Seminar Nasional Inovasi Teknologi
UN PGRI Kediri, 22 Februari 2017

ISBN : 978-602-61393-0-6
e-ISSN : 2549-7952

SISTEM BANTU PENCARIAN RUMAH KONTRAKAN DENGAN JARAK TERDEKAT MENGGUNAKAN ALGORITMA FLOYD-WARSHALL

Lina Farisa Rahayu[1], Patmi Kasih[2]
[1,2] Program Studi Teknik Informatika, Fakultas Teknik, Universitas Nusantara PGRI Kediri
Jl. K.H Ahmad Dahlan No. 76 Kediri
Email: [1]farisalina5@gmail.com, [2]fatkasih@gmail.com

Abstrak *- Sistem informasi yang dikemas dalam bentuk aplikasi bantu pada smartphone (android) adalah salah satu fasilitas dengan segala kemudahan yang dijanjikan oleh teknologi saat ini. Salah satunya sistem bantu pencarian lokasi dan posisi suatu tempat/ alamat. Dengan keinginan awal untuk membantu teman yang mencari rumah kontrakan di kota kediri dengan jarak yang tidak jauh dari tempat bekerja, maka terwujud penelitian sistem pencarian rumah kontrakan dan membandingkan antara rumah satu dengan rumah lainnya dalam hal rute/ jarak terdekat dan fasilitas. Sistem dan aplikasi dibuat dengan tema penunjang keputusan, berbasis android. Sistem tidak hanya memberikan informasi jarak terdekat dengan tempat kerja maupun kampus, juga informasi mengenai harga, jumlah kamar, jenis aliran air, tersedianya perabotan dan tersedianya kamar mandi atau tidak. Penerapan algoritma floyd-warshall dalam aplikasi karena floyd-warshall termasuk dalam model pemrograman dinamis. Setiap tahap yang dihasilkan dalam proses dijadikan sebagai dasar pengambilan keputusan selanjutnya. Cara kerja algoritma adalah data setiap rumah kontrakan yang diperoleh dibentuk kedalam sebuah graf, ditransformasi kan dalam bentuk matriks dua dimensi n x n. Selanjutnya menghitung matriks yang dihasilkan dari graf dan menghitung masing-masing perbandingan jarak antar keseluruhan data. Tahap terakhir, tahap ditentukannya titik akhir yang nantinya akan dijadikan rekomendasi bagi user dengan memanfaatkan kedua matriks yang dihasilkan pada tahap sebelumnya. Dengan aplikasi ini diharapkan dapat membantu pencarian kontrakan, pencari rumah kontrakan dapat memilih kontrakan sesuai kriteria yang diinginkan.*

Kata kunci: *rumah kontrakan, jarak terdekat, Algoritma Floyd-Warshall.*

Abstract *- The need for increasing the rented house, has not yet been offset by srana information will be rented house. Then the researchers problem is how to determine the route with the shortest distance and inform the user, how to inform the data of existing facilities at the rented house to the user, how to calculate the ratio of the distance using the Floyd-Warshall algorithm. From the data obtained is then formed into a graph. Then the graph is transformed into the form of a matrix. Matrix used is a two-dimensional matrix of size nx n. The next step calculates the resulting matrix of the graph. Of the matrix will be compared to each comparison between the overall distance. The last stage is the stage it determines the endpoint that will be recommended for the user by utilizing both matrices generated in the previous stage. Conclusion of the study is obtained destination that has the shortest path is the nearest point, the formation of the path by Floyd-Warshall algorithm can help solve the case of a boarding house search with the closest distance estimation.*

ISBN : 978-602-61393-0-6
e-ISSN : 2549-7952

Keywords: *rent house, the closest distance, Floyd-Warshall algorithm*

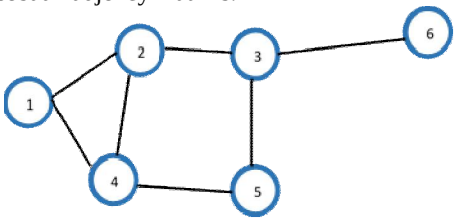
1. PENDAHULUAN

Sering kita jumpai orang yang bekerja atau melanjutkan studinya diharuskan memilih untuk berdomisili di luar daerahnya. Seseorang yang harus tinggal di kota lain karena bekerja ataupun melanjutkan studi pastinya membutuhkan tempat tinggal, dan tidak sedikit yang memutuskan untuk mencari rumah yang dapat disewa atau dikontrak sebagai tempat tinggal. Kontrakan adalah rumah yang disewakan dengan sistem kontrak yang berjangka waktu minimal satu tahun. Fasilitas ini cukup menguntungkan karena calon penghuni tidak perlu untuk membangun rumah sendiri, hanya perlu datang ke tempat orang memiliki penyewaan. Kebanyakan para pencari kontrakan harus keliling kota/daerah disekitar tempat kerjanya untuk mencari kontrakan yang sesuai dengan keinginannya. Contohnya di kota Kediri, kota dimana banyak sekali pekerja dan mahasiswa berasal dari kota lain yang bukan merupakan penduduk tetap. Banyak karyawa/ pekerja dan mahasiswa yang membutuhkan rumah sewa/ kontrakan. Untuk itu dibutuhkan informasi mengenai rumah-rumah yang disewakan beserta lokasi dan kondisi dari rumah tersebut, juga jarak serta rute dari tempat kerja ataupun tempat studi.

Mencari rumah sewa/ kontrakan dengan cara konvensional/ langsung seperti harus datang langsung ke daerah yang dituju akan memerlukan waktu yang lama untuk mendapatkan informasi tempat yang sesuai dengan keinginan calon penyewa. Disamping itu terkadang calon penyewa tidak ingin pulang pergi dalam waktu satu hari hanya untuk mencari sebuah kontrakan. Untuk itu dipandang perlu sistem yang dapat mempermudah pencarian kontrakan yang menyajikan informasi mengenai rumah-rumah yang disewakan, kondisi, lokasi, beserta jarak dan rute rumah tersebut. Selain itu sistem juga memberikan rekomendasi rumah kontrakan dengan jarak terdekat dari tempat bekerja atau tempat studi. Sistem ini dirancang untuk aplikasi berbasis android.

2. METODE PENELITIAN

Sistem bantu pencarian rumah kontrakan ini memberikan rekomendasi rumah terdekat dari beberapa rumah yang dijadikan alternatif pilihan. Algoritma yang digunakan dalam penentuan rumah dengan jarak terdekat adalah *Floyd-Warshall*. Menurut Mohamad Ray Rizaldy (2014:11): Algoritma Floyd (Algoritma Floyd-Warshall) adalah salah satu cabang dari ilmu matematika yang salah satu fungsinya adalah untuk menyelesaikan masalah lintasan terpendek. Dalam Algoritma Floyd terdapat fungsi (G = V, E) dengan G adalah graf yang merupakan kumpulan simpul (*nodes*) yang dihubungkan satu sama lain melalui sisi/busur (edges). Dengan kata lain algoritma ini mencari semua jarak node (*all pairs shortest path*) pada suatu jaringan graf.
Algoritma Floyd menggunakan matriks dua dimensi sebagai representasi dari sebuah jaringan graf. Jika suatu jaringan terdiri dari n buah *arc* maka matriks yang akan dibentuk oleh algoritma Floyd untuk proses penghitungan adalah sebesar n x n. Matriks ini merepresentasikan bobot (w) dari keseluruhan *arc* yang ada pada *graph* (S=awal, E=tujuan), dengan w(i, j) dimana i adalah node awal dan j adalah node tujuan.Dalam Algoritma Floyd-Warshall memiliki aturan untuk pembentukan matriks dari input graf berarah dan berbobot (V,E) yang berupa daftar titik (*node*/*vertex*) dan daftar sisi (*edge*). Nilai pada sebuah sisi antar node adalah bobot sisi tersebut. Sedangkan untuk node yang tidak terhubung langsung dengan node lain, memiliki bobot tak hingga. Sehigga dari graf tersebut akan membentuk sebuah adjency matriks.



Gambar 1. Contoh Graph Algoritma Floyd-Warshall dengan Vertex {1,2,3,4,5,6}dan Sisi {(1,2);(1,4);(2,4);(2,3);(3,5);(3,6);(4,5)}

Menurut Raden Aprian Diaz Noviandi (2014:11) mengenai Algoritma Floyd-Warshall adalah sebagai berikut:Algoritma Floyd-Warshall membandingkan semua kemungkinan lintasan pada graf untuk setiap

ISBN : 978-602-61393-0-6
e-ISSN : 2549-7952

sisi dari semua simpul. Menariknya, algoritma ini mampu mengerjakan proses perbandingan sebanyak V3 kali (bandingkan dengan kemungkinan jumlah sisi sebanyak V2 (kuadrat jumlah simpul) pada graf, dan setiap kombinasi sisi diujikan). Hal tersebut bisa terjadi karena adanya perkiraan pengambilan keputusan (pemilihan jalur terpendek) pada setiap tahap antara dua simpul, hingga perkiraan tersebut diketahui sebagai nilai optimal.Algoritma ini termasuk ke dalam model pemrograman dinamis. Dimana setiap tahap yang dihasilkan dalam proses dijadikan sebagai suatu dasar pengambilan keputusan selanjutnya. Berikut adalah langkah-langkah algoritma Algoritma Floyd-Warshall :

- Inisialisasikan node asal dengan node tujuan.
- Menuliskan jarak edge dari node asal ke node berikutnya.
- sMenambahkan jarak edge pada setiap node berikutnya sampai node tujuan.
- Membandingkan nilai edge pada setiap kemungkinan rute yang ditemukan.
- Memilih rute dengan nilai edge yang paling optimal (kecil).

  Misalkan terdapat suatu graf G dengan simpul-simpul V yang masing-masing bernomor 1-N (sebanyak N buah). Misalkan pula terdapat suatu fungsi shortestPath (i, j, k) yang mengembalikan kemungkinan jalur terpendek dari i ke j dengan hanya memanfaatkan simpul 1-k sebagai titik perantara.Tujuan akhir penggunaan fungsi ini adalah untuk mencari jalur terpendek dari setiap simpul 1 ke simpul j dengan perantara simpul 1 - k+1. Ada dua kemungkinan yang terjadi:
- Jalur terpendek yang sebenarnya hanya berasal dari simpul yang berada antara 1 hingga k.
- Ada sebagian jalur yang berasal dari simpul 1 hingga k+1, dan juga k+1 hingga j.

Perlu diketahui bahwa jalur terpendek dari 1 ke j yang hanya melewati simpul i-k telah didefinisikan pada fungsi shoertesPath (i, j, k) dan telah jelas bahwa jika ada solusi dari i - k+1 hingga j, maka panjang dari solusi tadi adalah jumlah dari jalur terpendek dari i - k+1 (yang melewati simpul-simpul 1-k), dan jalur terpendek dari k+1-j (juga menggunakan simpul-simpul dari 1-k).Maka dari itu, rumus untuk fungsi shortesPath (i, j, k) bisa ditulis sebagi suatu notasi rekrusif sebagai berikut:

Basis-0

shortesPath (i, j, 0) = edgeCost(i, j)

rekurens

shortesPath (i, j, k) = min(shortesPath (i, j, k-1), (shortesPath i, k, k-1) + (shortesPath k, j, k-1);

Rumus ini adalah inti dari algoritma Floyd warshall. Algoritma ini bekerja dengan menghitung shortesPath (i, j) untuk semua pasangan (i, j), dst. Proses ini akan terus berlanjut berlangsung hingga k=n dan kita telah menemukan jalur terpendek untuk semua pasangan (i, j) menggunakan simpul-simpul perantara.

3. HASIL DAN PEMBAHASAN

Dalam penelitian ini data yang digunakan diperoleh dari hasil observasi dari rumah-rumah dengan status disewakan di daerah Kota Kediri. Meliputi data seputar informasi fasilitas, harga, alamat, kontak pemilik yang tersedia. Dimana data tersebut akan digunakan sebagai informasi output pada calon penyewa. Proses selanjutnya memiliki beberapa tahap yaitu:

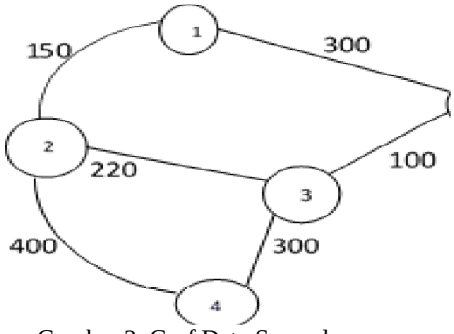
a. Capturing data Google Maps

   Tahap untuk memetakan ruas jalan dengan titik-titik rumah kontrakan yang telah didata supaya terhubung menjadi sebuah jaringan graf. Dari fitur *get-direction* Google Maps akan didapatkan estimasi jarak ruas jalan yang menghubungkan titik rumah. Sekaligus titik lokasi rumah yang terdaftar dicatat titik *longitude* dan *latitude*-nya untuk menampilkan output mapping pada user.

b. Transformasi Graf

   Tahap pembentukan graf lengkap dari hasil yang diperoleh dari tahap sebelumnya. Sejumlah node dan sisi yang telah dipetakan. Berikut sampel dari suatu denah yang dipetakan dan dibentuk dalam suatu graf. Dimana letak rumah hanya titik 4. Titik yang lain hanya sebagai titik perpotongan ruas jalan dengan nama berbeda. Ruas jalan merupakan alamat bagi state penentu lokasi kerja/studi penyewa.

ISBN : 978-602-61393-0-6
e-ISSN : 2549-7952



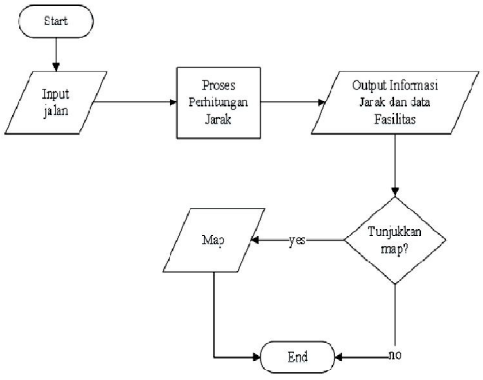
Gambar 2. Graf Data Sampel

c. Transformasi Matriks
Tahap pembentukan matriks sejumlah dari node x node sesuai dengan jaringan graf ang telah dibentuk. Berikut transformasi dari graf data sampel.

Tabel 1. Matriks Data Sampel

ID	0	1	2	3	4
0	0	300	∞	100	∞
1	300	0	150	∞	∞
2	∞	150	0	220	400
3	100	∞	220	0	300
4	∞	∞	400	300	0

Diagram berikut merupakan sebuah alur yang dirancang dalam pembuatan aplikasi. Dimana keseluruhan data dan implementasi penghitungan Algoritma Floyd akan dijalankan.



Gambar 3. Diagram Sistem

Matriks yang dihasilkan akan dihitung ke dalam Algoritma Floyd-Warshall. Dari matriks tersebut akan dibandingkan masing-masing perbandingan jarak antar keseluruhan titik. Jadi ketika menjalankan Algoritma Floyd, maka data yang dibutuhkan adalah data yang bersifat matang/ telah melalui beberapa proses pengubahan. Seperti model rekursif pada Floyd umumnya, berikut model algoritma yang digunakan:

Algoritma 1. Algoritma Floyd-Warshall
[Kegunaan Algoritma: menghasilkan matriks bobot perbandingan jarak masing-masing titik dan matriks jalur untuk rekonstruksi jalur nantinya]

```
1. Jalankan Floyd( int M[][]);
   //berisi matriks sesuai data
   yang diperoleh
2. Inisialisasi int z, int Jl[][]
   = {0, 0, 0, 0, 0,1, 1, 1, 1, 1,
   dst}; //mengikuti indeksnya dan
   panjang matriks sesuai M[][]
3. Create method Floyd(int
   adj[][]){Inisialisasi int n[] =
   adj.length;
   Inisialisasi int m[][] =
   [n][n];
   Jalankan Copy(m, adj[][])
   for (int k = 0; k < n; k++)
       for (int i = 0; i < n; i++)
   for (int j = 0; j < n; j++)
   z = m[i][k] + m[k][j];
   if(z < m[i][j])
       m[i][j] = z;
       Jl[i][j] = k;
   else
       m[i][j] = m[i][j];
     return m;
4. Method Copy(int a, int b)
   for (int i = 0; i < a.length;
   i++)
   for (int j = 0; j < a.length;
   j++)
   a[i][j] = b[i][j];
```

Algoritma 1 di atas mempunyai penjabaran sebagai berikut:

- Pada algoritma no.1 merupakan proses dijalankan matriks utama ke dalam sebuah fungsi bernama Floyd();
- Pada proses kedua adalah inisialisasi z sebagai variabel pembantu dan sebuah array Jl[][] sebagai matriks yang menghasilkan matriks pembentukan jalur nantinya.
- Proses proses ketiga, fungsi Floyd() berisi tahap dengan menjalankan fungsi Copy(). Dimana array utama disalin isinya ke dalam array m[][] melalui fungsi Copy(). Kemudian terdapat perulangan bertumpuk sebanyak tiga kali.

ISBN : 978-602-61393-0-6
e-ISSN : 2549-7952

- Perulangan dengan inisialisasi k, berfungsi sebagai pengecekan jalur intermediet. Sekaligus sebagai indeks perulangan dan batas perulangan sejumlah n kali sejumlah banyaknya node.

- Sedangkan perulangan j dan i untuk pengecekan ada atau tidaknya jalur langsung. Pengecekan dilakukan dengan menjumlahkannya pada sebuah variabel z = m[i][k] + m[k][j]. Jika hasi z < m[i][j] maka m[i][j] harus diganti dengan z. Kemudian nilai array Jl[i][j] = k.

- Pembentukan dengan perulangan di atas secara otomatis membentuk dua buah array, yaitu array perbandingan bobot jarak dan array pembentuk jalur.

Algoritma 2. Pembentukan Jalur dan Skoring
[Kegunaan algoritma: mencari perbandingan masing-masing jarak ke titik tujuan dan mencari nilai terkecil]

```
1. Inisialisasi int start, finish[];
2. Jalankan method minimumDistance[](start, finish[])
3. Fungsi minimumDistance(int a, int b[])
   for(int x = 0; x < tujuan.length; x++)
            jalankan path(a,b[x])
   Vector v = new Vector();
   for (int i = 0; i < jalur.length; i++)
             cetak jalur[i]
       for (int i = jalur.length-1; i >= 0 ; i--)
   int j = jalur[i];
   if(j >= 0)
   cetak(j + " ");
   v.addElement(j);
   int total = 0;
   for (int i = 0; i < v.size()-1; i++) {
   int j = (int)v.elementAt(i);
   int k = (int)v.elementAt(i+1);
   int dist = l[j][k];
   cetak("distance = " + dist);
   total += dist;
   tmp[x] = total;
4. Fungsi path(int awal, int tujuan)
   if( t == 0)
   jalur[t++] = tujuan;
   int nilai = via[asal][tujuan];
       if(nilai != asal)
   jalur[t++] = nilai;
   path(asal, nilai);
   else
   jalur[t] = asal;
5. Mengambil nilai terkecil
   min = tmp;
   int pou[];
   int dekat = 10000;
     //memilih, mengabalikan nilai terkecil
   for(int m = 0; m < min.length; m ++)
   for(int u = 0; u < min.length; u++)
   if(min[u] <= dekat)
   dekat = min[u];
   if(min[m] == dekat)
   pou = m;
   r[0] = b[pou];
   r[1] = min[pou];
```

Rekonstruksi Jalur dan penentuan titik terdekat, pada algoritma di atas akan dijelaskan sebagai berikut:

- Menginisialisasi titik awal dan beberapa titik tujuan
- Memproses titk awal dan beberapa titik tujuan ke dalam fungsi minimumDistance().
- Fungsi minimumDistance(), adalah sebuah fungsi yang di dalamnya menjalankan fungsi path.yaitu sebuah fungsi rekonstruksi jalur. Dengan pola seperti berikut, misalkan mencari jalur dari 0 ke 4, maka diambil data dari matriks Jalur dimana 0 dan 4 dijadikan indeks i dan j. Jalur[0, 4] = 3, Jalur[0, 3] = 0. Jika nilai pada matriks telah sesuai dengan titik awal maka pembentukan rute selesai.
- Sedangkan pada fungsi yang menggunakan elemen Vector(), digunakan untuk mempermudah menulis ulang sekaligus menampung nilai bobot dari rute yang dibentuk. Yang ditampung pada array total. Proses tersebut dilakukan berulang sebanyak panjangnya array tujuan.
- Sedangkan pada proses paling akhir, adalah menyalin keseluruhan array total kemudian membandingkan nilai mana yang paling kecil. Dan mengambalikan, nilai yang paling kecil tersebut kepada titik tujuan yang memiliki jarak terebut.

Berikut langkah penyelesaian dari data sampel, inisialisasi titik awal dan tujuan yang akan dicari:

ISBN : 978-602-61393-0-6
e-ISSN : 2549-7952

titik awal : 0
titik tujuan : [2, 4]
proses titik awal =0 menuju tujuan ke-1→2:
backprocess : 2 3 0 -1 -1
hasil rute : 0 →3 →2 : distance(0, 3) = 100 + distance(3, 2) = 220
jarak total = 320
proese titik awal: 0 menuju tujuan ke-2→4:
backprocess : 4 3 0 -1 -1
hasil rute : 0 →3 →4 : distance(0, 3) = 100 + distance(3, 4) = 300
jarak total = 400
hasil perbandingan jarak masing-masing tujuan :[320, 400]

Jarak Terdekat : 320 dan letak titik tujuan yang direkomendasikan adalah titik 2.

Dengan implementasi di atas maka akan di dapatkan perbandingan masing-masing jarak terdekat antara tempat yang ditentukan user dengan masing-masing tempat kontrakan, sehingga mampu didapatkan rumah dengan jarak terdekat.

Berdasarkan rancangan dan implementasi sistem yang dilakukan, diperoleh hasil aplikasi yang memunuhi tujuan yang diharapkan.

Gambar 4. Tampilan Awal Aplikasi

Pada awal menu aplikasi akan menampilkan menu pencarian yang dapat digunakan.

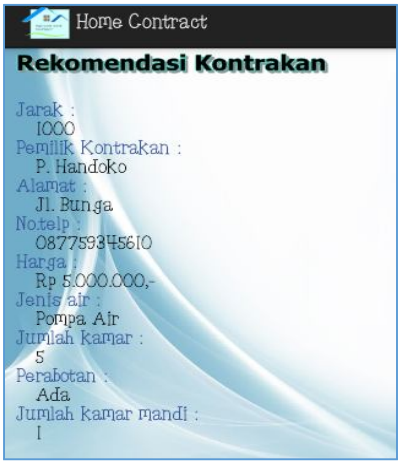
Gambar 5. Menu Yang Disediakan

Selanjutnya daftar nama-nama jalan yang terdaftar pada database dan terdaftar di daerah kota Kediri, gambar 6.
Selanjutnya user dapat memilih satu satu nama jalan atau lebih detail masuk pada menu pencarian. Pengguna dapat memilih jalan yang tersedia di menu drop down pada aplikasi. Setelah itu pilih nama jalan yang dikehendaki, selanjutnya klik button “mencari” untuk melakukan proses pencarian pada aplikasi. Maka aplikasi akan memproses jalan yang dipilih oleh pengguna dengan menampilkan data rumah kontrakan yang terdekat dengan jalan tersebut. Untuk hasil yang tidak sesuai dengan harapan dari user sebagai pencari rumah kontrakan, maka user dapat melakukan pencarian ulang dengan data nama jalan yang lain.

ISBN : 978-602-61393-0-6
e-ISSN : 2549-7952

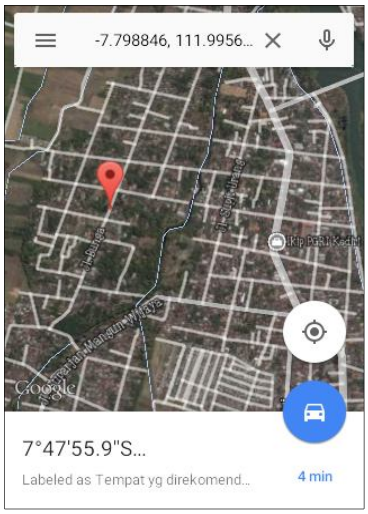
Gambar 6 Halaman Daftar Nama Jalan dan Pencarian

Hasil yang ditampilkan adalah yang mempunyai nilai kemiripan dengan kriteria yang diberikan oleh user sebagai pencari rumah kontrakan.



Gambar 7. Contoh Hasil Rekomendasi Aplikasi

Menu lain yang disediakan oleh aplikasi diantaranya adalah menunjukkan lokasi dan rute rumah kontrakan yang direkomendasikan.



Gambar 7. Tampilan Map Lokasi.

4. SARAN

Aplikasi pencarian kontrakan dengan jarak terpendek ini masih meliputi Kediri bagian kota. Maka perlu pengembangan lebih lanjut Data yang digunakan masih statis sehingga belum terdapat fitur penambahan data.

DAFTAR PUSTAKA

[1] Aprian Raden, D.2007.*Perbandingan Algoritma Djikstra dan Algoritma Floyd dalam Penentuan Lintasan Terpendek*. (Online), tersedia: http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/20062007/Makalah_2007/MakalahSTMIK2007-021.pdf, diunduh 14 Januari 2014.

[2] Ardiansyah Irfan. & Hakim K. Dimara.2012.*Rancang Bangun Aplikasi untuk Menentukan Jalur Terpendek Menggunakan Algoritma Floyd di lokasi Wisata Purbalingga*. (Online), tersedia: http://ojs.unud.ac.id, diunduh 6 Desember 2013.

[3] Iftadi Irwan, dkk.2011.*Perancangan Peta Evakuasi Menggunakan Algoritma Floyd-Warshall untuk Penentuan Lintasan Terpendek*. (Online), tersedia: http://eprints.uns.ac.id/1419/1/4_10_2_IRF_WAJ_BN_P95_P104.pdf, diunduh 27 Desember 2013.

[4] K Emt. 2013. *Floyd Algorithm.* (online). tersedia: http://www.youtube.com/watch?v=odeFemb3o-o, diunduh 25 Oktober 2014

[5] Lecture 15: *The Floyd-Warshall Agorithm*. (Online) tersedia: http://www.cse.ust.hk/faculty/golin/COMP271Sp03/Notes/MyL15.pdf, diunduh 02 Januari 2014.

[6] Lecture 24: *Floyd-Warshall Algorithm*. (Online) tersedia: http://cis.k.hosei.ac.jp/~rhuang/Miccl/Algorithm3/lect24-floyd_warshall.pdf, diunduh 02 Januari 2014.

[7] Puntambekar A.A. 2008. *Analysis and Design of Algorithms*. India: Technical Publications Pune.

[8] Rivera Jeff. 2012. *Floyd de Algorithm*. (online). tersedia: http://www.youtube.

ISBN : 978-602-61393-0-6
e-ISSN : 2549-7952

com/watch?v=DfgaBkp02HY, diunduh 25 Oktober 2014.

[9] Rizaldy M. Ray. 2007. *Pencarian Jalur Terpendek Dalam GPS dengan Menggunakan Teori Graf*. (online), tersedia:if15073@students. informatika.org, diuduh 5 Oktober 2014.

[10] Sani F. Ajeng, dkk.2013,*Algoritma Floyd Warshall untuk Menentukan Jalur Terpendek Evakuasi Tsunami di Kelurahan Sanur*. (Online), tersedia: http://ojs.unud.ac.id/index.php/mtk/article/download/4910/3696, diunduh 19 Desember 2013.

[11] Suprianto Dodit dan Agustina Rini, S.Kom, M.Pd. 2012. *Pemrograman Aplikasi Android*. Yogyakarta: Mediakom