SISTEM PENDUKUNG KEPUTUSAN PENENTUAN RUTE TERDEKAT PADA OPERATOR TAXI UNTUK MENCARI TEMPAT TUJUAN PENUMPANG MENGGUNAKAN ALGORITMA FLOYD-WARSHALL DI DHOHO TAXI KEDIRI

SKRIPSI

Diajukan Untuk Memenuhi Sebagian Syarat Guna Memperoleh Gelar
Sarjana Komputer (S.Kom) Pada Program Studi Teknik Informatika
Fakultas Teknik Universitas Nusantara PGRI Kediri

OLEH:

M. KHOIRUL ANWAR
NPM : 11.1.03.02.0240

FAKULTAS TEKNIK (FT)
UNIVERSITAS NUSANTARA PERSATUAN GURU REPUBLIK INDONESIA
UNP KEDIRI
2016

Skripsi oleh :

M. KHOIRUL ANWAR

NPM: 11.1.03.02.0240

Judul:

SISTEM PENDUKUNG KEPUTUSAN PENENTUAN RUTE TERDEKAT PADA OPERATOR TAXI UNTUK MENCARI TEMPAT TUJUAN PENUMPANG MENGGUNAKAN ALGORITMA FLOYD-WARSHALL

Telah disetujui untuk diajukan
Kepada Panitia Ujian/Sidang Skripsi
Program Studi Teknik Informatika FT UN PGRI Kediri

Tanggal : 16 Agustus 2016

Pembimbing I	Pembimbing II
ARDI SANJAYA, M.Kom.	**FAJAR ROHMAN HARIRI, M.Kom.**
NIDN : 0706118101	**NIDN : 0715058902**

ii

Skripsi Oleh:

M. KHOIRUL ANWAR
NPM: 11.1.03.02.0240

Judul:

SISTEM PENDUKUNG KEPUTUSAN PENENTUAN RUTE TERDEKAT PADA OPERATOR TAXI UNTUK MENCARI TEMPAT TUJUAN PENUMPANG MENGGUNAKAN ALGORITMA FLOYD-WARSHALL

Telah dipertahankan di depan Panitia Ujian/Sidang Skripsi
Program Studi Teknik Informatika FT UN PGRI Kediri
Pada Tanggal :

Dan Dinyatakan Telah Memenuhi Persyaratan

Panitia Penguji:

1. Ketua : (Ardi Sanjaya, M.Kom)
2. Penguji I : (Resty Wulanningrum, M.Kom)
3. Penguji II : (Fajar Rohman Hariri, M.Kom)

Mengetahui,
Dekan FT

Dr. Suryo Widodo, M.Pd.
NIP.19640202 199103 1 002

iii

SISTEM PENDUKUNG KEPUTUSAN PENENTUAN RUTE TERDEKAT PADA OPERATOR TAXI UNTUK MENCARI TEMPAT TUJUAN PENUMPANG MENGGUNAKAN ALGORITMA FLOYD-WARSHALL DI DHOHO TAXI KEDIRI

M. KHOIRUL ANWAR

NPM : 11.1.03.02.0240

Teknik – Teknik Informatika
konan0827@gmail.com
ADI SANJAYA, M.Kom. dan **FAJAR ROHMAN HARIRI, M.Kom.**

UNIVERSITAS NUSANTARA PGRI KEDIRI

ABSTRAK

Penelitian ini dilatar belakangi hasil pengamatan dan pengalaman peneliti, bahwa proses perjalanan mengatarkan penumpang *taxi* masih belum optimal. Serta pemasangan argometer *taxi* masih bisa diubah oleh oknum tertentu tanpa sepengetahuan pihak lain mengakibatkan biaya transportasi berlebih yang membuat penumpang *taxi* tudak puas.

Permasalahan penelitian ini adalah (1) Bagaimana cara menentukan rute terdekat sesuai dengan tujuan yang diinginkan penumpang taksi? (2) Bagaimana cara menentukan jarak terdekat dengan menggunakan *Alogaritma Floyd-Wharshal* untuk mencapai lokasi yang dituju?

Penelitian ini menggunakan pendekatan Penelitian Rekayasa Teknologi Informasi dengan subjek penentuan rute terdekat dan perhitungan argo jarak. Penelitian ini dilaksanakan dengan tuju siklus, menggunakan observasi pada pihak PT Dhoho *Taxi*, analisis data, perancangan sistem, pembuatan sistem, pembuatan program evaluasi program dan penyusunan laporan

Kesimpulan hasil penelitian ini adalah (1) Peneliti berhasil merancang sistem pendukung keputusan penentu rute terdekat tujuan penumpang menggunakan algoritma *Floyd-Warshall*. (2) Aplikasi dapat menentukan rute terdekat sesuai antara titik awal dengan titik tujuan serta dapat menghitung perhitungan argometer jarak.

Berdasarkan simpulan hasil penelitian ini drekomendasikan: (1) Kinerja aplikasi terlalu berat dan memerlukan koneksi yang baik untuk menjalankanya. (2) Perlu penambahan pada argometer dalam perhitungan waktu tempuh.

Kata kunci : rute terdekat, algoritma *floyd-warshall*, taxi, android

I. LATAR BELAKANG

Saat ini banyak perkembangan teknologi yang maju pesat, salah satunya dibidang perangkat elektroink khususna dibidang *mobile*. Perangkat *mobile* telah mendominasi kehidupan manusia dengan segala macam fasilitas yang ditawarkan. Salah satunya adalah pada bidang teknologi komunikasi yang dinilai paling cepat perkembangannya. Perusahaan penyedia layanan komunikasi, menyediakan banyak teknologi guna mendukung hubungan komunikasi seperti telepon seluler.

Android merupakan salah satu tipe perangkat *mobile*. *Android* memiliki tujuan utama untuk memajukan inovasi piranti telepon bergerak agar pengguna mampu mengeksplorasi kemampuan dan menambah pengalaman lebih dibandingkan dengan *platform mobile* lainnya. Hingga saat ini Android terus berkembang, baik secara sistem maupun aplikasinya.

GPS (*Global Positioning System*) yaitu teknologi *navigasi* yang memanfaatkan satelit. Dengan menggunakan fasilitas GPS ini pengguna ponsel *Android* akan mendapatkan informasi posisi dan waktu dengan akurasi yang sangat tinggi. Fasilitas GPS ini dapat juga untuk menentukan rute terpendek ketempat yang ingin dituju, dengan menggunakan Algoritma *Floy-Warshall* untuk menentukan rute terpendek.

Jasa *taxi* merupakan suatu jasa mengantarkan penumpang ketempat tujuan, penggunaan jasa taksi ini bertujuan untuk mengurangi jumlah kendaraan di jalan raya yang mengakibatkan kemacetan, serta mengantar penumpang *taxi* ketempat tujuan dengan cepat dan aman. Dengan penggunaan Alogaritma *Floid-Warshall* pada pemanfaatan teknologi GPS pada ponsel berbasis android yang digunakan untuk menentukan rute terpendek pada penggunaan jasa transportasi *taxi* dapat menampilkan rute terdekat untuk mencapai tempat yang inginkan.

PT Dhoho Taxi Kediri adalah *agent taxi* yang terletak di jalan Mauni No. 75, Kelurahan Bangsal, Kecamatan Pesantren, Kota Kediri dan beroprasi di wilayah kota kediri dan skitarnya. PT Dhoho Taxi Kediri berdiri sejak tahun 2013 dan sampai sekarang terus beroprasi sebagai jasa trasportai darat diwilayah kota Kediri dan sekitarnya.

Sistem kerja yang dijalankan Dhoho Taxi Kediri masih menggunakan dua alat yang dipasang pada mobil *taxi* yaitu GPS unutk memantau kinerja *taxi* dan argometer untuk menghitung biaya transportasi, tentunya hal itu masih belum bisa membantu kinerja taxi secara optimal.

Berdasarkan pemaparan di atas, penulis tertarik untuk merancang sebuah Sistem Pendukung Keputusan untuk menentukan suatu rute terdekat sebagai bahan untuk penelitian skripsi dengan judul yaitu ”Sistem Pendukung Keputusan Penentuan Rute Terdekat Pada Operator Taksi Untuk Mencari Tempat Tujuan Penumpang Menggunakan Algoritma *Floyd-Warshall*”.

Menurut jurnal yang berjudul *Perbandingan Algoritma Dijkstra dan Algoritma Floyd Warshall dalam penentuan Lintasan Terpendek*, Raden Aprian D. N (2007), algoritma *Floyd-Warshall* menerapkan pemrograman dinamis yang lebih menjamin penentuan solusi optimum untuk kasus penentuan rute terpendek.

Sistem ini tidak hanya memberikan informasi seputar rute-rute terdekat untuk menuju tempat tujuan, tetapi dengan ditambahkannya perhitungan argometer didalamnya juga dapat menampikan total harga yang harus dibayar setelah mencapai tujuan dengan memasukkan *digid* jarak yang telah dilalui, sehingga depat mempermuah proses transaksi antara sopir *taxi* dan penumpang, serta penambahan aplikasi *dasboard* sebagai *server* pusat yang berfungsi mantau keberadaan taksi yang nggunakan aplikasi tersebut sehingga dapat memantau keberadaan *taxi* tersebut.

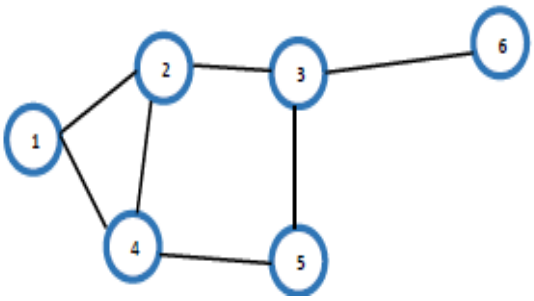
II. Floyd Warshall

Algoritma Floyd-Warshall ditemukan oleh Stephen Warshall dan Robert W. Floyd. Stephen Warshall lahir di New York pada tahun 1935 dan meninggal pada tanggal 11 Desember 2006. Robert W. Floyd, lahir di New York pada 8 Juni 1936 dan meninggal pada tanggal 25 September2001.

Algoritma Floyd (Algoritma Floyd-Warshall) adalah salah satu cabang dari ilmu matematika yang salah satu fungsinya adalah untuk menyelesaikan masalah lintasan terpendek (Sani. 2003).

Algoritma floyd menggunakan matriks dua dimensi sebagai representasi dari sebuah jaringan. 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 terdapat fungsi (G=V,E) dengan G = graf yang merupakan kumpulan simpul (nodes) yang dihubungkan satu sama lain melalui sisi/busur (edges). Suatu graf G terdiri dari dua himpunan yaitu himpunan V dan himpunan E.



Gambar 2.2 : Contoh *Graf Alogaritma Floyd-Wrshall*

Dari contoh gambar diatas dapat diketahui sisi yang dilalui {(1,2);(1,4);(2,4);(2,3);(3,5);(3,6);(4,5)} dan vertex yang dilewati {1,2,3,4,5,6}. Dari kedua keterangan tersebut diubah menjasi matriks (nxn) yang sejumlah dengan vertex. Setelah terbentuk matriks akan dilanjutkan dengan perhitungan perulangan untuk mencari jarak terdekat dari rute tersebut.

Berikut adalah contoh tabel matrikx dari contoh gambar diatas :

Tabel 2.1 : Contoh Tabel Matriks

	1	2	3	4	5	6
1	0	(1,2)	∞	∞	∞	∞
2	(1,2)	0	(2,3)	(2,4)	∞	∞
3	∞	(2,3)	0	∞	(3,5)	(3,6)
4	∞	(2,4)	∞	0	(4,5)	∞
5	∞	∞	(3,5)	(4,5)	0	∞
6	(1,6)	∞	(3,6)	∞	∞	0

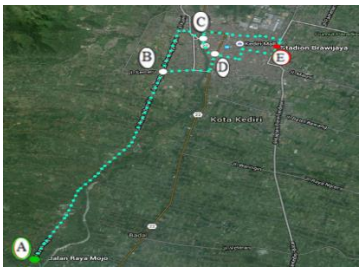
Simbol takhingga (∞) adalah jarak yang dicari untuk menentukan rute terdekat yang dilalui untuk menuju titik tujuan dengan menggunakan rumus perulangan berikut :

S(r) + E(r) < S(E) maka [S(E) = S(r) + E(r)]

III. HASIL DAN KESIMPULAN

A. Hasil Perhitungan

1. Pemilihan rute



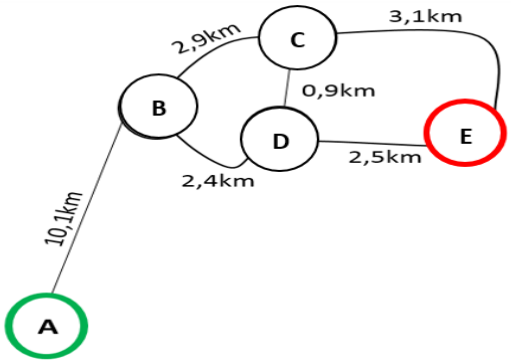
Gambar 2.3 : Peta Lokasi

Keterangan :

a. (A) merupakan titik awal (Jalan Raya Mojo).
b. (B) merupakan tanda jalan (Jalan Penanggungan).
c. (C) merupakan tanda jalan (Jalan Jembatan Lama).
d. (D) merupakan tanda jalan.(Jalan Panglima Besar Sudirman).
e. (E) merupakan titik tujuan (Stadion Braijaya).
f. Garis bertitik biru adalah jalan yang dilalui.

2. Langkah Pertama

Langkah pertama kita transformasikan ke bentuk graph seperti berikut :



Gambar 2.4 : Graph Hasil Transdormasi dari Rute Kote Kediri

Lalu ubah dalam bentuk matriks (nxn) sejumlah node, dengan ketentuan jika node tidak terhubung langsung maka diisi dengan simbol takhingga (∞).

Tabel 2.2 : Transformsi Bentuk Graph Kedalam Bentu Matriks

	A	B	C	D	E	
A	0	10,1	∞	∞	∞	1
B	10,1	0	2,9	2,4	∞	2
C	∞	2,9	0	0,9	3,1	3
D	∞	2,4	0,9	0	2,5	4
E	∞	∞	3,1	2,5	0	5
	1	2	3	4	5	

Keterangan Tabel :

a) Simpul A, B, C, D, E dengan warna hijau merupakan node dengan titik awal.
b) Simpul A, B, C, D, E dengan warna merah merupakan node dengan titik tujuan.
c) Angka 1,2,3,4,5 merupakan angka penunjuk indeks proses, dimana warna dari indeks menyesuaikan warna daripada simpul yang bersesuaian.
d) Rumus dalam perulangan :
S(r) + E(r) < S(E) maka [S(E) = S(r) + E(r)]
e) Hasil daripada iterasi sebelumnya menjadi matriks iterasi tahap selanjutnya.

3. Langkah Kedua

a. Iterasi pertama r0=r1

Tabel 2.3 : Iterasi Ke Satu

	A	B	C	D	E	
A	0	10,1	∞	∞	∞	1
B	10,1	0	2,9	2,4	∞	
C	∞	2,9	0	0,9	3,1	
D	∞	2,4	0,9	0	2,5	
E	∞	∞	3,1	2,5	0	
	1					

S = {A(r) = 0 ; B(r) = 10,1 ; C(r) = ∞ ; D(r) = ∞ ; E(r) = ∞}

E = {A(r) = 0 ; B(r) = 10,1 ; C(r) = ∞ ; D(r) = ∞ ; E(r) = ∞}

S(r) + E(r) < S(E) maka [S(E) = S(r) + E(r)]

b. Iterasi r1=r2

Tabel 2.4 : Iterasi Ke Dua

	A	B	C	D	E	
A	0	10,1	∞	∞	∞	
B	10,1	0	2,9	2,4	∞	2
C	∞	2,9	0	0,9	3,1	
D	∞	2,4	0,9	0	2,5	
E	∞	∞	3,1	2,5	0	
		2				

S = {A(r) = 10,1 ; B(r) = 0 ; C(r) = 2,9 ; D(r) = 2,4 ; E(r) = ∞ }

E = {A(r) = 10,1 ; B(r) = 0 ; C(r) = 2,9 ; D(r) = 2,4 ; E(r) = ∞ }

S(r) + E(r) < S(E) maka [S(E) = S(r) + E(r)]

c. Interasi r2=r3

Tabel 2.5 : Iterasi Ke Tiga

	A	B	C	D	E	
A	0	10,1	13	12,5	∞	
B	10,1	0	2,9	2,4	∞	
C	13	2,9	0	0,9	3,1	3
D	12,5	2,4	0,9	0	2,5	
E	∞	∞	3,1	2,5	0	
			3			

S = {A(r) = 13 ; B(r) = 2,9 ; C(r) = 0 ; D(r) = 0,9 ; E(r) = 3,1 }

E = {A(r) = 13 ; B(r) = 2,9 ; C(r) = 0 ; D(r) = 0,9 ; E(r) = 3,1 }

S(r) + E(r) < S(E) maka [S(E) = S(r) + E(r)]

d. Interasi ketiga r3=r4

Tabel 2.6 : Iterasi Ke Empat

	A	B	C	D	E	
A	0	10,1	13	12,5	16,2	
B	10,1	0	2,9	2,4	6	
C	13	2,9	0	0,9	3,1	
D	12,5	2,4	0,9	0	2,5	4
E	16,1	6	3,1	2,5	0	
				4		

S = {A(r) = 12,5 ; B(r) = 2,4 ; C(r) = 0,9 ; D(r) = 0 ; E(r) = 2,5 }

E = {A(r) = 12,5 ; B(r) = 2,4 ; C(r) = 0,9 ; D(r) = 0 ; E(r) = 2,5 }

S(r) + E(r) < S(E) maka [S(E) = S(r) + E(r)]

e. Interasi ketiga r4=r5

Tabel 2.7 : Iterasi Ke Lima

	A	B	C	D	E	
A	0	10,1	13	12,5	15	
B	10,1	0	2,9	2,4	4,9	
C	13	2,9	0	0,9	3,1	
D	12,5	2,4	0,9	0	2,5	
E	15	4,9	3,1	2,5	0	5
					5	

S = {A(r) = 15 ; B(r) = 4,9 ; C(r) = 3,1; D(r) = 2,5 ; E(r) = 0 }

E = {A(r) = 15 ; B(r) = 4,9 ; C(r) = 3,1; D(r) = 2,5 ; E(r) = 0 }

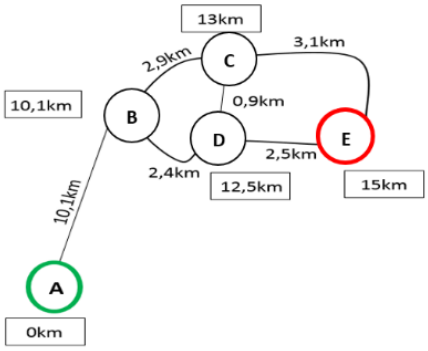
S(r) + E(r) < S(E) maka [S(E) = S(r) + E(r)]

f. Matriks Hasil

Tabel 2.8 : Matriks Hasil

	A	B	C	D	E	
A	0	10,1	13	12,5	15	1
B	10,1	0	2,9	2,4	4,9	2
C	13	2,9	0	0,9	3,1	3
D	12,5	2,4	0,9	0	2,5	4
E	15	4,9	3,1	2,5	0	5
	1	2	3	4	5	

Perhitungan perbandingan jarak :



Gambar 2.5 : Graph Dengan Bobot Masing-Masing Sampel

Dari gambar diatas diketahui jarak antara titik A dan masing-masing titik. Dari titik semula A menuju ketitik E dapat dilalui dengan 4 rute berikut adalah uraianya :

Dari A menuju E terdapat 4 rute :

a. Melalui rute A, B, C, E

E ➔ C = 15 – 3,1 = 11,9

C ➔ B = 13 – 2,9 = 10,1

B ➔ A = 10,1 – 10,1 = 0 (selesai)

AE = AB + BC + CE

= 10,1 + 2,9 + 3,1

= 16,1 km

b. Melalui rute A, B, C, D, E

E ➔ D = 15 – 2,5 = 12,5

D ➔ C = 12,5 – 0,9 = 11,6

C ➔ B = 13 – 2,9 =10,1

B ➔ A = 10,1 – 10,1 = 0 (selesai)

AE = AB + BC + CD + DE

= 10,1 + 2,9 + 0,9 + 2,5

= 16,4

c. Melalui rute A, B, D, C, E

E ➔ C = 15 – 3,1 = 11,9

C ➔ D = 13 – 0,9 = 12,1

D ➔ B = 12,5 – 2,4 = 10,1

B ➔ A = 10,1 – 10,1 = 0 (selesai)

AE = AB + BD + DC + CE

= 10,1 + 2,4 + 0,9 + 3,1

= 16,1 km

d. Melalui rute A, B, D, E

E ➔ D = 15 – 2,5 = 11,9 (dilalui)

D ➔ B = 12,5 – 2,4 = 10,1

B ➔ A = 10,1 – 10,1 = 0 (selesai)

AE = AB + BD + DE

= 10,1 + 2,4 + 2,5

= 15 km

Tabel 2.9 : Tabel Rangking Jalur Terdekat

No	Titik awal-tujuan	Rute	Jarak (km)	Rangking
1	A – E	A-B-D-E	15	1
2	A – E	A-B-C-E	16,1	2
3	A – E	A-B-D-C-E	16,1	3
4	A – E	A-B-C-D-E	16,4	4

Jadi dari perhitungan penentuan rute terdekat menggunakan alogaritma floyd warshal diatas, ditemukan rute terdekat dari A (jalan raya mojo) yang merupakan tempat semula menuju E (stadion brawijaya) tempat tujuan melalui rute A, B, D, E dengan jarak tempuh 15 km.

Setelah rute terekat diketahui jarak A ke E dengan jarak 15 km maka dilakukan perhitungan harga untuk jarak tersebut, berikut ini adalah perhitungan untuk menghitung jumlah harga pada jarak tersebut:

Diketahui :

a. Tarif awal sampai satu kilometer pertama = Rp 6.000,-
b. Tarif setelah satu kilometer pertama = Rp 3.000,-
c. Tarif minimal = Rp 30.000,-
d. Tarif bukapintu = 2 x 3000 (tarif akan tetap selama pengguna 1 orang).

Berikut adalah perhitungan jarak pada argometer :

a. Rute A-B-D-E dengan jarak 15 km

AE = (total jarak – 1 km pertma) x 3000 + 6000

= (15 – 1) x 3000 + 6000

= Rp 48.000,-

b. Rute A-B-C-E dengan jarak 16,1 km

AE = (total jarak – 1 km pertma) x 3000 + 6000

= (16,1 – 1) x 3000 + 6000

= Rp 51.300,-

c. Rute A-B-D-C-E dengan jarak 16,1 km

AE = (total jarak – 1 km pertma) x 3000 + 6000

= (16,1 – 1) x 3000 + 6000

= Rp 51.300,-

d. Rute A-B-C-D-E dengan jarak 16,4 km

AE = (total jarak – 1 km pertma) x 3000 + 6000

= (16,4 – 1) x 3000 + 6000

= Rp 52.200,-

Tabel 2.10 : Tabel Rangking Jalur Terdekat Dengan Tarif

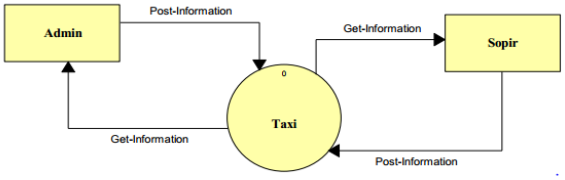
No	Rute	Jarak(km)	Tarif	Rangking
1	A-B-D-E	15	Rp 48.000,-	1
2	A-B-C-E	16,1	Rp 51.000,-	2
3	A-B-D-C-E	16,1	Rp 51.000,-	3
4	A-B-C-D-E	16,4	Rp 52.200,-	4

Dari tabel diatas dijelaskan tentang perengkingan tarif taksi dan jarak tempuh terdekat yang dituju penumpang taksi pada rangking pertama dengan jarak terdekat adalah rute A-B-D-E dengan jarak 15 km dan waktu tempuh 29 menit dengan tarif Rp 48.000,-..

B. Diagram Konteks Dan Tampilan

1. Diagram Konteks

Diagram konteks atau *data flow diagram* (DFD) level 0, dimana sistem pertama kali dibuat.



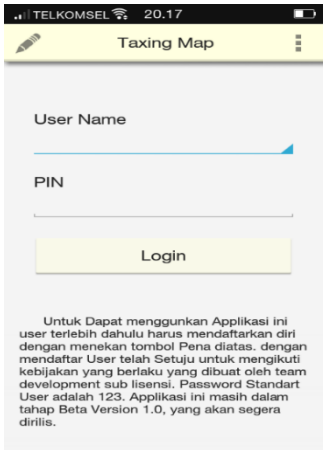
Gambar 4.1 : Diagram Konteks

2. Tampilan Aplikasi *Mobile*

a. Tampilan *login*

Didalam tampilan *login* terdapat 3 menu selain menu *login* itu sendiri, menu-menu tersebut memiliki fungsi yang berbeda namun saling terhubung satu atar lainya. Menu tersebut adalah menu login, menu daftar dan menu koneksi dan ke-3 menu tersebut saling keterkaitan atara satu sama lain.

Berikut ini adalah tampilan menu *login* pada aplikasi *mobile taxi* yang berfungsi menampilkan form *input username* dan *passsword* dari *user* (sopir *taxi*) untuk masuk menu selanjutnya pada aplikasi *mobile* :



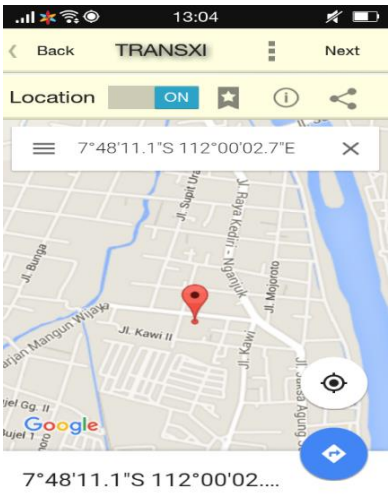
Gambar 5.1 : Menu *Login*

Gambar 5.2 : Conection PC

Gambar 5.3 : *New User*

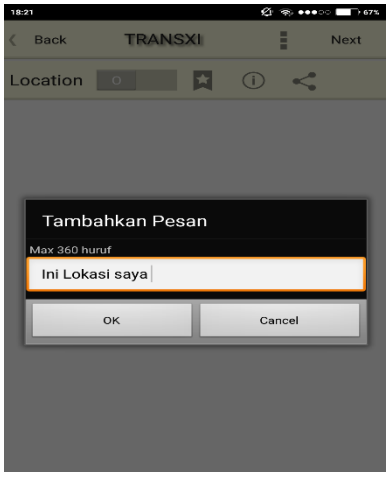
b. Menu awal

Berikut ini adalah tampilan menu awal pada aplikasi *mobile taxi* :



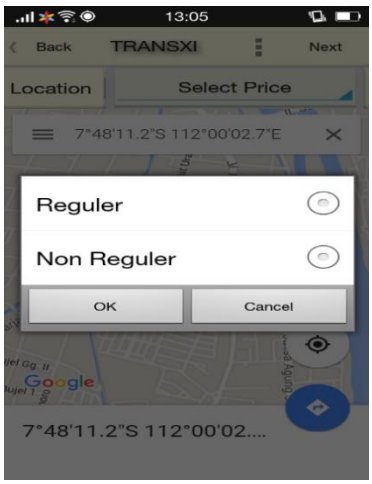
Gambar 5.4 : Menu Awal

c. Menu kirim lokasi



Gambar 5.5 : Tampilan Kirim Lokasi

d. Menu *select price*

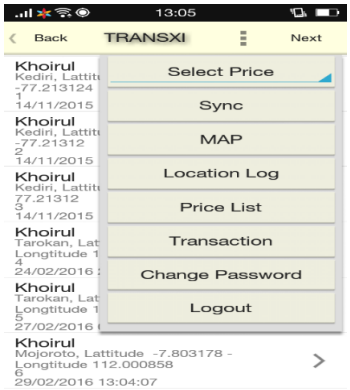


Gambar 5.6 : *Select Price*

e. Menu *tollbox select price*

Gambar 5.7 : Insert Jarak

f. Pilihan dalam *next*



Gambar 5.8 : Pilihan *Next*

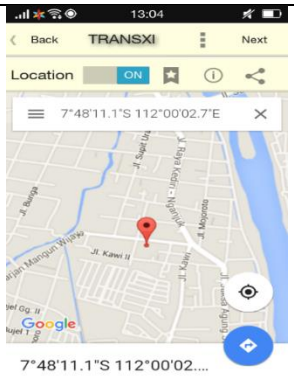
Menu yang tersedia tersebut memiliki informasi dan masing-masing sesuai dengan penggunaanya. Berikut adalah pemaparan menu yang ada di pilihan *next* :

1) *sync*

merupakan singkatan dari singkron yang berfungsi untuk mengirim dan mengambil data perbaruan ke aplikasi *dasboard*.

2) *MAP*

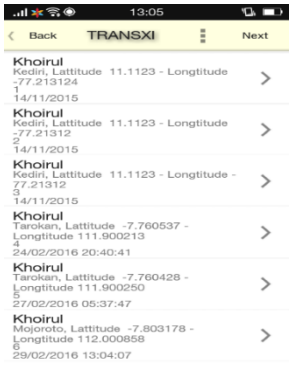
Merupakan pilihan menu untuk menuju ke map sebelumnya atau menu awal. Berikut adalah tampilannya :



Gambar 5.9 : *MAP* (menu awal)

3) *Location Log*

Berikut adalah tampilan menu *location log* pada aplikasi *mobile* taxi :



Gambar 5.10 : *Location Log*

Menu ini merupakan menu awal yang menampilkan lokasi dimana aplikasi tersebut berada.

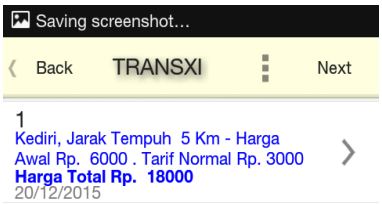
4) *Price List*

Berikut adalah tampilan menu *price list* pada aplikasi *mobile taxi* :

Gambar 5.11 : *Price List*

5) *Transaction*

Berikut adalah tampilan menu *transaction* pada aplikasi *mobile taxi* :

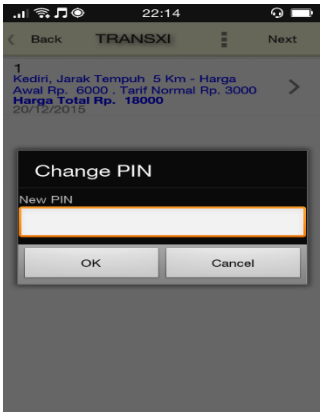


Gambar 5.12 : *Transaction*

Tampilan ini menampilkan daftar transaksi perjalanan yang telah dilakukan dan nominal harga yang diterima.

6) *Change Password*

Pilihan menu ini untuk mengubah *password* sesuai dengan yang diinginkan sopir. Berikut adalah tampilannya :



Gambar 5.13 : *Change Password*

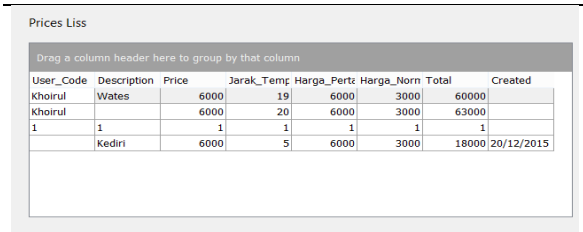
7) Logout

Pilihan ini untuk kembali ke menu login.

3. Tampilan Aplikasi *Dasboard*

a. Tampilan *Price Liss*

Berikut adalah tampilan harga pada aplikasi *dasboard* taxi :

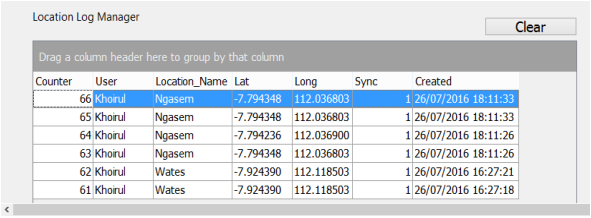


User_Code	Description	Price	Jarak_Temp	Harga_Perta	Harga_Norn	Total	Created
Khoirul	Wates	6000	19	6000	3000	60000	
Khoirul		6000	20	6000	3000	63000	
1	1	1	1	1	1	1	
	Kediri	6000	5	6000	3000	18000	20/12/2015

Gambar 5.14 : Tampilan *Price Liss*

b. Tampilan *location log*

Berikut adalah tampilan *location log* pada aplikasi *dasboard taxi* :

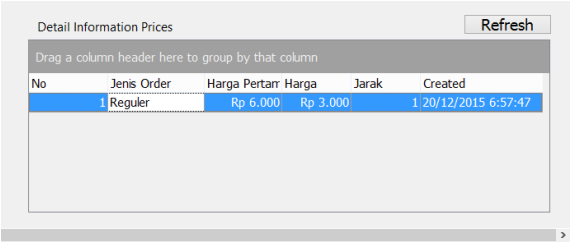


Counter	User	Location_Name	Lat	Long	Sync	Created
66	Khoirul	Ngasem	-7.794348	112.036803	1	26/07/2016 18:11:33
65	Khoirul	Ngasem	-7.794348	112.036803	1	26/07/2016 18:11:33
64	Khoirul	Ngasem	-7.794236	112.036900	1	26/07/2016 18:11:26
63	Khoirul	Ngasem	-7.794348	112.036803	1	26/07/2016 18:11:26
62	Khoirul	Wates	-7.924390	112.118503	1	26/07/2016 16:27:21
61	Khoirul	Wates	-7.924390	112.118503	1	26/07/2016 16:27:18

Gambar 5.15 : Tampilan *Location Log*

c. Tampilan Seting *Prices*

Berikut adalah tampilan seting *prices* pada aplikasi *dasboard taxi* :



No	Jenis Order	Harga Pertam	Harga	Jarak	Created
1	Reguler	Rp 6.000	Rp 3.000	1	20/12/2015 6:57:47

Gambar 5.16 : Tampilan Seting *Prices*

C. SIMPULAN

Berdasarkan permasalahan yang telah dibahas dan diselesaikan melalui laporan tugas akhir ini, maka terdapat beberapa kesimpulan yaitu sebagai berikut :

1. Peneliti berhasil menghitung perhitungan argometer jarak yang terdapat pada argo *taxi*.
2. Peneliti berhasil merancang sistem pendukung keputusan penentu rute terdekat tujuan penumpang menggunakan algoritma *Floyd-Warshall* berbasis aplikasi *Android*.

IV. DAFTARPUSTAKA

Admin. GigaPurbalingga. 2013. http://gigapurbalingga.com/premiumsoft-navicat-premium-enterprise-full/.

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/2006-2007/Makalah_2007/MakalahSTMIK2007-021.pdf, diakses, 14 Februari 2015.

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, diakses, 22 Maret 2015.

Eko. R. 2002. Program aplikasi GPS dan GIS untuk mencarilokasi dan jarak SPBU di Tangerang selatan denganpetadan *Augmented Reality Camera-View* pada perangkat bergerak berbasis *android*. Disertasi. Tidak dipublikasikan Jurusan Teknik Informatika,Universitas Gunadarma.

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, diakses, 22 Februari 2015.

Johari. 2003. *Sejarah Taxi*. tersedia: http://m.kaskus.co.id/thread/5218c9371cd719a24c000001/sejarah-taxi/ diakses, 07 Januari 2015.

Lecture 12: tariff taksi di Jakarta. (online) tersedia : http://www.numbeo.com/taxi-fare/city_result.jsp?country=Indonesia&city=Jakarta, diakses, 07 Januari 2015

Lina. 2014. *Aplikasi Pencarian Rumah Kontrakan Dengan Jarak Terdekat Menggunakan Algoritma Floyd-Warshall*. UNP Kediri jurusan Teknik Informatika

Nofriansyah. Dicky. (2014). Sistem Pendukung Keputusan http://infodanpengertian.blogspot.co.id/2015/04/pengertian-sistem-pendukung-keputusan.html?m=1, 22 februari 2015.

Pamangus 2013. *Beginilah Perhitungan Argo Taksi*. (online), tersedia : http://m.kaskus.co.id/thread/5218c9371cd719a24c000001/beginilah-perhitungan-argo-taxi/, diakses : 22 Februari 2015.

Rahmat,albert. 2013http://lsi.si.fti.unand.ac.id/mengenal-google-maps/.

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 22 Maret 2015.

Seffens, W. 2002. Graph Theory Patterns in the Genetic Codes, *Forma*, 17, 309–320, 2013.

Yuli Sugianto. 2012. *argometer digital (untuk semua jenis kendaraan)*. (Online), tersedia: file:///G:/New%20Folder/MICROTECH%20ENGINEERING%20%20ARGOMETER%20DIGITAL%20%28untuk%20semua%20jenis%20kendaraan%29%29.htm, dikses : 1 mei 2015.