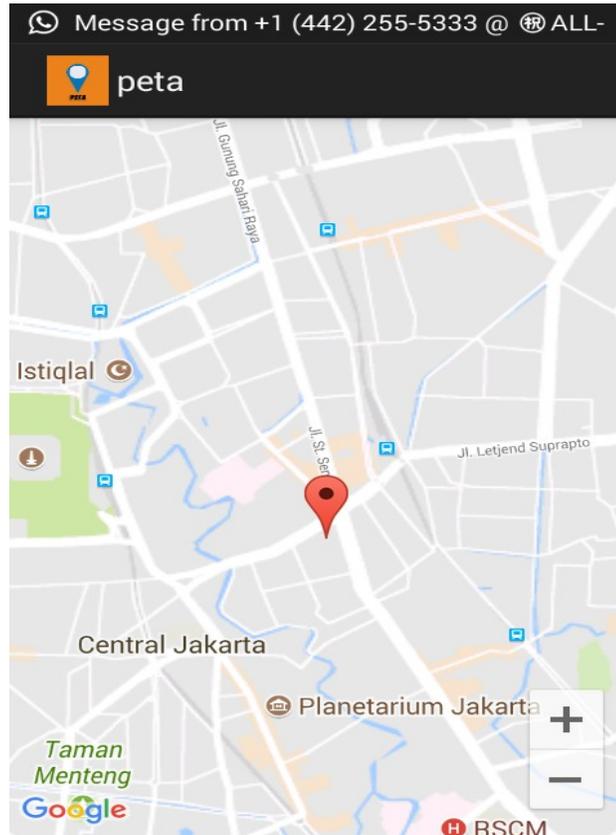
7. Tampilan hasil input tidak lengkap



Gambar III.16 Tampilan hasil input tidak lengkap

8. Tampilan Menu Peta

Dalam ikon peta ini pengguna bisa melihat posisi dari nama-nama tempat ibadah yang sudah ada dalam daftar List, Tapi menunjukkan posisi kita berada. Dengan menggunakan GPS dalam android kita sendiri.



Gambar III.17 Tampilan Menu Peta

9. Tampilan Menu List

Dalam ikon list ini pengguna bisa melihat daftar daaari nama tempat ibadah yang sebelumnya sudah dimasukkan dalam ikon input.



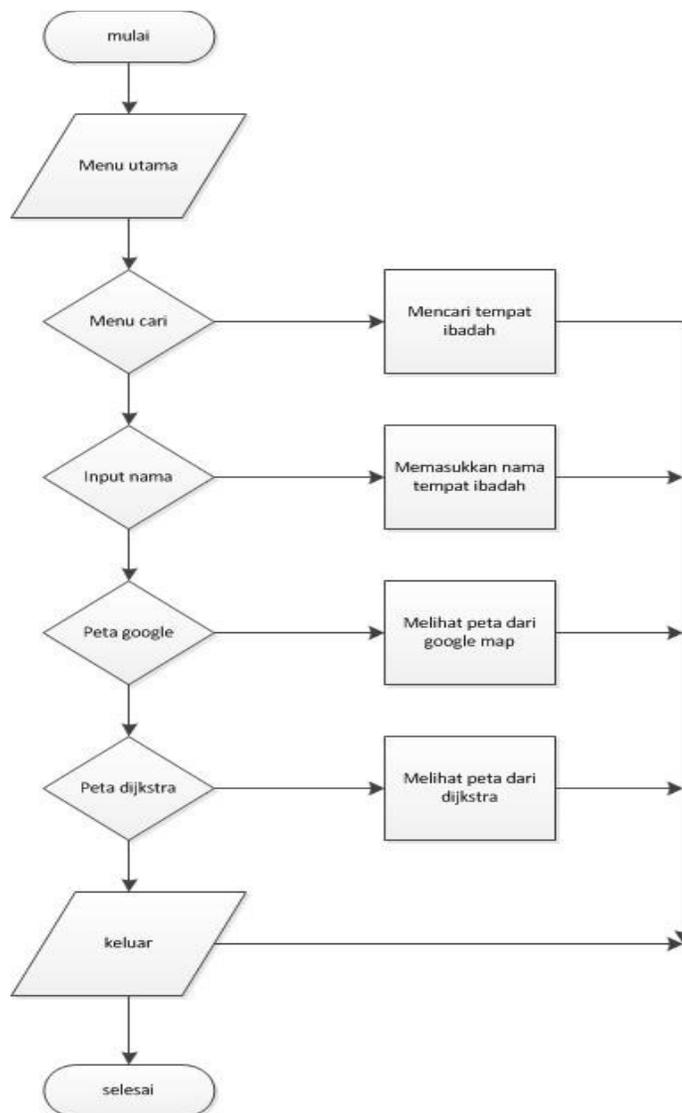
Gambar III.18 Tampilan Menu List

3.4 Testing

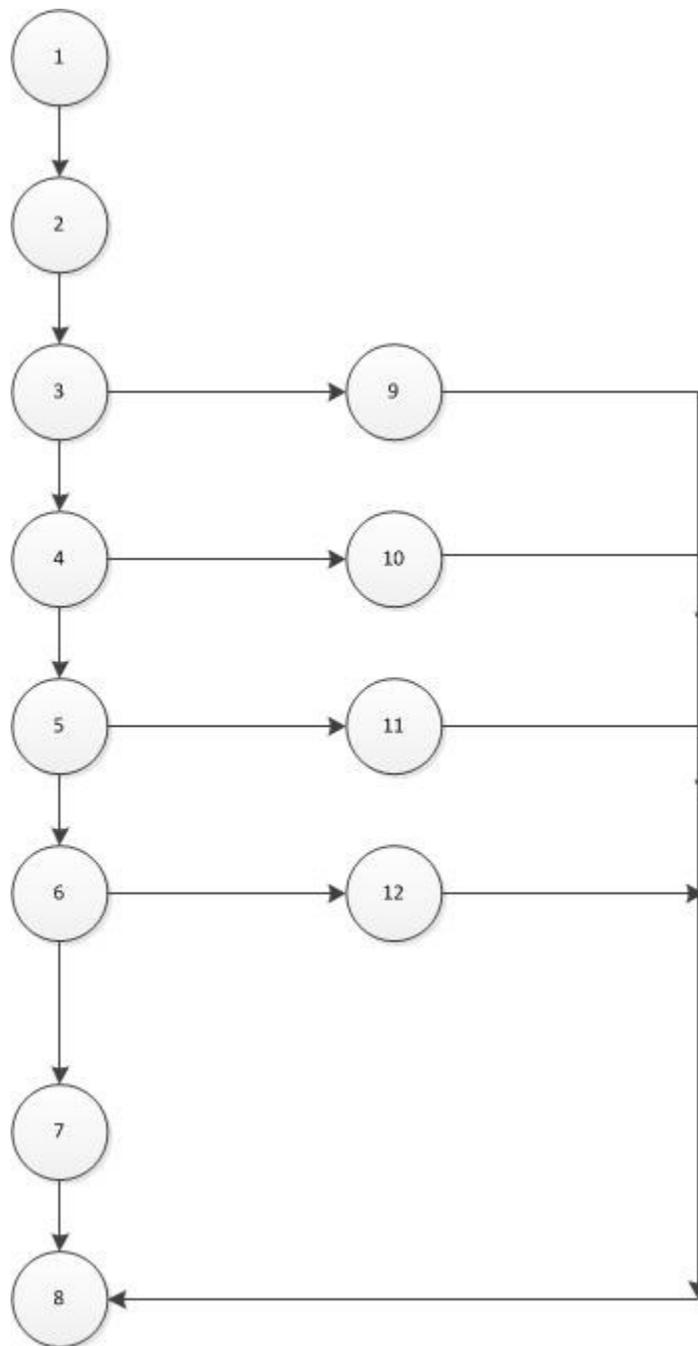
Untuk melihat hasil keakurasian yang diperoleh dari aplikasi dan untuk mengurangi kesalahan yang ada dalam aplikasi maka penulis menggunakan Uji coba menggunakan *white box* dan *black box testing*.

3.4.1 *white box testing*

Bagan Alir form cari tempat ibadah



Gambar III.15 *Bagan alir*



Gambar III.16 Grafik alir pencarian tempat ibadah

Pada pengujian ini penulis akan menjelaskan digram alir di atas untuk memudahkan kita menentukan region nya.

Kompleksitas siklomatis $V(G)$ dirumuskan dengan

$$V(G) = E - N + 2$$

Dimana

E = jumlah lingkaran pada diagram alir

N = jumlah panah pada diagram alir

Berdasarkan perhitungan diatas di dapat jumlah jalur independent nya adalah 7.

Berikut adalah jalur independent yang dihasilkan oleh kompleksitas siklomatis nya:

1. 1-2-3-9-8
2. 1-2-3-4-10-8
3. 1-2-3-4-5-11-8
4. 1-2-3-4-5-6-12-8
5. 1-2-3-4-5-6-7-8

Pada aplikasi ini ada beberapa sample yang diuji yaitu pengujian *whitebox* aplikasi pencarian tempat ibadah. Algoritma nya yaitu:

1. Pilih tombol cari maka user akan masuk ke halaman kombo box nama tempat ibadah yang akan di cari.
2. Pengguna bisa mencari nama tempat ibadah yang dimasukkan di kotak.
3. Pilih tombol direction maka user akan melihat peta dari google map.
4. Pilih tombol dijkstra maka user akan melihat jalur terpendek dari posisi pengguna.

3.4.2 black box testing

A. *black box testing* form Cari tempat ibadah

NO	Input	Output	Hasil
1	Memilih tombol Cari	Tampil Combo box untuk memasukkan nama tempat ibadah yang akan di cari	Berhasil
2	Memasukkan nama tempat ibadah	Tampil nama ibadah yang dicari	Berhasil
3	Memilih tombol direction	Tampil jalur yang akan dilalui oleh pengguna dari google map	Berhasil
4	Memilih tombol dijkstra	Tampil jalur terpendek yang akan dipilih pengguna	Berhasil

Tabel III.1 *Black box testing*

3.5 Support

Aplikasi pencarian tempat ibadah ini dibangun menggunakan *android development tools* (ADT) V21.0.1. Aplikasi ini bisa berjalan pada hardware manapun yang berbasis android. Spesifikasi system operasi android nya adalah android yang paling minimum yang digunakan adalah android 4.1 (jelly bean). Sedangkan hardware computer yang digunakan adalah prosesor intel CPU N2600 @ 1.60hz dengan RAM 2GB.