APLIKASI SISTEM INFORMASI RUMAH KOST KOTA PANGKALPINANG BERBASIS ANDROID MENGGUNAKAN ALGORITMA DIJKSTRA

Laurentinus, S.Kom., M.Kom[1] , Vinsensius Julio[2]
STMIK ATMA LUHUR Jl. Jend. Sudirman, Selindung, Pangkalpinang Kepulauan Bangka Belitung laurentinus@atmaluhur.ac.id, 1311500001@mahasiswa.atmaluhur.ac.id

Abstrak

Perkembangan teknologi di era globalisasi saat ini berlangsung sangat cepat. Teknologi informasi sudah menjadi salah satu kebutuhan dan keharusan dalam segala aspek kehidupan. Tanpa kita sadari bahwa dengan terus berkembangnya teknologi sekarang ini telah membawa manusia untuk berpikir melakukan sesuatu dengan cepat mudah. Praktis dan bersifat mobile terutama dalam bidang informasi dan komunikasi, Salah satu teknologi yang berkembang yaitu sistem informasi geografis, dengan media layanan internet dan sistem navigasi atau GPS yang terdapat pada smartphone,serta Google Maps produk google jasa peta yang bersifat virtual,gratis dan online. Rumah Kost merupakan tempat yang selalu dicari oleh masyarakat menengah kebawah, mahasiswa, pekerja swasta, pelajar, dll, Namun sayangnya lokasi kost yang tidak terpublikasi membuat masyarakat kesulitan dalam mencari lokasinya. Dibutuhkan aplikasi yang dapat memetakan lokasi kost di wilayah Pangkalpinang untuk mempermudah masyarakat dalam mencari kost serta bersifat mobile sehingga masyarakat bisa mengakses peta dimanapun dan kapanpun saat dibutuhkan. Model yang digunakan pada pembuatan aplikasi ini adalah model prototype dan metode yang digunakan adalah metode berorientasi objek, diagram UML yang digunakan antara lain adalah diagram activity, use case diagram, class diagram, sequence diagram dan deployment diagram, algortitma yang diterapkan untuk aplikasi ini adalah algoritma dijkstra dimana algoritma tersebut digunakan untuk menentukan jalur terpendek user dalam mencapat kost tujuan. Hasil dari penelitian ini merupakan sistem informasi geografis rumah kost di pangkalpinang yang dapat membantu masyarakat lokal maupun luar dari pangkalpinang dalam menemukan rumah kost.

Kata kunci : Metode Berorientasi Objek, Mobile, Maps, Sistem Informasi Geografis, Kost, Prototype, UML, dan Dijkstra.

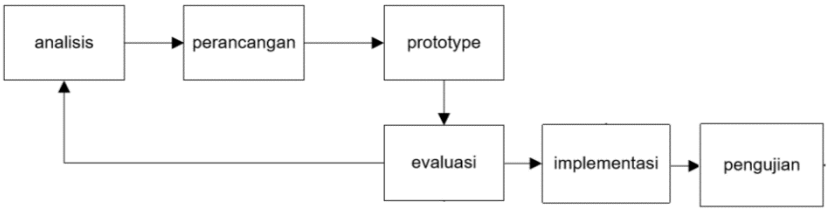
I. PENDAHULUAN

Pertumbuhan penduduk yang semakin tinggi mengakibatkan bertambahnya kebutuhan tempat tinggal, terutama oleh orang-orang yang berkerja atau pergi kuliah/sekolah ditempat yang jauh dari tempat tinggal aslinya. Hal ini juga terjadi dikota Pangkalpinang dimana semakin banyak pendatang dari luar daerah maupun dari seputaran kota Pangkalpinang, ketika para pendatang ini datang ke kota Pangkalpinang mereka akan mencari tempat untuk tinggal, terutama pelajar-pelajar dari luar daerah yang mencari kost-kostan untuk tinggal kesulitan mencari tempat kost dengan harga yang murah dan dekat dengan tempat sekolah ataupun kerja terjadi karena mereka berada didaerah baru. Selain harga yang murah dan dekat dengan tempat kerja atau sekolah mereka juga menginginkan tempat kost yang memiliki fasilitas yang baik seperti air, ranjang tidur, dan sebagainya.

Berdasarkan masalah diatas, maka diperlukan aplikasi yang dirancang untuk mempermudah dalam pencarian rumah kost terdekat. Algoritma yang digunakan dalam pencarian rute terpendek ke kost yaitu dengan menerapkan algoritma Dijkstra. Algoritma Dijkstra merupakan algoritma yang dapat memecahkan masalah pencarian jalur terpendek dari suatu graf pada setiap simpul yang bernilai tidak negatif.. Kelebihan dari algoritma dijkstra yaitu : Algoritma Dijkstra dapat menentukan jalur tercepat dengan waktu yang lebih cepat dibandingkan algoritma lainnya, menggunakan Algoritma Dijkstra mempermudah kita dalam mengetahui jarak atau lintasan terpendek dari suatu titik tertentu ke semua titik yang lain, menggunakan Algoritma Dijkstra dalam penerapan di dalam sistem geografis akan menampilakan visualisasi data dalam bentuk peta , pada penampilan rute atau peta Algoritma Dijkstra lebih mudah di baca dan di pahami, sehingga penampilan Algoritma Dijkstra lebih menarik dan lebih mudah untuk membedakan dari suatu titik tertentu ke titik yang lain.

II. METODOLOGI PENELITIAN

Model Pada penelitian ini model yang digunakan adalah prototype dimana penulis melakukan beberapa tahapan :



Gambar 1. Langkah-langkah pembangunan sistem

1. Analisis Penulis menganalisa dan mendefinisikan format kebutuhan perangkat lunak, dan garis besar sistem yang akan dibangun dengan menggunakan bantuan UML ,metode objek berorientasi objek. Pada tahap ini penulis juga menggunakan teknik pengumpulan data yaitu ;
   a. Study literatur
      Penulis mendapatkan teori-teori yang dibutuhkan dari buku , jurnal dan internet sebagai referensi untuk penulisan ini.
   b. Wawancara Penulis bertanya kepada pemilik kost dan masyarakat.
2. Perancangan Berdasarkan analisis permasalahan yang ada maka dilakukan perancangan sistem GIS dengan algortima dijkstra. Perancangan sistem dalam penelitian ini berdasarkan Object Oriented Analysis and Design.
3. Prototipe Penulis membangun aplikasi dengan analisis yang telah didapat.
4. Evaluasi Sistem Penulis mengavaluasi apakah sistem yang telah dibangun sesuai dengan analisis penulis. Jika proses evaluasi telah sesuai dengan algoritma maka dilanjut ke proses implementasi, sedangkan jika evaluasi tidak sesuai maka di ulang ke proses analisis.
5. Implementasi Pada tahan ini penulis berfokus pada penerapan aplikasi pada handphone.
6. Pengujian Penulis melakukan pengujian pada setiap fungsi dan algoritma aplikasi dengan metode blackbox.

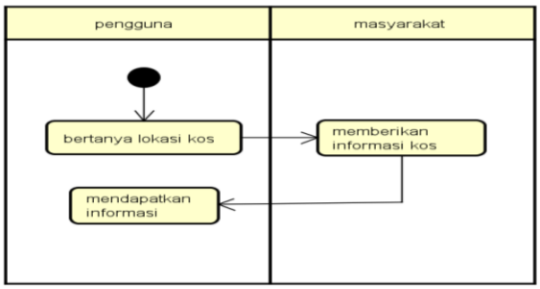
A. Metode Pengembangan Sistem Metode yang digunakan yaitu menggunakan metode berorientasi objek dimana didalam metode ini terdapat classes, methods, objects, dan message yang berkaitan dengan Sistem Informasi yang akan dibuat.

B. Alat Bantu Pengembangan Sistem Pada pembuatan aplikasi ini penulis menggunakan UML sebagai alat bantu dalam pengembangan sistem.

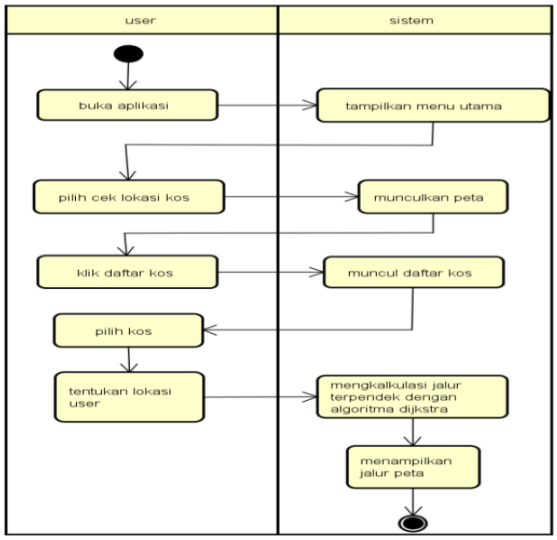
III. HASIL DAN PEMBAHASAN

A. Activity Diagram

Activity diagram digunakan untuk menggambarkan proses aktivitas yang terjadi. Berikut ini adalah beberapa proses yang digambarkan antara lain :



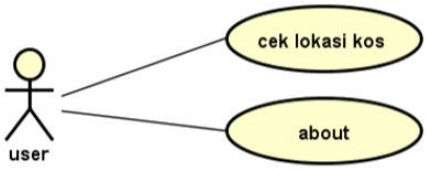
Gambar 2. Activity Diagram Pencarian Kost



Gambar 2. Activity Diagram Cek Lokasi Kost

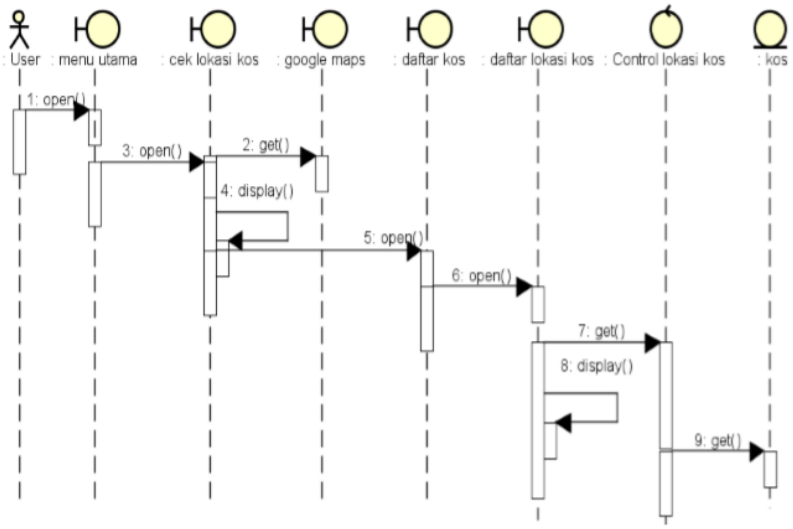
B. Use Case Diagram

Use case diagram untuk menggambarkan kebutuhan dan fungsionalitas sistem dari sudut pandang user.



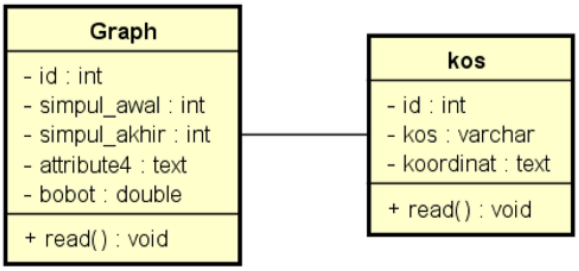
Gambar 3. Use case Sistem Usulan

C. Sequence Diagram



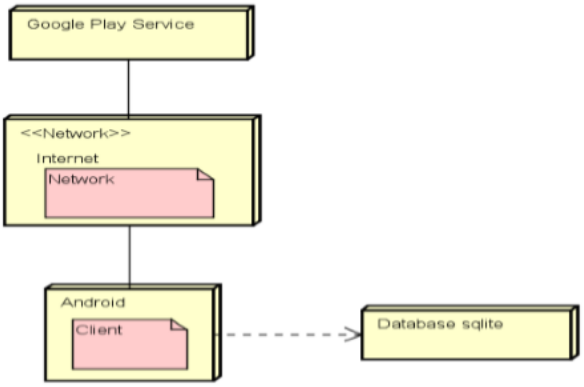
Gambar 4. Sequence Diagram Cek Lokasi Kost

D. Class Diagram



Gambar 5. Class Diagram

E. Deployment Diagram



Gambar 6. Deployment Diagram

F. Algoritma Dijkstra

Gambar 7. Flow Algoritma Dijkstra

1. Mendapatkan simpul awal dan tujuan

```
graph = arg_graph;
int simpul_awal = simpulAwal;
int simpul_maju = simpulAwal;
int simpul_tujuan = simpulTujuan;
```

2. Tandai simpul awal

```
List<Integer> simpulYangDikerjakan = new ArrayList<Integer>();

List<Integer> simpulYangSudahDikerjakan_bawah = new ArrayList<Integer>();
```

3. Perulangan mendapatkan simpul yang dikerjakan

```
for(int perulanganSimpul = 0; perulanganSimpul < simpulYangDikerjakan.size(); perulanganSimpul++){
   int jml_baris_fix = 0;
       for(int min_batas_bris = 0; min_batas_bris < 100; min_batas_bris++){
           if(graph[simpulYangDikerjakan.get(perulanganSimpul)][min_batas_bris] != null){
               jml_baris_fix += 1;
           }
       }

List<Double> bobot = new ArrayList<Double>();
int status_baris = 0;
 for(int min_batas_bris_fix = 0; min_batas_bris_fix < jml_baris_fix; min_batas_bris_fix++){
  String bobot_dan_ruas = graph[simpulYangDikerjakan.get(perulanganSimpul)][min_batas_bris_fix];
  String[] explode;
  explode = bobot_dan_ruas.split("->");
    if(explode.length == 2){
      status_baris += 1;
             if(!simpulYangSudahDikerjakan_bawah.isEmpty()){
          if(simpulYangSudahDikerjakan_bawah.contains(simpulYangDikerjakan.get(perulanganSimpul))){
             nilaiSimpulYgDitandai = 0;
          }else{
            nilaiSimpulYgDitandai = nilaiSimpulFixYgDitandai;
          }
      }
      bobot.add((Double.parseDouble(explode[1])+nilaiSimpulYgDitandai));
     graph[simpulYangDikerjakan.get(perulanganSimpul)][min_batas_bris_fix] =
     String.valueOf(explode[0]+"->"+(Double.parseDouble(explode[1])+nilaiSimpulYgDitandai));
   }
}
```

4. Perulangan mendapatkan simpul dengan bobot terkecil

```
for(int index_bobot = 0; index_bobot < bobot.size(); index_bobot++){
   if(bobot.get(index_bobot) <= bobot.get(0)){
       bobot.set(0, bobot.get(index_bobot));
   }
}
perbandinganSemuaBobot.add(bobot.get(0));
}
else{
}
 if(!simpulYangSudahDikerjakan_bawah.contains(simpulYangDikerjakan.get(perulanganSimpul))){
    simpulYangSudahDikerjakan_bawah.add(simpulYangDikerjakan.get(perulanganSimpul));
}

 int indexAwalAsli = 0;
int status_baris1 = 0;
int dapat_indexAsliBobot = 0;
int simpul_lama = 0;
for(Integer indexAsli_bobot : simpulYangDikerjakan){
    for(int baris1 = 0; baris1 < 100; baris1++){
        if(graph[simpulYangDikerjakan.get(indexAwalAsli)][baris1] != null){
            String bobot_dan_ruas1 = graph[simpulYangDikerjakan.get(indexAwalAsli)][baris1];
             String[] explode1;
            explode1 = bobot_dan_ruas1.split("->");
            if(explode1.length == 2){
              if(perbandinganSemuaBobot.get(0) == Double.parseDouble(explode1[1])){
                    dapat_indexAsliBobot = baris1;
                    simpul_lama = simpulYangDikerjakan.get(indexAwalAsli);
                    simpul_maju = Integer.parseInt(explode1[0]);
                    status_baris1 += 1;
                }
            }
        }
    indexAwalAsli++;
}
```

5. Hapus ruas yang lain

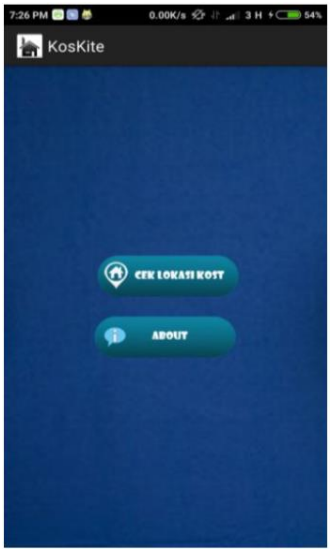
```
if(status_baris1 > 0){
    graph[simpul_lama][dapat_indexAsliBobot] = graph[simpul_lama][dapat_indexAsliBobot]+"->y";
    for(int min_kolom = 0; min_kolom < jml_simpul; min_kolom++){
        for(int min_baris = 0; min_baris < 100; min_baris++){

            if(graph[min_kolom][min_baris] != null){
                String ruasYgAkanDihapus = graph[min_kolom][min_baris];
                String[] explode3 = ruasYgAkanDihapus.split("->");
                if(explode3.length == 2){
                    if(explode3[0].equals(String.valueOf(simpul_maju))){
                        graph[min_kolom][min_baris] = graph[min_kolom][min_baris]+"->t";
                    }
                }
            }
        }
    }
}
```

6. Mendapatkan jalur terpendek

```
Collections.reverse(simpulFix_finish);
String jalur_terpendek = "";
for(int x = 0; x < simpulFix_finish.size(); x++){
    if(x == simpulFix_finish.size()-1){
        jalur_terpendek += simpulFix_finish.get(x);
    }else{
        jalur_terpendek += simpulFix_finish.get(x)+"->";
    }
}
jalur_terpendek1 = jalur_terpendek;
```

G. Implementasi

Berikut ini adalah tampilan layar menu utama aplikasi koskite yang berisi 2 menu yaitu : cek lokasi kos dan about.



Gambar 8. Tampilan menu utama

Berikut adalah tampilan layar menu cek lokasi kos yang dimana terdapat peta yang mengarah ke Pangkalpinang dan terdapat menu daftar kos.

Gambar 9. Tampilan menu cek lokasi kost

Berikut adalah tampilan layar apabila user mengklik menu daftar kos maka akan ditampilkan lis-list kos.

Gambar 10 . Tampilan Menu Daftar Kos

Berikut adalah tampilan layar apabila user telah memilih salah satu list kos dan menentukan lokasi user. Dalam hal ini juga terdapat jalur terpendek dari lokasi user dan lokasi kos berdasarkan hasil perhitungan dijkstra

Gambar 11. Tampilan Jalur Terpendek

H. Pengujian BlackBox

No	pengujian	Hasil yang diharapkan	Hasil pengujian	Benar / Salah
1	Tap icon kost	Menampilkan tampilan utama dan peta kota Pangkalpinang	Menampilkan tampilan utama dan peta kota Pangkalinang	Benar
2	Tap pada menu drop down pilih kost	Menampilkan semua daftar Kost yang telah dimasukkan dan dapat memilih kost sesuai keinginan	Menampilkan semua daftar lokasi kost dan dapat dipilih	Benar
3	Tap pada peta untuk menentukan lokasi pengguna	Menampilkan jalur terpendek yang akan dilalui untuk menuju lokasi kost tujuan	Menampilkan jalur terpendek yang akan dilalui untuk menuju lokasi kost tujuan	Benar

IV. PENUTUP

A. Kesimpulan

Penulis dapat mengambil kesimpulan bahwa aplikasi yang telah dibuat dengan menggunakan pemrograman java android dapat mengakses informasi lokasi kos yang ada di wilayah Pangkalpinang, selain itu penerapan teknologi GPS dalam aplikasi ini membuat pengguna dapat mengetahui lokasi keberadaannya. Hasil dari kesimpulan dalam menggunakan aplikasi ini adalah :

1. Aplikasi yang dibuat dengan model prototype, metode berorientasi objek, dan tools UML dapat membantu pengguna mencari rumah kost di daerah Pangkalpinang.
2. Aplikasi dapat memberi informasi kost kepada pengguna.
3. Aplikasi dapat menunjukkan jalur terpendek kepada pengguna menuju rumah kost menggunakan algoritma dijkstra.

B. Saran

Penulis menyadari dalam pembuatan aplikasi ini masih terdapat banyak kekurangan yang dapat diperbaiki maupun dilengkapi oleh peneliti atau pengembang selanjutnya.Maka penulis menyarankan hal-hal seperti :

1. Memperluas cakupan jenis item yang dipetakan sehingga aplikasi ini tidak terbatas hanya untuk menampilkan lokasi kost saja.
2. Memperluas lokasi daerah kost yang dipetakan sehingga tidak hanya terbatas pada area pangkalpinang saja.
3. Memperluas cakupan are jalur terpendek diluar area pangkalpinang 4. Aplikasi SIG ini, untuk saat ini hanya bisa dijalankan di handphone platform android, kedepannya untuk pengembang diharapkan aplikasi ini bias dijalankan di semua Sistem operasi. Menambahkan fitur agar aplikasi dapat menambahkan lokasi kost baru

DAFTAR PUSTAKA

[1] Stevia, S.S,. 2013, Perancangan Aplikasi Gis Pencarian Rute Terpendek Peta Wisata Di Kota Manado Berbasis Mobile Web Dengan Algoritma Dijkstra, Skripsi, Program Studi Teknik Informatika, Univ. Dian Nuswantoro, Semarang.

[2] Imron, F,. 2011, Penggunaan Algorimta Dijkstra dalam Pencarian Rute Tercepat dan Rute Terpendek, Skripsi, Program Studi Teknik Informatika, Univ. UIN Syarif Hidayatullah, Jakarta.

[3] Muhammad , A.W,. 2010, Perancangan Sistem Informasi Geografis Penentuan Jalur Jalan Optimum Menggunakan Metode Dijkstra Kota Yogyakarta Berbasis Web, Skripsi, Program Studi Teknik Informatika, Yogyakarta.

[4] Yogi, P,. 2015, Pencarian Rute Terpendek Menggunakan Algoritma Dijkstra Pada Sig Berbasis Web Untuk Distribusi Minuman, Skripsi, Fakultas Ilmu Komputer, STMIK Bina Nusantara Jaya Lubuk Linggau, Padang.

[5] Antonio, G,. 2013, Sistem Informasi Geografis Pariwisata Berbasis Web Dan Pencarian Jalur Terpendek Dengan Algoritma Dijkstra, Tesis, Program Magister Jurusan Teknik Elektro, Univ. Brawijaya, Malang.

[6]. Endi, R., 2015, Aplikasi Informasi Geografis Pemetaan Lokasi Dokter Praktek Umum Berbasis Android Diwilayah Kabupaten Bangka , STMIK Atma Luhur, Pangkalpinang.

[7] Prahasta, E.,2009, Sistem Informasi Geografis: Konsep-konsep Dasar (Perspektif Geodesi dan Geomatika), Penerbit Informatika, Bandung.