

**PENERAPAN ALGORITMA TRAVELLING SALESMAN PROBLEM
PROBLEM WITH PRECEDENCE CONSTRAINT (TSPPC) PADA
OPTIMALISASI RUTE TERPENDEK PEMANDU WISATA DI
YOGYAKARTA BERBASIS ANDROID**

Skripsi

Diajukan Sebagai Persyaratan
Memperoleh Gelar Sarjana Strata Satu
Program Studi Teknik Informatika



Disusun oleh
MARDHON TSAH ZAKI
10651066

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA
2015**

**PENERAPAN ALGORITMA TRAVELLING SALESMAN PROBLEM
PROBLEM WITH PRECEDENCE CONSTRAINT (TSPPC) PADA
OPTIMALISASI RUTE TERPENDEK PEMANDU WISATA DI
YOGYAKARTA BERBASIS ANDROID**

Skripsi

Diajukan Sebagai Persyaratan
Memperoleh Gelar Sarjana Strata Satu
Program Studi Teknik Informatika



Disusun oleh
MARDHON TSAH ZAKI
10651066

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA**

2015



PENGESAHAN SKRIPSI/TUGAS AKHIR

Nomor : UIN.02/D.ST/PP.01.1/ o66 /2016

Skripsi/Tugas Akhir dengan judul : Penerapan Algoritma Travelling Salesmen Problem Problem With Precedence Constraint (TSPPC) Pada Optimisasi Rute Terpendek Pemandu Wisata di Yogyakarta Berbasis Android

Yang dipersiapkan dan disusun oleh :
Nama : Mardhon Tsah Zaki
NIM : 10651066
Telah dimunaqasyahkan pada : Kamis, 10 Desember 2015
Nilai Munaqasyah : B +
Dan dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga

TIM MUNAQASYAH :

Ketua Sidang

Sumarsono, M.Kom
NIP. 19710209 200501 1 003

Penguji I

Nurochman, M.Kom
NIP.19801223 200901 1 007

Penguji II

Dr. Bambang Sugiantoro, M.T
NIP. 19751024 200912 1 002

Yogyakarta, 7 Januari 2016
UIN Sunan Kalijaga
Fakultas Sains dan Teknologi



Dekan

Dr. Maizer Said Nahdi, M.Si.
NIP. 19550427 198403 2 001



SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Persetujuan Skripsi
Lamp : 1 Bendel Laporan Skripsi

Kepada
Yth. Dekan Fakultas Sains dan Teknologi
UIN Sunan Kalijaga Yogyakarta
di Yogyakarta

Assalamu 'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Mardhon Tsah Zaki
NIM : 10650055
Judul Skripsi : Penerapan Algoritma Travelling Salesman Problem Problem With
Precedence Constraint (TSPPC) Pada Optimalisasi Rute Terpendek
Pemandu Wisata Di Yogyakarta Berbasis Android

Sudah dapat diajukan kembali kepada Program Studi Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Prodi Teknik Informatika

Dengan ini kami berharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqsyahkan. Atas perhatiannya kami ucapkan terima kasih.

Wassalamu 'alaikum wr. wb.

Yogyakarta, 04 September 2015

Pembimbing

Sumarsono, M.Kom

NIP. 19710209 200501 1 003

PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan di bawah ini:

Nama : Mardhon Tsah Zaki

NIM : 10651066

Program Studi : Teknik Informatika

Fakultas : Sains dan Teknologi

Menyatakan bahwa skripsi dengan judul **“Penerapan Algoritma Travelling Salesman Problem Problem With Precedence Constraint (TSPPC) Pada Optimalisasi Rute Terpendek Pemandu Wisata Di Yogyakarta Berbasis Android”** tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 04 September 2015

Yang menyatakan



Mardhon Tsah Zaki
NIM. 10651066

HALAMAN MOTTO

1. Be You' re Self
2. “Jangan dating untuk mengeluh, tapi datang untuk belajar.”



KATA PENGANTAR



Alhamdulillah, puji syukur kehadirat Allah SWT, Tuhan semesta alam yang tak pernah berhenti memberikan segala limpahan rakhmat, taufik, hidayah dan nikmat-Nya, penulis mampu menyelesaikan skripsi yang berjudul Penerapan Algoritma *Travelling Salesman Problem Problem With Precedence Constraint (TSPPC)* Pada Optimalisasi Rute Terpendek Pemandu Wisata Di Yogyakarta Berbasis Android.

Penulis mengucapkan terima kasih kepada pihak-pihak yang turut serta baik bantuan, nasehat, bimbingan dan dukungannya sehingga dapat menyelesaikan pembuatan skripsi ini. Ucapan terima kasih penulis sampaikan kepada :

1. Orang tua dan seluruh anggota keluarga yang selalu memberikan nasehat, dukungan dan doanya.
2. Prof. Drs. H. Akh. Minhaji, MA. Ph.D, selaku Rektor Universitas Islam Negeri Sunan Kalijaga.
3. Dr. Maizer Said Nahdi, M.Si selaku Dekan Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
4. Bapak Sumarsono, ST., M.Kom selaku Ketua Program Studi Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
5. Bapak Sumarsono, ST., M.Kom selaku Dosen Pembimbing Akademik dan Dosen Pembimbing Skripsi yang telah memberikan arahan dan masukan selama penyusunan skripsi ini secara teliti dan rinci.

6. Bapak dan Ibu Dosen Program Studi Teknik Informatika yang telah memberikan bekal ilmu pengetahuannya kepada penulis selama mengabdikan di UIN Sunan Kalijaga, semoga ilmu ini penulis manfaatkan dengan sebaik-baiknya.
7. Teman-teman se-angkatan 2010, khususnya angkatan 2010 Mandiri, terima kasih atas kebersamaannya, pahit manis mengikuti perkuliahan selama ini, semoga jangan sampai putus silaturahmi diantara kita.
8. Teman-teman Pandanaran, teman-teman Klabanan, IKPMB, teman-teman wisma gaplek, teman-teman Kos 26B, Didik N, Ibnu Brewok, dan teman-teman yang lain yang selama ini selalu ada dan hidup bersama-sama di Kota Rantau ini.

Penulis berharap semoga Allah SWT membalas kebaikan dan ketulusan semua pihak yang telah banyak membantu penulis dalam menyelesaikan skripsi ini Amiin. Besar harapan besar penulis, semoga skripsi ini dapat bermanfaat bagi kemajuan dan perkembangan ilmu pengetahuan dan menjadi sumbangan pemikiran bagi pihak yang membutuhkan terutama dalam bidang Teknik Informatika. Amin ya Robbal Alamiin.

Yogyakarta, 10 Oktober 2015
Penyusun

Mardhon Tsah Zaki
10651066

HALAMAN PERSEMBAHAN

Alhamdulillahirrabbi'l'alamiin, Puji Syukur kehadiran Allah SWT yang telah memberikan nikmat iman, nikmat islam, hingga nikmat kesehatan, serta petunjuk dalam kesulitan. Sholawat serta salam senantiasa tercurahkan kepada Nabi Muhammad SAW. Skripsi ini dipersembahkan kepada:

1. Abah, Mamah, Mba oti, Mas Wafa, dan hadin yang selalu memberikan nasehat, dukungan dan doanya.
2. Prof. Drs. H. Akh. Minhaji, MA. Ph.D, selaku Rektor Universitas Islam Negeri Sunan Kalijaga.
3. Dr. Maizer Said Nahdi, M.Si selaku Dekan Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
4. Bapak Sumarsono, ST., M.Kom selaku Ketua Program Studi Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
5. Bapak Sumarsono, ST., M.Kom selaku Dosen Pembimbing Akademik dan Dosen Pembimbing Skripsi yang telah memberikan arahan dan masukan selama penyusunan skripsi ini secara teliti dan rinci.
6. Bapak dan Ibu Dosen Program Studi Teknik Informatika yang telah memberikan bekal ilmu pengetahuannya kepada penulis selama mengabdikan di UIN Sunan Kalijaga, semoga ilmu ini penulis manfaatkan dengan sebaik-baiknya.
7. Teman-teman se-angkatan 2010, khususnya angkatan 2010 Mandiri, terima kasih atas kebersamaannya, pahit manis mengikuti perkuliahan selama ini, semoga jangan sampai putus silaturahmi diantara kita.

8. Teman-teman Pandanaran, teman-teman Klabanan, IKPMB, teman-teman wisma gaplek, teman-teman Kos 26B, Didik N, Ibnu Brewok, dan teman-teman yang lain yang selama ini selalu ada dan hidup bersama-sama di Kota Rantau ini.
9. Semua pihak yang mendukung dan membantu dalam penyusunan dan penyelesaian skripsi ini.



DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN SKRIPSI.....	ii
SURAT PERSETUJUAN SKRIPSI.....	iii
PERNYATAAN KEASLIAN SKRIPSI.....	iv
HALAMAN MOTO	v
KATA PENGANTAR	vi
HALAMAN PERSEMBAHAN	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR	xiv
DAFTAR TABEL.....	xv
DAFTAR LAMPIRAN.....	xvi
INTISARI.....	xvii
ABSTRACT.....	xviii
BAB I PENDAHULUAN	1
1.1 LatarBelakang	1
1.2 RumusanMasalah.....	4
1.3 BatasanMasalah	5
1.4 TujuanPenelitian	6
1.5 ManfaatPenelitian	6
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI	7
2.1 TinjauanPustaka.....	7
2.2 LandasanTeori.....	10
2.2.1 SistemInformasiGeografis.....	10
2.2.2 Android.....	11
2.2.3 <i>Object Oriented Programming</i> (OOP).....	12
a. <i>Object</i>	12
b. <i>Class</i>	13
c. <i>Attributes</i>	14

d.	<i>Behavior</i>	15
e.	Abstraksi.....	17
2.2.4	<i>Java</i>	17
2.2.5	<i>Eclipse IDE</i>	18
2.2.6	<i>MySQL</i>	19
2.2.7	<i>Google Maps</i>	20
2.2.8	<i>ADT Plugin for Eclipse Android Development Tools</i> (ADT)	20
2.2.9	<i>Global Positioning System (GPS)</i>	21
2.2.10	<i>Algoritma Travelling Salesman Problem with Precedence</i> <i>Constraints (TSPPC)</i>	22
a.	TSPPC Dengan 3 Kota	23
b.	TSPPC Dengan 4 Kota	24
c.	TSPPC Dengan 5 Kota	25
d.	TSPPC Dengan N Kota	26
2.2.11	<i>UML (Unified Modelling Language)</i>	27
a.	<i>Diagram Usecase (Usecase Diagram)</i>	28
b.	<i>Class Diagram</i>	29
c.	<i>Activity Diagram</i>	31
d.	Diagram Kolaborasi	31
e.	Diagram Sekuensial.....	32
f.	Diagram <i>Statechart</i>	33
g.	Diagram Komponen	34
h.	Diagram <i>Deployment</i>	35
2.2.12	<i>Snowball Sampling</i>	36
2.2.13	<i>Web Server</i>	37
2.2.14	<i>Sistem Web Server</i>	40
BAB III METODE PENGEMBANGAN SISTEM		42
3.1	Studi Pendahuluan	42
a.	Deteksi lokasi sekarang (<i>Current Location</i>) dengan GPS	42
b.	<i>Google Maps API</i>	43

3.2 Metode Pengembangan Sistem	43
a. Studi Literatur	43
b. Wawancara	44
c. Observasi	44
3.2.1 Desain Sistem	44
3.2.2 Implementasi Rancangan	44
3.2.3 Pengujian Sistem	45
3.2.4 Analisis dan Kesimpulan	45
3.2.5 Pembuatan Laporan	45
BAB IV ANALISIS DAN PERANCANGAN SISTEM	46
4.1 Analisis Masalah	46
4.2 Analisis Kebutuhan Sistem	47
4.3 Analisis Kebutuhan Non Fungsional	48
4.3.1 Analisis Pengguna	48
4.3.2 Analisis Kebutuhan Perangkat Keras	48
4.3.3 Analisis Kebutuhan Perangkat Lunak	48
4.3.4 Analisis Kebutuhan Data	49
4.4 Analisis Kebutuhan Fungsional	50
4.4.1 Desain <i>Interface</i> (Antarmuka)	50
4.4.2 Desain Antarmuka Sistem Klien	50
a. Desain Antarmuka <i>Form</i> Menu Utama	50
b. Desain Antarmuka <i>Form</i> Menu Kategori	51
c. Desain Antarmuka <i>Form</i> Data <i>List</i> Tempat wisata	52
d. Desain Antarmuka <i>Form</i> Detail tempat wisata	53
e. Desain Antarmuka <i>Form</i> Pilihan Lokasi	54
f. Desain Antarmuka <i>Form</i> <i>Direction</i>	55
g. Desain Antarmuka <i>Form</i> Pencarian	56
4.5 <i>Usecase</i>	57
a. Definisi aktor	58
b. Definisi <i>usecase</i>	59
4.6 <i>Activity Diagram</i>	60

a. <i>Activity diagram</i> Lihat Tempat wisata.....	60
b. <i>Activity diagram</i> Pencarian Tempat wisata.....	61
c. <i>Activity diagram</i> Penentuanlokasiterdekat.....	62
d. <i>Activity diagram</i> TambahobyekWisata.....	63
4.7 <i>Class Diagram</i>	64
BAB V IMPLEMENTASI DAN PENGUJIAN SISTEM	66
5.1 Implementasi.....	66
5.1.1 Implementasi Sistem Klien.....	66
a. Implementasi Menu Utama.....	66
b. Implementasi Menu Data <i>TravelAgent</i> Berdasar Kategori	70
c. Implementasi Menu Data <i>List TravelAgent</i>	71
d. Implementasi Menu Detail <i>TravelAgent</i>	73
e. Implementasi Menu Detail <i>TravelAgent</i> PilihanLokasi.....	75
f. Implementasi Menu Detail <i>TravelAgentDirection</i>	78
g. Implementasi Menu pencarian <i>TravelAgent</i>	80
h. Implementasi Menu TambahLokasi.....	84
5.2 Pengujian.....	89
5.2.1 Pengujian Alpha	89
5.2.2 Pengujian Beta.....	90
BAB VI HASIL DAN PEMBAHASAN	93
6.1 HasilPengujianSistem	93
6.2 HasildanPembahasanPengujian Alpha.....	94
6.3 HasildanPembahasanPengujianBeta	95
a. PengujianFungsionalitasSistem	95
b. PengujianAntarmukaSistem.....	97
c. KesimpulanHasilPengujian Beta	98
BAB VII PENUTUP	100
7.1 Kesimpulan	100
7.2 Saran	100
DAFTAR PUSTAKA	102
LAMPIRAN	104

DAFTAR GAMBAR

2.1	Gambar <i>TSPPC</i> Dengan 3 Kota.....	23
2.2	Gambar <i>TSPPC</i> Dengan 4 Kota.....	24
2.3	Gambar Hasil pengujian <i>TSPPC</i> Dengan 4 Kota	24
2.4	Gambar <i>TSPPC</i> Dengan 5 Kota.....	25
2.5	Gambar Hasil Pengujian <i>TSPPC</i> Dengan 5 Kota	26
2.6	Gambar Cara kerja <i>web server</i>	43
4.1	Gambar Desain Antarmuka Form Menu Utama.....	50
4.2	Gambar Desain Antarmuka Form Menu Kategori.....	51
4.3	Gambar Desain antarmuka form data list tempat wisata	52
4.4	Gambar Desain Antarmuka Form Detail tempat wisata	53
4.5	Gambar Desain Antarmuka Form Pilihan Lokasi.....	54
4.6	Gambar Desain Antarmuka Form <i>Direction</i>	55
4.7	Gambar Desain Antarmuka Form Pencarian	56
4.8	Gambar Desain Rancangan <i>Usecase</i>	58
4.9	Gambar Activity diagram Lihat Tempat wisata.....	60
4.10	Gambar Activity diagram Pencarian Tempat wisata	61
4.11	Gambar Activity diagram Penentuan lokasi terdekat	62
4.12	Gambar Activity diagram Tambah obyek Wisata	63
4.13	Gambar Class Diagram Klien	65
5.1	Gambar Menu Utama Klien.....	67
5.2	Gambar Menu Data <i>TravelAgent</i> Berdasar Kategori.....	70
5.3	Gambar Menu Data <i>List TravelAgent</i>	72
5.4	Gambar Menu Detail data <i>TravelAgent</i>	74
5.5	Gambar Menu Detail <i>TravelAgent</i> Pilihan Lokasi.....	76
5.6	Gambar Menu Detail <i>TravelAgent Direction</i>	79
5.7	Gambar Menu Pencarian <i>TravelAgent</i>	81
5.8	Gambar Menu Tambah Lokasi	85

DAFTAR TABEL

2.1	TabelDaftar Perbandingan Penelitian	9
2.2	TabelKomponenUsecase.....	29
2.3	TabelKomponen <i>Class</i> Diagram.....	30
2.4	Komponen <i>Activity</i> Diagram.....	31
2.5	Komponen Diagram Kolaborasi	32
2.6	Komponen Diagram Sekuensial	33
2.7	Komponen Diagram <i>Statechart</i>	33
2.8	Komponen Diagram Komponen	35
2.9	Komponen Diagram <i>Deployment</i>	36
4.1	TabelDaftar TravelAgent	138
4.2	TabelDaftar Aktor	58
4.3	TabelUsecase	59
5.1	Tabel Rencana Pengujian Alpha.....	89
5.2	Tabel Pengujian Fungsional Sistem.....	90
5.3	Tabel Pengujian Antarmuka Sistem.....	91
6.1	Tabel Daftar Responden	93
6.2	Tabel Hasil Pengujian Fungsional Sistem	95
6.3	Tabel Hasil Pengujian Antarmuka Sistem	97

DAFTAR LAMPIRAN

LAMPIRAN A <i>Source Code</i>	104
LAMPIRAN B <i>DaftarTravelAgent</i>	136
LAMPIRAN C <i>LembarAngketPengujianSistem</i>	142



**Penerapan Algoritma Travelling Salesman Problem Problem With Precedence
Constraint (TSPPC) Pada Optimalisasi Rute Terpendek Pemandu Wisata Di
Yogyakarta Berbasis Android**

MARDHON TSAH ZAKI
NIM. 10651066

INTISARI

Yogyakarta merupakan salah satu pusat wisata dan budaya. Banyak wisatawan baik wisatawan dalam negeri maupun mancanegara yang datang ke kota Yogyakarta karena beragamnya lokasi wisata yang dapat dikunjungi dengan keunikan yang beragam. Mulai dari wisata alam, museum-museum pra sejarah, tempat perbelanjaan oleh-oleh, hingga candi-candi tempat peninggalan pra sejarah. Hal tersebut merupakan salah satu daya tarik wisatawan untuk berkunjung ke kota Yogyakarta.

Sistem ini menggunakan algoritma Travelling Salesman Problem With Precedence Constraints Tujuan dari penggunaan algoritma travelling salesman problem with precedence constraints (TSPPC) ini adalah untuk menentukan jalur terpendek berdasarkan jarak terpendek dari satu titik ke titik lainnya. Misalkan titik menggambarkan suatu lokasi yang ingin kita tempuh dan garis merupakan jalur atau jalannya, maka pada penerapan algoritma Travelling Salesman Problem With Precedence Constraints (TSPPC) ini, algoritma melakukan kalkulasi terhadap semua kemungkinan jarak terpendek dari setiap titik. Perbedaan lain yang menjadi titik berat penelitian adalah menerapkan algoritma Travelling Salesman Problem With Precedence Constraints (TSPPC) dengan konsep pencarian jarak terdekat terhadap suatu lokasi untuk multi tujuan dengan mendapatkan solusi optimal secara global dari sisi jarak. Pencarian rute terpendek diawali dengan mengetahui posisi asal dan tujuan wisata yang akan dikunjungi. Posisi asal dan tujuan wisata yang telah diketahui akan digunakan pada proses pembangunan tur perjalanan dan pencarian rute terpendek

Sistem ini mampu menampilkan lokasi pengguna dengan lokasi wisata beserta informasi lokasi wisata. Sistem dapat menampilkan lokasi wisata terdekat dari pengguna pada peta virtual Google Map tanpa mempertimbangkan masalah-masalah yang dapat mempengaruhi waktu tempuh seperti kemacetan, lebar jalan, dan traffic light. Berdasarkan hasil pengujian fungsional sistem, semua responden setuju bahwa aplikasi yang dibuat berfungsi sebagaimana mestinya. Berdasarkan hasil pengujian antarmuka sistem, menunjukkan bahwa 85% menyatakan setuju dan 15% menyatakan netral.

Kata Kunci : Android, Google Maps, Mobile , Traveling , Travelling Salesman Problem With Precedence Constraints (TSPPC)

**Application of Travelling Salesman Problem With Precedence Constraint
(TSPPC) Algorithm on Optimization the Shortest Tour Guide Route in
Yogyakarta Based Android**

MARDHON TSAH ZAKI
NIM. 10651066

ABSTRACT

Yogyakarta is one of tourism and cultural centers. There are many tourists whether domestic or international tourists who come to Yogyakarta because it has many tourist attractions which can be visited with various uniqueness. There are natural tourism, museums, souvenirs shops, and temples. Those are tourist attractions which can be visited in Yogyakarta.

This system is using Travelling Salesman Problem With Precedence Constraints algorithm. The purpose of using this algorithm is to decide the shortest path based on the shortest path from one point to another point. For example, a point shows a location where we want to go and the line is the path, then on the application of Travelling Salesman Problem With Precedence Constraints (TSPPC) algorithm, the algorithm calculates for all the possible shortest distances from any point. Another difference which become the focus of the research is to apply the Travelling Salesman Problem With Precedence Constraints (TSPPC) algorithm with the concept of searching for closest distance to a location for the multi-purpose. The use of the searching concept to a location is done to get globally optimal solution in terms of distance. The search of shortest route begins by knowing the start point and the destination that will be visited. The start point and the destination which are known will be used on the development process of route and the shortest route search.

This system can show the user's location with tourism location with its information. The system can show the closest tourism to the user on Google Maps without considering problems that can affect travel times such as traffic jam, road width, and traffic light. According to the result of functional system testing, all respondents agree that the application functions properly. According to interface system testing, shows that 85% is agree and 15% is neutral.

Key Words : Android, Google Maps, *Mobile* , *Traveling* ,*Travelling Salesman Problem With Precedence Constraints (TSPPC)*

BAB I

PENDAHULUAN

1.1. LATAR BELAKANG MASALAH

Yogyakarta merupakan salah satu pusat wisata dan budaya. Banyak wisatawan baik wisatawan dalam negeri maupun mancanegara yang datang ke kota Yogyakarta karena beragamnya lokasi wisata yang dapat dikunjungi dengan keunikan yang beragam. Mulai dari wisata alam, museum-museum pra sejarah, tempat perbelanjaan oleh-oleh, hingga candi-candi tempat peninggalan pra sejarah. Hal tersebut merupakan salah satu daya tarik wisatawan untuk berkunjung ke kota Yogyakarta.

Pada era global dalam masyarakat dewasa ini terdapat masalah-masalah yang dihadapi, salah satunya adalah mobilitas yang tinggi terhadap kehidupannya, dengan pemanfaatan berbagai macam gadget-gadget berupa smartphone (telpon pintar) berupa Android, Apple, Windows Phone dan lain sebagainya. Faktor-faktor yang menunjang mobilitas tersebut salah satunya adalah dengan menentukan rute terdekat dalam mencapai suatu lokasi sebagai tempat tujuannya.

Masalah yang terjadi dan sering kita temukan pada komputasi salah satunya adalah dalam pencarian rute terpendek (shortes path) suatu lokasi. Banyak metode-metode yang digunakan dalam pencarian suatu lokasi terdekat, metode-metode yang digunakan biasanya menggunakan

algoritma dijkstra, dan algoritma warshall. Metode lainnya untuk masalah pencarian lokasi terdekat yaitu dengan menggunakan branch and bound, nearest neighbour, algoritma genetika, Ant colony algorithm, bee colony optimization, dan *Travelling Salesman Problem With Precedence Constraints (TSPPC)*.

Dalam menentukan rute terpendek terdapat banyak masalah-masalah yang terjadi. Dalam penyelesaian masalah tersebut banyak cara yang digunakan, salah satunya adalah dengan menggunakan algoritma *Travelling Salesman Problem With Precedence Constraints* Tujuan dari penggunaan algoritma *Travelling Salesman Problem With Precedence Constraints (TSPPC)* ini adalah untuk menentukan jalur terpendek berdasarkan jarak terpendek dari satu titik ke titik lainnya. Misalkan titik menggambarkan suatu lokasi yang ingin kita tempuh dan garis merupakan jalur atau jalannya, maka pada penerapan algoritma *Travelling Salesman Problem With Precedence Constraints (TSPPC)* ini, algoritma melakukan kalkulasi terhadap semua kemungkinan jarak terpendek dari setiap titik.

Penelitian yang akan dilakukan ini akan diterapkan pada masalah pemandu wisatawan untuk menentukan objek wisata yang ada di Daerah Istimewa Yogyakarta dengan menggunakan algoritma *Travelling Salesman Problem With Precedence Constraints (TSPPC)*. Faktor yang dipertimbangkan dalam penelitian ini adalah faktor jarak tempuh tanpa mempertimbangkan masalah-masalah yang dapat mempengaruhi waktu

tempuh seperti kemacetan, lebar jalan, dan traffic light. Penelitian yang akan dilakukan menggunakan teknik pencarian lokal menggunakan pencarian rute terdekat dari posisi awal ke tempat-tempat yang akan dikunjungi. Perbedaan lain yang menjadi titik berat penelitian adalah menerapkan algoritma *Travelling Salesman Problem With Precedence Constraints (TSPPC)* dengan konsep pencarian jarak terdekat terhadap suatu lokasi untuk multi tujuan. Penggunaan konsep pencarian jarak terdekat terhadap suatu lokasi ini dilakukan untuk mendapatkan solusi optimal secara global dari sisi jarak. Pencarian rute terpendek diawali dengan mengetahui posisi asal dan tujuan wisata yang akan dikunjungi. Posisi asal dan tujuan wisata yang telah diketahui akan digunakan pada proses pembangunan tur perjalanan dan pencarian rute terpendek dengan algoritma *Travelling Salesman Problem With Precedence Constraints (TSPPC)*.

1.2. RUMUSAN MASALAH

Berdasarkan latar belakang di atas, maka terdapat beberapa rumusan masalah sebagai berikut:

1. Bagaimana membangun aplikasi pemandu wisata untuk mengoptimalkan rute terpendek lokasi wisata dengan menampilkan informasi yang ada di Daerah Istimewa Yogyakarta?
2. Bagaimana membangun aplikasi pemandu wisata yang berbasis android?
3. Bagaimana memvisualisasikan aplikasi pemandu wisata untuk pencarian optimal rute terpendek terhadap lokasi wisata di Yogyakarta dengan memanfaatkan Google Maps API untuk menampilkan lokasi ke dalam bentuk map lokasi Yogyakarta?
4. Bagaimana menerapkan algoritma *Travelling Salesman Problem With Precedence Constraints (TSPPC)* terhadap masalah pencarian rute optimal pada aplikasi pemandu wisata?
5. Bagaimana membangun aplikasi yang memberikan informasi-informasi mengenai lokasi wisata?

1.3. BATASAN MASALAH

Ruang lingkup permasalahan yang dibatasi dalam penelitian ini adalah:

1. Aplikasi ini hanya berlaku pada platform Android versi Jellybean.
2. Aplikasi ini memanfaatkan Google Maps API untuk menampilkan lokasi dan jarak yang harus ditempuh ke dalam bentuk bentuk map lokasi Yogyakarta
3. Aplikasi ini menerapkan metode *Travelling Salesman Problem With Precedence Constraints (TSPPC)* algorithm berbasis android untuk mengoptimalkan rute terpendek dalam pencarian lokasi-lokasi wisata di Yogyakarta.
4. Aplikasi ini mengoptimalkan jarak terdekat dari beberapa lokasi wisata tanpa mempertimbangkan masalah-masalah yang dapat mempengaruhi waktu tempuh seperti kemacetan, lebar jalan, dan traffic light.
5. Aplikasi ini memberikan informasi-informasi mengenai lokasi wisata tersebut.

1.4. TUJUAN PENELITIAN

Berdasarkan latar belakang dan rumusan masalah di atas, maka tujuan dari penelitian ini adalah:

1. Mengimplementasikan perancangan ke dalam aplikasi dalam platform sistem operasi mobile berbasis Android versi Jellybean.
2. Memanfaatkan Google Maps API untuk menampilkan lokasi dan jarak yang harus ditempuh ke dalam bentuk map lokasi Yogyakarta.
3. Menerapkan algoritma *Travelling Salesman Problem With Precedence Constraints (TSPPC)* untuk mengoptimalkan rute terpendek dalam pencarian lokasi-lokasi wisata di Yogyakarta.
4. Memberikan rute terpendek dalam pencarian lokasi wisata di Yogyakarta.
5. Memberikan informasi mengenai lokasi wisata yang ada di Yogyakarta.

1.5. MANFAAT PENELITIAN

Hasil penelitian ini diharapkan dapat membantu pemandu wisata dalam menentukan dan mengoptimalkan penentuan jarak tempuh pencarian beberapa lokasi wisata yang akan dikunjungi di kota Yogyakarta dengan smartphone android.

BAB VII

KESIMPULAN DAN SARAN

7.1. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan penulis pada Aplikasi yang menggunakan GoogleMaps Berbasis Android maka dapat diambil kesimpulan yaitu, penelitian ini berhasil mengembangkan Aplikasi *TravelAgent* menggunakan GoogleMaps Berbasis Android.

7.2. SARAN

Penelitian yang dilakukan tidak terlepas dari kekurangan dan kelemahan. Oleh karena itu, untuk kebaikan pengembangan sistem lebih lanjut, maka perlu diperhatikan beberapa hal, diantaranya:

1. Aplikasi bisa dikembangkan dengan menambah kan data lokasi wisata terbaru di setiap daerah.
2. Aplikasi ini dapat dikembangkan untuk mencari *TravelAgent* terdekat pada posisi saat ini tanpa mengisikan jarak.
3. Aplikasi dapat dikembangkan untuk mencari *TravelAgent* dengan mempertimbangkan kemacetan, berapa lama jarak tempuh sehingga data yang telah di lihat bias lebih jelas dengan beberapa pertimbangan yang ada.

4. Aplikasi bisa dikembangkan dengan menambahkan fitur penambahan lokasi di suatu daerah sehingga mendapatkan informasi mengenai daerah-daerah terbaru dan terupdate.

Akhirnya dengan segala keterbatasan hasil penelitian ini, penulis tetap berharap bahwa penelitian ini akan memberikan gagasan baru bagi pembaca untuk mengembangkan lebih lanjut.



DAFTAR PUSTAKA

- ~ Digilib Perpustakaan Universitas Binus, “*Euclidian Distance*”. 2009.
<http://thesis.binus.ac.id/doc/Bab2/2009-1-00477-TIAS%20Bab%202.pdf> (diakses 26 Juni 2015).
- ~ Digilib Perpustakaan Pusat Unikom, “*Client Server*”. 2009.
<http://thesis.binus.ac.id/doc/Bab2/2009-1-00477-TIAS%20Bab%202.pdf> (diakses 16 Mei 2015).
- Ersyaf, Fikri. *Rancang Bangun Aplikasi KindergartenFinder dan Call Dialler Menggunakan Google Maps Berbasis Android (Studi Kasus : Kindergarten Kota Yogyakarta)*. Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta 2013.
- Hermawan, Beny. *Menguasai Java dan Object Oriented Programming*. Yogyakarta: Penerbit Andi, 2004.
- ~ Kurniawan Khannedy, Eko “Java Dasar”. 2006.
https://b5edbab56750c0789c81a690a6f83bec7bf2426.googledrive.com/host/0B_j-ZV7TzBFAUIVHZ1ZNSGU3TE0/pemograman%20java%20aplication.pdf
(diakses 04 April 2015).
- Mufti, Yusuf. *Rancang Bangun Emergency Security Caller Berbasis Android untuk Situasi Perampokan*. Yogyakarta: Teknik Informatika FSaintek UIN Sunan Kalijaga 2013.
- Munir, Rinaldi. 2005. *Strategi Algoritmik*. Bandung. Institut Teknologi Bandung.
- Nazruddin Safaat H. (2011), *Pemograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Informatika, Bandung.
- ~ Noviani Fatimah, Wina. “*Pengenalan Eclipse*”. 2011.
<https://wi01.files.wordpress.com/2011/02/pengenalan-eclipse.pdf>
(diakses 04 Februari 2015).
- Nugroho, Adi. *Algoritma dan Struktur Data dalam Bahasa Java*. Yogyakarta: Penerbit Andi, 2008.
- ~ “Pengertian SIG (Sistem Informasi Geografis)”. 2013.
<http://www.pengertianahli.com/2013/10/pengertian-sig-sistem-informasi.html>
(diakses 15 November 2014).
- Peranginangin, Kasiman. *Aplikasi Web dengan PHP dan MySQL*. Yogyakarta: Penerbit Andi, 2006.
- Purnadi, Christian Wisnu, 2010, *Aplikasi Peta Mobile untuk Pencarian Jalur Terpendek Pada Sistem Operasi Android (Mobile Map Application For The Shortest Path Searching On Android Operating System)*. Skripsi Ilmu Komputer FMIPA UGM, Yogyakarta.

Sanjaya, R. *Pengolahan Database MSQl 5 dengan Java 2*. Yogyakarta: Penerbit Andi, 2005

Setiawan, Suryana, 2004 , *Implementasi Location Based Service dengan Bluetooth dan web service*. Jakarta : Fakultas Ilmu Komputer Universitas Indonesia

Sholih, 2006, *Permodelan Sistem Informasi Berorientasi Obyek dengan UML*, Graha Ilmu, Yogyakarta.

Sulistianto,Widy. *Aplikasi Mobile GIS Berbasis Android Lokasi Perguruan Tinggi Provinsi Daerah Istimewa Yogyakarta*. Teknik Informatika FT Industri Universitas Pembangunan Nasional “Veteran” Yogyakarta 2012

Surya Putra, Edo. *Rancang Bangun Aplikasi Pencarian Lokasi Toko Oleh-Oleh Yogyakarta Dengan Metode Location Based Service (Lbs) Berbasis Android*. Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta 2014.

~ Widianti, Santi, “Pengembangan Sistem”. 2009.

<http://santiw.staff.gunadarma.ac.id/Downloads/files/7689/pengembangansistem.doc>
(Diakses pada 15 April 2015).

~ Wikipedia “Pemrograman Beroorientasi Objek”. 2015.

https://id.wikipedia.org/wiki/Pemrograman_berorientasi_objek (diakses 15 November 2014).

~ Wikipedia “Traveling Salesman Problem”. 2015.

http://en.wikipedia.org/wiki/Travelling_salesman_problem (diakses 17 Agustus 2015)

LAMPIRAN A

Source Code

```
//class ini berguna untuk menampilkan detail obyek wisata

@SuppressLint("DefaultLocale")
public class DetailActivity extends Activity {

    public Wisata wisata;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail);

        Bundle bundle = this.getIntent().getExtras();
        wisata = (Wisata) bundle.getParcelable("wisata");

        ImageView bg = (ImageView) findViewById(R.id.image);
        TextView kategori = (TextView) findViewById(R.id.kategori);
        TextView nama = (TextView) findViewById(R.id.nama);
        TextView deskripsi = (TextView) findViewById(R.id.deskripsi);

        bg.setImageResource(wisata.getBGResource());
        kategori.setText(toTitleCase(wisata.getKategori()));
        nama.setText(wisata.getNama());
        deskripsi.setText(wisata.getDeskripsi());
    }

    private static String toTitleCase(String givenString) {
        String[] arr = givenString.split(" ");
        StringBuffer sb = new StringBuffer();

        for (int i = 0; i < arr.length; i++) {
            sb.append(Character.toUpperCase(arr[i].charAt(0)))
                .append(arr[i].substring(1)).append(" ");
        }
        return sb.toString().trim();
    }
}
```

```
//class ini berguna untuk menampilkan detail obyek wisata
@SuppressLint("DefaultLocale")
public class DetailFromKategoriActivity extends Activity {

    public Wisata wisata;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail);

        Bundle bundle = this.getIntent().getExtras();
        wisata = (Wisata) bundle.getParcelable("wisata");

        ImageView bg = (ImageView) findViewById(R.id.image);
        TextView kategori = (TextView) findViewById(R.id.kategori);
        TextView nama = (TextView) findViewById(R.id.nama);
        TextView deskripsi = (TextView) findViewById(R.id.deskripsi);

        bg.setImageResource(wisata.getBGResource());
        kategori.setText(toTitleCase(wisata.getKategori()));
        nama.setText(wisata.getNama());
        deskripsi.setText(wisata.getDeskripsi());
    }
}
```

```

//class ini berfungsi untuk menambatkan perhitungannya algoritma pada Google Map
public class DirectionMapActivity extends FragmentActivity {

    List<Wisata> daftar_obyek = new ArrayList<Wisata>();
    List<Wisata> daftar_obyek_kedua, daftar_obyek_ketiga, daftar_obyek_keempat;
    List<Wisata> daftar_obyek_terlewat = new ArrayList<Wisata>();
    List<Wisata> daftar_obyek_tema = new ArrayList<Wisata>();
    List<List<Wisata>> group_daftar_obyek = new ArrayList<List<Wisata>>();
    HashMap<Float, List<Wisata>> keymap_obyek_jarak = new HashMap<Float, List<Wisata>>();
    HashMap<Float, List<Wisata>> sorted_keymap_obyek_jarak = new HashMap<Float, List<Wisata>>();

    GoogleMap googleMap;
    final String TAG = "Pat:GoogleMapActivity";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_direction_map);
        SupportMapFragment fm = (SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map);
        googleMap = fm.getMap();

        // TODO: perancangan algoritma
        // -----
        // node pertama adalah obyek pertama (posisi pertama)
        Wisata node_pertama = daftar_obyek.get(0);
        // // siapkan variabel untuk menghitung total jarak
        float jarak_temp = 0;
        // menghapus node pertama dari daftar obyek
        daftar_obyek.remove(node_pertama);
        // mencari node kedua
        for (Wisata wisata_1 : daftar_obyek) {
            // siapkan variabel untuk menghitung total jarak
            float jarak_total = 0;
            // mengosongkan daftar obyek terlewat
            daftar_obyek_terlewat.clear();
            // menambahkan node pertama dalam daftar obyek terlewat
            daftar_obyek_terlewat.add(node_pertama);

```



```

// simpan node kedua dalam array
daftar_obyek_terlewatI.add(wisata_1);
// mengembalikan daftar obyek kedua, ketiga, & keempat
daftar_obyek_kedua = new ArrayList<Wisata>(daftar_obyek);
daftar_obyek_ketiga = new ArrayList<Wisata>(daftar_obyek);
daftar_obyek_keempat = new ArrayList<Wisata>(daftar_obyek);
// hapus node kedua pada daftar obyek
daftar_obyek_kedua.remove(wisata_1);
daftar_obyek_ketiga.remove(wisata_1);
daftar_obyek_keempat.remove(wisata_1);
// hitung jarak dengan node kedua
float[] distance = new float[1];
// menghitung jarak menggunakan metode WGS84
Location.distanceBetween(node_pertama.getLatitude(),
    node_pertama.getLongitude(), wisata_1.getLatitude(),
    wisata_1.getLongitude(), distance);
// menghitung jarak dalam kilometer
float jarak_pertama = distance[0] * 0.000621371192f;
// ubahlah jarak
jarak_tota1 += jarak_pertama;
// tampilkan jarak node pertama dalam console
Log.w("jarak dgn node pertama", "jarak " + node_pertama.getName()
    + " dengan " + wisata_1.getName() + " adalah "
    + jarak_pertama + " km");
// ulangi node ketiga
for (Wisata wisata_2 : daftar_obyek_kedua) {
    // simpan node ketiga dalam daftar sementara
    daftar_obyek_terlewatI.add(wisata_2);
    // hapus node ketiga pada daftar obyek
    daftar_obyek_ketiga.remove(wisata_2);
    daftar_obyek_keempat.remove(wisata_2);
    // menghitung jarak menggunakan metode WGS84
    Location.distanceBetween(wisata_1.getLatitude(),
        wisata_1.getLongitude(), wisata_2.getLatitude(),
        wisata_2.getLongitude(), distance);
    // menghitung jarak dalam kilometer
    float jarak_kedua = distance[0] * 0.000621371192f;
}

```

```

// menambahkan jarak
jarak_total += jarak_kedua;
// tampilkan jarak node kedua dalam console
Log.w("jarak dgn node kedua", "jarak " + wisata_1.getNama()
      + " dengan " + wisata_2.getNama() + " adalah "
      + jarak_kedua + " km");
// mencari node keempat
for (Wisata wisata_3 : daftar_obyek_ketiga) {
    // menyimpan node keempat dalam daftar sementara
    daftar_obyek_terlewat.add(wisata_3);
    // hapus node keempat pada daftar obyek
    daftar_obyek_keempat.remove(wisata_3);
    // menghitung jarak menggunakan metode WGS84
    Location.distanceBetween(wisata_2.getLatitude(),
                             wisata_2.getLongitude(), wisata_3.getLatitude(),
                             wisata_3.getLongitude(), distance);
    // menghitung jarak dalam Kilometer
    float jarak_ketiga = distance[0] * 0.000621371192f;
    // menambahkan jarak
    jarak_total += jarak_ketiga;
    // tampilkan jarak node keempat dalam console
    Log.w("jarak dgn node ketiga",
          "jarak " + wisata_2.getNama() + " dengan "
          + wisata_3.getNama() + " adalah "
          + jarak_ketiga + " km");
    // menambahkan sisa obyek pada daftar obyek
    for (Wisata wisata_4 : daftar_obyek_keempat) {
        // menyimpan node kelima dalam daftar sementara
        daftar_obyek_terlewat.add(wisata_4);
        // menghitung jarak menggunakan metode WGS84
        Location.distanceBetween(wisata_3.getLatitude(),
                                 wisata_3.getLongitude(),
                                 wisata_4.getLatitude(),
                                 wisata_4.getLongitude(), distance);
        // menghitung jarak dalam Kilometer
        float jarak_keempat = distance[0] * 0.000621371192f;
    }
}

```

```

// menghitung jarak dalam kilometer
float jarak_keempat = distance[0] * 0.000621371192f;
// menambahkan jarak
jarak_total += jarak_keempat;
// tampilkan jarak node kelima dalam console
Log.w("jarak dan node keempat",
    "jarak " + wisata_3.getNama() + " dengan "
    + wisata_4.getNama() + " adalah "
    + jarak_keempat + " km");
Log.w("jarak dan rute",
    "jarak = " + jarak_total + ", rute = "
    + daftar_obyek_terlewat.toString());
keymap_obyek.put(jarak_total,
    daftar_obyek_terlewat);
}
}
}
// perulangan untuk mencari rute dgn jarak terdekat
Iterator<Entry<float, List<Wisata>>> it = keymap_obyek.entrySet().iterator();
while (it.hasNext()) {
    @SuppressWarnings("rawtypes")
    Map.Entry pair = (Map.Entry)it.next();
    float jarak = (Float) pair.getKey();
    if (jarak_temp == 0 || jarak < jarak_temp) {
        // menyimpan obyek terlewat dalam daftar sementara
        daftar_obyek_temp = (List<Wisata>) pair.getValue();
        jarak_temp = jarak;
    }
    it.remove(); // menghindari ConcurrentModificationException
}
// daftar obyek terlewat ditentukan oleh jarak terdekat
daftar_obyek_terlewat = daftar_obyek_temp;
daftar_obyek_terlewat.add(node_pertama);
Log.w("daftar obyek terlewat", daftar_obyek_terlewat.toString());
//
MarkerOptions options = new MarkerOptions();
daftar_obyek.add(node_pertama);
for (Wisata wisata : daftar_obyek) {
    options.position(new LatLng(wisata.getLatitude(), wisata
        .getLongitude()));
}
googleMap.addMarker(options);
googleMap.moveCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(
    daftar_obyek.get(0).getLatitude(), daftar_obyek.get(0)
        .getLongitude()), 13));
addMarkers();
new connectAsyncTask().execute();
}

```

```

        new connectAsyncTask().execute();
    }

    private void addMarkers() {
        if (googleMap != null) {
            for (Wisata wisata : daftar_obyek) {
                googleMap.addMarker(new MarkerOptions()
                    .position(
                        new LatLng(wisata.getLatitude(), wisata
                            .getLongitude())).title(
                            wisata.toString()));
            }
        }
    }

    private String makeURI (double sourcelat, double sourcelog, double destlat,
        double destlog) {
        StringBuilder urlString = new StringBuilder();
        urlString.append("http://maps.googleapis.com/maps/api/directions/json");
        urlString.append("?origin="); // from
        urlString.append(Double.toString(sourcelat));
        urlString.append(", ");
        urlString.append(Double.toString(sourcelog));
        urlString.append("&destination="); // to
        urlString.append(Double.toString(destlat));
        urlString.append(", ");
        urlString.append(Double.toString(destlog));
        urlString.append("&sensor=false&mode=driving&alternatives=true");
        return urlString.toString();
    }
}

```

```

private class connectAsyncTask extends AsyncTask<Void, Void, List<String>> {
    private ProgressDialog progressDialog;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        progressDialog = new ProgressDialog(DirectionMapActivity.this);
        progressDialog.setMessage("Mengambil rute, mohon tunggu...");
        progressDialog.setIndeterminate(true);
        progressDialog.show();
    }

    @Override
    protected List<String> doInBackground(Void... params) {
        List<String> results = new ArrayList<String>();
        for (int i = 1; i < daftar_obyek_terlewati.size(); i++) {
            // TODO: add A* algorithm here
            JSONParser jParser = new JSONParser();
            String json = jParser.getJSONFromUrl(makeURL(
                daftar_obyek_terlewati.get(i - 1).getLatitude(),
                daftar_obyek_terlewati.get(i - 1).getLongitude(),
                daftar_obyek_terlewati.get(i).getLatitude(),
                daftar_obyek_terlewati.get(i).getLongitude()));
            results.add(json);
            Log.w("nama source", daftar_obyek_terlewati.get(i - 1)
                .getNama());
            Log.w("latlng source", ""
                + daftar_obyek_terlewati.get(i - 1).getLatitude()
                + ", "
                + daftar_obyek_terlewati.get(i - 1).getLongitude());
            Log.w("nama dest", daftar_obyek_terlewati.get(i).getNama());
            Log.w("latlng dest", ""
                + daftar_obyek_terlewati.get(i).getLatitude() + ", "
                + daftar_obyek_terlewati.get(i).getLongitude());
        }
        return results;
    }

    private void drawPath(String result, int color) {
        try {
            final JSONObject json = new JSONObject(result);
            JSONArray routeArray = json.getJSONArray("routes");
            JSONObject routes = routeArray.getJSONObject(0);
            JSONObject overviewPolylines = routes
                .getJSONObject("overview_polyline");
            String encodedString = overviewPolylines.getString("points");
            List<LatLng> list = decodePoly(encodedString);
            for (int z = 0; z < list.size() - 1; z++) {
                LatLng src = list.get(z);
                LatLng dest = list.get(z + 1);
                googleMap.addPolyline(new PolylineOptions()
                    .add(new LatLng(src.latitude, src.longitude),
                        new LatLng(dest.latitude, dest.longitude))
                    .width(7).color(color).geodesic(true));
            }
        } catch (JSONException e) {
        }
    }
}

```

```

//class ini berfungsi untuk menampilkan daftar obyek wisata berdasarkan kategori
@SuppressWarnings("DefaultLocale")
public class KategoriDetailActivity extends ListActivity {

    private MainAdapter mainAdapter;
    private List<Wisata> daftar_wisata = new ArrayList<Wisata>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_kategori_detail);

        Bundle bundle = this.getIntent().getExtras();
        daftar_wisata = bundle.getParcelableArrayList("kategori_wisata");

        mainAdapter = new MainAdapter(daftar_wisata);
        setListAdapter(mainAdapter);

        EditText search = (EditText) findViewById(R.id.search);
        search.addTextChangedListener(new TextWatcher() {
            @Override
            public void onTextChanged(CharSequence s, int start, int before,
                int count) {
                if (count == 0) {
                    setListAdapter(null);
                    mainAdapter = new MainAdapter(daftar_wisata);
                    setListAdapter(mainAdapter);
                    mainAdapter.notifyDataSetChanged();
                } else {
                    mainAdapter.getFilter().filter(s);
                }
            }
        });

        @Override
        public void beforeTextChanged(CharSequence s, int start, int count,
            int after) {
        }
    }
}

```

```

        @Override
        public void afterTextChanged(Editable s) {
        }
    });
    listView.setOnItemClickListener(new OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> arg0, View arg1,
            int position, long arg3) {
            Intent intent = new Intent(getApplicationContext(),
                DetailFromKategoriActivity.class);
            Bundle bundle = new Bundle();
            bundle.putParcelable("wisata",
                (Wisata) mainAdapter.getItem(position));
            intent.putExtras(bundle);
            startActivity(intent);
        }
    });
}

@SuppressWarnings("ViewHolder")
private class MainAdapter extends BaseAdapter implements Filterable {

    private Filter wisataFilter;
    private List<Wisata> list_wisata = new ArrayList<Wisata>();

    public MainAdapter(List<Wisata> list_wisata) {
        this.list_wisata = list_wisata;
    }

    @Override
    public int getCount() {
        return list_wisata.size();
    }

    @Override
    public Object getItem(int position) {
        return list_wisata.get(position);
    }
}

```



```

//class ini berfungsi sebagai induk dari fragment yang bertugas membuat
//Lampiran menjadi sebagai menu termasuk Beranda, Keluar, Petunjuk And, dan Tampilkan
//class ini adalah class utama
@SuppressLint("CommitPrefEdits")
public class MainActivity extends ActionBarActivity implements
    NavigationDrawerFragment.NavigationDrawerCallbacks {

    private SharedPreferences sp;
    private SharedPreferences.Editor edit;
    private NavigationDrawerFragment mNavigationDrawerFragment;
    private CharSequence mTitle;

    private List<Wisata> daftar_wisata = new ArrayList<Wisata>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        sp = getSharedPreferences("travel", MODE_PRIVATE);
        edit = sp.edit();

        MyLocation myLocation = new MyLocation();
        myLocation.getLocation(this, locationResult);

        if (CustomHttpClient.checkInternetConnection(MainActivity.this))
            new getDataWisata().execute();

        mNavigationDrawerFragment = (NavigationDrawerFragment) getSupportFragmentManager()
            .findFragmentById(R.id.navigation_drawer);
        mTitle = getTitle();

        mNavigationDrawerFragment.setup(R.id.navigation_drawer,
            (DrawerLayout) findViewById(R.id.drawer_layout));
    }
}

```

```

@Override
public void onNavigationDrawerItemSelected(int position) {
    FragmentManager fragmentManager = getSupportFragmentManager();
    switch (position) {
        case 0:
            fragmentManager
                .beginTransaction()
                .replace(R.id.container, new BerandaFragmct(daftar_wisata))
                .commit();
            break;
        case 1:
            fragmentManager
                .beginTransaction()
                .replace(R.id.container,
                    new KategoriFragment(daftar_wisata)).commit();
            break;
        case 2:
            Wisata awal = new Wisata(0, "Lokasi Pengguna", "lokasi",
                Double.parseDouble(sp.getString("lat", "0")),
                Double.parseDouble(sp.getString("lng", "0")),
                "lokasi awal pengguna aplikasi");
            fragmentManager
                .beginTransaction()
                .replace(R.id.container,
                    new PetunjukArahFragment(daftar_wisata, awal))
                .commit();
            break;
        case 3:
            fragmentManager.beginTransaction()
                .replace(R.id.container, new TambahDataFragment()).commit();
            break;
        default:
            break;
    }
}

```

```

//class ini merupakan fragment untuk menampilkan halaman search
@SuppressLint({ "DefaultLocale", "ValidFragment" })
public class BerandaFragment extends ListFragment {

    private MainAdapter mainAdapter;

    private List<Wisata> daftar_wisata = new ArrayList<Wisata>();

    public static ListFragment newInstance(List<Wisata> daftar_wisata) {
        BerandaFragment fragment = new BerandaFragment(daftar_wisata);
        return fragment;
    }

    public BerandaFragment(List<Wisata> daftar_wisata) {
        this.daftar_wisata = daftar_wisata;
    }

    public void updateList() {
        getActivity().runOnUiThread(new Runnable() {
            @Override
            public void run() {
                mainAdapter.notifyDataSetChanged();
            }
        });
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_main, container,
            false);

        mainAdapter = new MainAdapter(daftar_wisata);
        listView.setAdapter(mainAdapter);

        EditText search = (EditText) rootView.findViewById(R.id.search);
        search.addTextChangedListener(new TextWatcher() {

```

```

@Override
public void onTextChanged(CharSequence s, int start, int before,
    int count) {
    if (count == 0) {
        setListAdapter(null);
        mainAdapter = new MainAdapter(caftar_wisata);
        setListAdapter(mainAdapter);
        mainAdapter.notifyDataSetChanged();
    } else {
        mainAdapter.getFilter().filter(s);
    }
}

@Override
public void beforeTextChanged(CharSequence s, int start, int count,
    int after) {
}

@Override
public void afterTextChanged(Editable s) {
}
});
return rootView;
}

@Override
public void onActivityCreated(Bundle savedInstanceState) {
    super.onActivityCreated(savedInstanceState);
    getListView().setOnItemClickListener(new OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> arg0, View arg1,
            int position, long arg2) {
            Intent intent = new Intent(getActivity()
                .getApplicationContext(), DetailActivity.class);
            Bundle bundle = new Bundle();
            bundle.putParcelable("wisata",
                (Wisata) mainAdapter.getItem(position));
            intent.putExtras(bundle);

```

```

        startActivity(intent);
    }
}

@Override
public void onAttach(Activity activity) {
    super.onAttach(activity);
    ((Main/Activity) activity).onSectionAttached(1);
}

@SuppressWarnings("ViewHolder")
private class MainAdapter extends BaseAdapter implements Filterable {

    private Filter wisataFilter;
    private List<Wisata> list_wisata = new ArrayList<Wisata>();

    public MainAdapter(List<Wisata> list_wisata) {
        this.list_wisata = list_wisata;
    }

    @Override
    public int getCount() {
        return list_wisata.size();
    }

    @Override
    public Object getItem(int position) {
        return list_wisata.get(position);
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {

```

```

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    LayoutInflater inflater = (LayoutInflater) getActivity()
        .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    convertView = inflater.inflate(R.layout.list_item_beranda, parent,
        false);

    ImageView icon = (ImageView) convertView.findViewById(R.id.icon);
    TextView name = (TextView) convertView.findViewById(R.id.name);
    TextView desc = (TextView) convertView.findViewById(R.id.desc);
    TextView distance = (TextView) convertView
        .findViewById(R.id.distance);

    Wisata wisata = list_wisata.get(position);

    icon.setImageResource(wisata.getImageResource());
    name.setText(wisata.getNama());
    desc.setText(wisata.getDeskripsi());
    distance.setText(formatDecimal(wisata.getJarak()));
    return convertView;
}

@Override
public Filter getFilter() {
    if (wisataFilter == null) {
        wisataFilter = new WisataFilter();
    }
    return wisataFilter;
}

private class WisataFilter extends Filter {
    @Override
    protected FilterResults performFiltering(CharSequence constraint) {
        FilterResults results = new FilterResults();
        if (constraint != null && constraint.length() > 0) {
            ArrayList<Wisata> filterList = new ArrayList<Wisata>();
            for (int i = 0; i < list_wisata.size(); i++) {
                if ((list_wisata.get(i).getNama().toUpperCase())

```

```

        .contains(constraint.toString().toUpperCase())) {
            Wisata wisata = list_wisata.get(i);
            filterList.add(wisata);
        }
        results.count = filterList.size();
        results.values = filterList;
    } else {
        results.count = list_wisata.size();
        results.values = list_wisata;
    }
    return results;
}

@Override
protected void publishResults(CharSequence constraint,
    FilterResults results) {
    // Now we have to inform the adapter about the new list filtered
    if (results.count == 0)
        notifyDataSetInvalidated();
    else {
        list_wisata = (List<Wisata>) results.values;
        notifyDataSetChanged();
    }
}
}
}
}

```

```

//class ini adalah fragment untuk menampilkan daftar kategori

@SuppressLint("ValidFragment")
public class KategoriFragment extends Fragment implements OnClickListener {

    private List<Wisata> daftar_wisata = new ArrayList<Wisata>();
    private ArrayList<Wisata> kategori_wisata = new ArrayList<Wisata>();
    private TextView belanja, sejarah, alam, pantai, kuliner;

    @SuppressLint("ValidFragment")
    public KategoriFragment(List<Wisata> daftar_wisata) {
        this.daftar_wisata = daftar_wisata;
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_kategori, container,
            false);
        belanja = (TextView) rootView.findViewById(R.id.belanja);
        sejarah = (TextView) rootView.findViewById(R.id.sejarah);
        alam = (TextView) rootView.findViewById(R.id.alam);
        pantai = (TextView) rootView.findViewById(R.id.pantai);
        kuliner = (TextView) rootView.findViewById(R.id.kuliner);
        return rootView;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        belanja.setOnClickListener(this);
        sejarah.setOnClickListener(this);
        alam.setOnClickListener(this);
        pantai.setOnClickListener(this);
        kuliner.setOnClickListener(this);
    }
}

```



```

@Override
public void onAttach(Activity activity) {
    super.onAttach(activity);
    ((MainActivity) activity).onSectionAttached(2);
}

@Override
public void onClick(View v) {
    Intent intent = new Intent(getActivity().getApplicationContext(),
        kategoriDetailActivity.class);
    Bundle bundle = new Bundle();
    kategori_wisata.clear();
    if (v == belanja) {
        for (Iterator<Wisata> iterator = daftar_wisata.iterator(); iterator
            .hasNext(); ) {
            Wisata wisata = iterator.next();
            if (wisata.getKategori().equals("wisata belanja")) {
                if (!kategori_wisata.contains(wisata))
                    kategori_wisata.add(wisata);
            }
        }
    } else if (v == sejarah) {
        for (Iterator<Wisata> iterator = daftar_wisata.iterator(); iterator
            .hasNext(); ) {
            Wisata wisata = iterator.next();
            if (wisata.getKategori().equals("wisata sejarah")) {
                if (!kategori_wisata.contains(wisata))
                    kategori_wisata.add(wisata);
            }
        }
    } else if (v == alam) {
        for (Iterator<Wisata> iterator = daftar_wisata.iterator(); iterator
            .hasNext(); ) {
            Wisata wisata = iterator.next();
            if (wisata.getKategori().equals("wisata alam")) {
                if (!kategori_wisata.contains(wisata))
                    kategori_wisata.add(wisata);
            }
        }
    } else if (v == pantai) {
        for (Iterator<Wisata> iterator = daftar_wisata.iterator(); iterator
            .hasNext(); ) {
            Wisata wisata = iterator.next();
            if (wisata.getKategori().equals("wisata pantai")) {
                if (!kategori_wisata.contains(wisata))
                    kategori_wisata.add(wisata);
            }
        }
    } else if (v == kuliner) {
        for (Iterator<Wisata> iterator = daftar_wisata.iterator(); iterator
            .hasNext(); ) {
            Wisata wisata = iterator.next();
            if (wisata.getKategori().equals("wisata kuliner")) {
                if (!kategori_wisata.contains(wisata))
                    kategori_wisata.add(wisata);
            }
        }
    }
    bundle.putParcelableArrayList("kategori wisata", kategori_wisata);
    intent.putExtras(bundle);
    startActivity(intent);
}

```

```

//class ini bertugas menampilkan form untuk memilih rute perjalanan
@SuppressLint("ValidFragment")
public class PetunjukArahFragment extends Fragment {

    private List<Wisata> daftar_wisata = new ArrayList<Wisata>();
    private ArrayList<Wisata> pilihan_wisata = new ArrayList<Wisata>();
    private Wisata awal;

    private AutoCompleteTextView name1, name2, name3, name4;
    private List<AutoCompleteTextView> list_name = new ArrayList<AutoCompleteTextView>();

    public static Fragment newInstance(List<Wisata> daftar_wisata, Wisata awal) {
        PetunjukArahFragment fragment = new PetunjukArahFragment(daftar_wisata,
            awal);
        return fragment;
    }

    public PetunjukArahFragment(List<Wisata> daftar_wisata, Wisata awal) {
        this.daftar_wisata = daftar_wisata;
        this.awal = awal;
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_petunjuk_arah,
            container, false);
        rootView.findViewById(R.id.search).setOnClickListener(
            new OnClickListener() {
                @Override
                public void onClick(View v) {
                    pilihan_wisata.removeAll(Collections.singleton(null));
                    if (pilihan_wisata.isEmpty()) {
                        Toast.makeText(getActivity(),
                            "Mohon pilih minimal dua tempat.",
                            Toast.LENGTH_SHORT).show();
                    } else if (pilihan_wisata.size() < 2) {
                        Toast.makeText(getActivity(),
                            "Pilihan harus lebih dari satu.",
                            Toast.LENGTH_SHORT).show();
                    } else {
                        Intent intent = new Intent(getActivity()
                            .getApplicationContext(),
                            DirectionMapActivity.class);
                        Bundle bundle = new Bundle();
                        bundle.putParcelableArrayList("pilihan_wisata",
                            pilihan_wisata);
                        intent.putExtras(bundle);
                        startActivity(intent);
                    }
                }
            });
        rootView.findViewById(R.id.reset).setOnClickListener(
            new OnClickListener() {
                @Override
                public void onClick(View v) {
                    for (AutoCompleteTextView autoCompleteTextView : list_name) {
                        autoCompleteTextView.setText("");
                    }
                    pilihan_wisata.clear();
                }
            });
        name1 = (AutoCompleteTextView) rootView.findViewById(R.id.name1);
        name2 = (AutoCompleteTextView) rootView.findViewById(R.id.name2);
        name3 = (AutoCompleteTextView) rootView.findViewById(R.id.name3);
        name4 = (AutoCompleteTextView) rootView.findViewById(R.id.name4);
        list_name.add(name1);
        list_name.add(name2);
        list_name.add(name3);
        list_name.add(name4);
    }
}

```

```

pilihan_wisata.add(aw1);
name1.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1,
        int position, long arg3) {
        Wisata wisata = (Wisata) arg0.getAdapter().getItem(position);
        for (int i = 0; i < list_name.size(); i++) {
            pilihan_wisata.add(null);
        }
        if (pilihan_wisata.get(1) == null)
            pilihan_wisata.set(1, wisata);
    }
});
name2.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1,
        int position, long arg3) {
        Wisata wisata = (Wisata) arg0.getAdapter().getItem(position);
        for (int i = 0; i < list_name.size(); i++) {
            pilihan_wisata.add(null);
        }
        if (pilihan_wisata.get(2) == null)
            pilihan_wisata.set(2, wisata);
    }
});
name3.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1,
        int position, long arg3) {
        Wisata wisata = (Wisata) arg0.getAdapter().getItem(position);
        for (int i = 0; i < list_name.size(); i++) {
            pilihan_wisata.add(null);
        }
        if (pilihan_wisata.get(3) == null)
            pilihan_wisata.set(3, wisata);
    }
});

```

```

//class ini berfungsi untuk menambahkan data ke server
public class TambahDataFragment extends Fragment {

    private EditText nama, latitude, longitude, deskripsi;
    private Spinner kategori;
    private List<EditText> list_textbox = new ArrayList<EditText>();

    public static Fragment newInstance() {
        TambahDataFragment fragment = new TambahDataFragment();
        return fragment;
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_add_object,
            container, false);
        kategori = (Spinner) rootView.findViewById(R.id.kategori);
        nama = (EditText) rootView.findViewById(R.id.nama);
        latitude = (EditText) rootView.findViewById(R.id.latitude);
        longitude = (EditText) rootView.findViewById(R.id.longitude);
        deskripsi = (EditText) rootView.findViewById(R.id.deskripsi);
        list_textbox.add(nama);
        list_textbox.add(latitude);
        list_textbox.add(longitude);
        list_textbox.add(deskripsi);
        rootView.findViewById(R.id.add).setOnClickListener(
            new OnClickListener() {
                @Override
                public void onClick(View v) {
                    for (EditText edittext : list_textbox) {
                        if (edittext.getText().toString().trim().isEmpty()) {
                            Toast.makeText(getActivity(),
                                "Mohon isi dengan lengkap.",
                                Toast.LENGTH_SHORT).show();
                            return;
                        }
                    }
                }
            }
        );
    }
}

```

```

        new postDataWisata().execute();
    }
    });
    rootView.findViewById(R.id.reset).setOnClickListener(
        new OnClickListener() {
            @Override
            public void onClick(View v) {
                for (EditText editText : list_textbox) {
                    editText.setText("");
                }
            }
        }
    );
    return rootView;
}

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    List<String> categories = new ArrayList<String>();
    categories.add("Wisata Beleris");
    categories.add("Wisata Sejarah");
    categories.add("Wisata Alam");
    categories.add("Wisata Pantai");
    categories.add("Wisata Kuliner");

    ArrayAdapter<String> dataAdapter = new ArrayAdapter<String>(
        getActivity(), android.R.layout.simple_spinner_item, categories);
    dataAdapter
        .setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    kategori.setAdapter(dataAdapter);
}

@Override
public void onAttach(Activity activity) {
    super.onAttach(activity);
    ((MainActivity) activity).onSectionAttached(4);
}
}

```

```

private class postDataWisata extends AsyncTaskVoid, Void, String {

    private ProgressDialog pDialog;
    private boolean isSuccess = true;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        pDialog = ProgressDialog.show(getActivity(), "",
            "Mengirimkan data.", true, true);
        pDialog.setOnCancelListener(new OnCancelListener() {
            @Override
            public void onCancel(DialogInterface arg0) {
                postDataWisata.this.cancel(true);
            }
        });
    }

    @SuppressWarnings("DefaultLocale")
    @Override
    protected void doInBackground(Void... params) {
        try {
            ArrayList<NameValuePair> postDataParameters = new ArrayList<NameValuePair>();
            postDataParameters.add(new BasicNameValuePair("nama", nama
                .getText().toString().trim()));
            postDataParameters.add(new BasicNameValuePair("kategori", kategori
                .getItemAtPosition(kategori.getSelectedItemId())
                .toString()));
            postDataParameters.add(new BasicNameValuePair("latitude", latitude
                .getText().toString().trim()));
            postDataParameters.add(new BasicNameValuePair("longitude",
                longitude.getText().toString().trim()));
            postDataParameters.add(new BasicNameValuePair("deskripsi",
                deskripsi.getText().toString().trim()));
            String success = CustomHttpClient.executeHttpPost(
                CustomHttpClient.URL_ADU_ALBATA, postDataParameters);
            Log.w("success", success.trim());
            if (success.trim().equals("success"))

```

```

@SuppressLint("DefaultLocale")
@Override
protected String doInBackground(Void... params) {
    try {
        ArrayList<NameValuePair> postParameters = new ArrayList<NameValuePair>();
        postParameters.add(new BasicNameValuePair("nama", nama
            .getText().toString().trim()));
        postParameters.add(new BasicNameValuePair("kategori", kategori
            .getItemAtPosition(kategori.getSelectedItemId())
            .toString()));
        postParameters.add(new BasicNameValuePair("latitude", latitude
            .getText().toString().trim()));
        postParameters.add(new BasicNameValuePair("longitude",
            longitude.getText().toString().trim()));
        postParameters.add(new BasicNameValuePair("deskripsi",
            deskripsi.getText().toString().trim()));
        String success = CustomHttpClient.executeHttpPost(
            CustomHttpClient.URL_ADD_MTSATA, postParameters);
        Log.w("success", success.trim());
        if (success.trim().equals("success"))
            isSuccess = true;
        else
            isSuccess = false;
    } catch (Exception e) {
        e.printStackTrace();
        isSuccess = false;
    }
    return null;
}

@Override
protected void onPostExecute(String result) {
    super.onPostExecute(result);
    progressDialog.dismiss();
    if (!isSuccess) {
        Toast.makeText(getActivity(), "Data gagal dikirimkan!",
            Toast.LENGTH_SHORT).show();
    }

    @Override
    protected void onPostExecute(String result) {
        super.onPostExecute(result);
        progressDialog.dismiss();
        if (!isSuccess) {
            Toast.makeText(getActivity(), "Data gagal dikirimkan!",
                Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(getActivity(), "Data berhasil dikirimkan!",
                Toast.LENGTH_SHORT).show();
            for (EditText edittext : list_textbox) {
                edittext.setText("");
            }
        }
    }
}
}
}
}

```

```

//class ini memiliki attribute latitude dan longitude serta address sebagai
public class MyLocation {
    Timer timer;
    LocationManager lm;
    LocationResult locationResult;
    boolean gps_enabled = false;
    boolean network_enabled = false;

    public boolean getLocation(Context context, LocationResult result) {
        // I use LocationResult callback class to pass location value from
        // MyLocation to user code.
        locationResult = result;
        if (lm == null)
            lm = (LocationManager) context
                .getSystemService(Context.LOCATION_SERVICE);

        // exceptions will be thrown if provider is not permitted.
        try {
            gps_enabled = lm.isProviderEnabled(LocationManager.GPS_PROVIDER);
        } catch (Exception ex) {
            ex.printStackTrace();
        }
        try {
            network_enabled = lm
                .isProviderEnabled(LocationManager.NETWORK_PROVIDER);
        } catch (Exception ex) {
            ex.printStackTrace();
        }

        // don't start listeners if no provider is enabled
        if (!gps_enabled && !network_enabled)
            return false;

        if (gps_enabled)
            lm.requestLocationUpdates(LocationManager.GPS_PROVIDER, 5000, 10,
                locationListenerGps);
        if (network_enabled)
            lm.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 5000, 10,

```



```

        locationListenerNetwork);
        timer1 = new Timer();
        timer1.schedule(new GetLastLocation(), 5000);
        return true;
    }

    LocationListener locationListeners = new LocationListener() {
        @Override
        public void onLocationChanged(Location location) {
            timer1.cancel();
            locationResult.getLocation(location);
            lm.removeUpdates(this);
            lm.removeUpdates(locationListenerNetwork);
        }

        @Override
        public void onProviderDisabled(String provider) {
        }

        @Override
        public void onProviderEnabled(String provider) {
        }

        @Override
        public void onStatusChanged(String provider, int status, Bundle extras) {
        }
    };

    LocationListener locationListenerNetwork = new LocationListener() {
        @Override
        public void onLocationChanged(Location location) {
            timer1.cancel();
            locationResult.getLocation(location);
            lm.removeUpdates(this);
            lm.removeUpdates(locationListeners);
        }
    }

```

```

@Override
public void onProviderDisabled(String provider) {
}

@Override
public void onProviderEnabled(String provider) {
}

@Override
public void onStatusChanged(String provider, int status, Bundle extras) {
}
};

LocationListener locationListenerNetwork = new LocationListener() {
@Override
public void onLocationChanged(Location location) {
    timer.cancel();
    locationManager.getLastLocation();
    locationManager.removeUpdates(this);
    locationManager.removeUpdates(locationListenerGps);
}

@Override
public void onProviderDisabled(String provider) {
}

@Override
public void onProviderEnabled(String provider) {
}

@Override
public void onStatusChanged(String provider, int status, Bundle extras) {
}
};

```

```

class GetLastLocation extends TimerTask {
    @Override
    public void run() {
        lm.removeUpdates(locationListenerGps);
        lm.removeUpdates(locationListenerNetwork);

        Location net_loc = null, gps_loc = null;
        if (gps_enabled)
            gps_loc = lm.getLastKnownLocation(LocationManager.GPS_PROVIDER);
        if (network_enabled)
            net_loc = lm
                .getLastKnownLocation(LocationManager.NETWORK_PROVIDER);

        // if there are both values use the latest one
        if (gps_loc != null && net_loc != null) {
            if (gps_loc.getTime() > net_loc.getTime())
                locationResult.getLocation(gps_loc);
            else
                locationResult.getLocation(net_loc);
            return;
        }

        if (gps_loc != null) {
            locationResult.getLocation(gps_loc);
            return;
        }
        if (net_loc != null) {
            locationResult.getLocation(net_loc);
            return;
        }
        locationResult.getLocation(null);
    }
}

public static abstract class LocationResult {
    public abstract void getLocation(Location location);
}
}

```

```

//class ini digurakan sebagai model dalam aplikasi
public class Wisata implements Parcelable {

    int id;
    String nama;
    String kategori;
    double latitude;
    double longitude;
    String deskripsi;
    float jarak;
    float bobot;

    public Wisata(Parcel in) {
        super();
        readFromParcel(in);
    }

    public Wisata(int id, String nama, String kategori, double latitude,
        double longitude, String deskripsi) {
        super();
        this.id = id;
        this.nama = nama;
        this.kategori = kategori;
        this.latitude = latitude;
        this.longitude = longitude;
        this.deskripsi = deskripsi;
    }

    @Override
    public String toString() {
        return nama;
    }

    public int getId() {
        return id;
    }
}

```

```
public void setId(int id) {
    this.id = id;
}

public String getName() {
    return nama;
}

public void setName(String nama) {
    this.nama = nama;
}

public String getKategori() {
    return kategori;
}

public void setKategori(String kategori) {
    this.kategori = kategori;
}

public double getLatitude() {
    return latitude;
}

public void setLatitude(double latitude) {
    this.latitude = latitude;
}

public double getLongitude() {
    return longitude;
}

public void setLongitude(double longitude) {
    this.longitude = longitude;
}

public String getDeskripsi() {
    return deskripsi;
}
```

```

    }

    public void setDescription(String deskripsi) {
        this.deskripsi = deskripsi;
    }

    public float getJarak() {
        return jarak;
    }

    public void setJarak(float jarak) {
        this.jarak = jarak;
    }

    public float getBobot() {
        return bobot;
    }

    public void setBobot(float bobot) {
        this.bobot = bobot;
    }

    public int getImageResource() {
        int resource = R.drawable.logo_icon;
        if (kategori.equals("wisata pantai")) {
            resource = R.drawable.dot_blue;
        } else if (kategori.equals("wisata belanja")) {
            resource = R.drawable.dot_purple;
        } else if (kategori.equals("wisata sejarah")) {
            resource = R.drawable.dot_yellow;
        } else if (kategori.equals("wisata alam")) {
            resource = R.drawable.dot_green;
        } else if (kategori.equals("wisata kuliner")) {
            resource = R.drawable.dot_red;
        }
        return resource;
    }
}

```

```

public int getBkresource() {
    int resource = R.drawable.logo_icons;
    if (kategori.equals("wisata pantai")) {
        resource = R.drawable.bg_pantai;
    } else if (kategori.equals("wisata belanja")) {
        resource = R.drawable.bg_belanja;
    } else if (kategori.equals("wisata sejarah")) {
        resource = R.drawable.bg_sejarah;
    } else if (kategori.equals("wisata alam")) {
        resource = R.drawable.bg_alam;
    } else if (kategori.equals("wisata kuliner")) {
        resource = R.drawable.bg_kuliner;
    }
    return resource;
}

public static final Parcelable.Creator<Wisata> CREATOR = new Parcelable.Creator<Wisata>() {
    @Override
    public Wisata createFromParcel(Parcel in) {
        return new Wisata(in);
    }

    @Override
    public Wisata[] newArray(int size) {
        return new Wisata[size];
    }
};

public void readFromParcel(Parcel in) {
    id = in.readInt();
    nama = in.readString();
    kategori = in.readString();
    latitude = in.readDouble();
    longitude = in.readDouble();
    deskripsi = in.readString();
}

@Override
public int describeContents() {
    return 0;
}

@Override
public void writeToParcel(Parcel dest, int flags) {
    dest.writeInt(id);
    dest.writeString(nama);
    dest.writeString(kategori);
    dest.writeDouble(latitude);
    dest.writeDouble(longitude);
    dest.writeString(deskripsi);
}
}
}

```

Tabel 4.1 Daftar *TravelAgent*

LAMPIRAN B
DAFTAR TRAVELAGENT

WISATA BELANJA

NO	NAMA LOKASI	LATITUDE	LONGITUDE
1	Malioboro	-7.793533	110.365693
2	Pasar Beringharjo	-7.769004	110.390696
3	NR Silver	-7.826886	110.400251
4	Kerajinan Gerabah Kasongan	-7.845838	110.338366
5	Pasar Seni Gabusan	-7.877261	110.350152
6	Sagio Puppet Handicraft	-7.839333	110.310708
7	AS Java Leather	-7.81922	110.371544
8	Diamond Baru	-7.774033	110.361062
9	Gading Craft	-7.756068	110.399594
10	Viko Collection	-7.828561	110.351142

WISATA SEJARAH

NO	NAMA LOKASI	LATITUDE	LONGITUDE
1	Keraton Yogyakarta	-7.805284	110.364203
2	Tugu Jogja	-7.782984	110.367035
3	Monumen Jogja Kembali	-7.760298	110.36067
4	Pabrik Gula Madukismo	-7.826856	110.34514
5	Masjid Kotagede	-7.827068	110.401631
6	Gua Jepang	-7.808227	110.47386
7	Candi Prambanan	-7.751919	110.492006
8	Ramayana Ballet	-7.752005	110.488649
9	Museum Affandi	-7.782713	110.396397
10	Museum Sonobudoyo	-7.80224	110.363961
11	Gereja Hati Kudus Tuhan Yesus Ganjuran	-7.926318	110.318963
12	Museum Sasmitaloka	-7.802433	110.375485
13	Gua Maria Tritis	-8.082652	110.556704

Tabel 4.1 Daftar *TravelAgent*

14	Museum Sasana Wiratama	-7.787119	110.351405
15	Sendang Sriningsih	-7.798858	110.525083

WISATA ALAM

NO	NAMA LOKASI	LATITUDE	LONGITUDE
1	Hutan Pinus Mangunan	-7.925085	110.461959
2	Gunung Nglanggeran (Gunung Api Purba)	-7.840556	110.545
3	Kaliurang	-7.590846	110.385991
4	Air Terjun Sri Gethuk	-7.963855	110.51384
5	Wisata Alam Kali Biru	-7.805862	110.128008
6	Waduk Sermo	-7.825324	110.122939
7	Gunung Merapi	-7.556267	110.438125
8	Goa Pindul	-7.92953	110.648955
9	Cave Tubing Kalisuci	-8.011531	110.638257
10	Air Terjun LEPO Dlingo	-7.949634	110.471727
11	Embung Batara Sriten	-7.832139	110.631909
12	Taman Sungai Mudal	-7.763955	110.115626
13	Kedung Pedut	-7.753982	110.175998
14	Kebun Teh Ngglinggo	-7.652552	110.145746
15	Goa Kiskendo	-7.74693	110.130872
16	Gua Jomblang	-7.963298	110.602429
17	Selokan Mataram	-7.765717	110.379617

WISATA PANTAI

NO	NAMA LOKASI	LATITUDE	LONGITUDE
1	Pantai Parangtritis	-8.010057	110.313009

Tabel 4.1 Daftar *TravelAgent*

2	Pantai Ngobaran	-8.119502	110.505849
3	Pantai Drini	-8.138452	110.57752
4	Pantai Indrayanti	-8.15022	110.612004
5	Pantai Pok Tunggal	-8.155447	110.621757
6	Pantai Nglambor	-8.182703	110.67924
7	Pantai Kesirat	-8.096154	110.435048
8	Pantai Sadranan	-8.145468	110.604108
9	Pantai Jogan	-8.180114	110.676363
10	Pantai Glagah	-7.916242	110.075748
11	Pantai Krakal	-8.145316	110.599048
12	Pantai Siung	-8.181754	110.68312
13	Pantai Baron	-8.12886	110.548746
14	Pantai Depok	-8.016111	110.300278
15	Pantai Wedi Ombo	-8.184896	110.709102
16	Pantai Sadeng	-8.188879	110.799028
17	Pantai Ngrenehan	-8.121147	110.514234
18	Pantai Sundak	-8.147082	110.607932
19	Pantai Parangkusumo	-8.017409	110.30987
20	Pantai Sepanjang	-8.136581	110.565707

WISATA KULINER

NO	NAMA LOKASI	LATITUDE	LONGITUDE
1	Gudeg Wijilan	-7.805014	110.366621
2	Angkringan Lik Man	-7.788819	110.364308
3	Soto Sulung Stasiun Tugu	-7.789618	110.362858
4	Bakpia Pathuk	-7.797593	110.357532

Tabel 4.1 Daftar *TravelAgent*

5	Oseng-oseng Mercon Bu Narti	-7.801099	110.357784
6	Banyu Mili Resto	-7.775769	110.346323
7	The Waroeng of Raminten	-7.676003	110.41718
8	Ayam Bakar Klaten Miroso	-7.780468	110.322804
9	Manggar Manding	-7.895992	110.352843
10	Lotek Teteg	-7.79001	110.380433
11	Entok Slenget	-7.646514	110.385935
12	Ayam goreng Mbah Cemplung	-7.83949	110.334078
13	Bakmi Shibitsu	-7.833481	110.353308
14	Bakpiapia	-7.793912	110.367732
15	Bong Kopitown	-7.781229	110.377413

LAMPIRAN C

LEMBAR ANKET PENGUJIAN APLIKASI

Petunjuk pengisian :

1. Tulislah nama dan pekerjaan anda pada tempat yang telah disediakan.
2. Berilah tanda centang () pada kolom jawaban yang anda pilih, yaitu Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS), Sangat Tidak Setuju (STS)

Nama : Ambarwati Nur Jannah S. Hi

Pekerjaan : Pegawai Swasta

1. Pengujian Fungsionalitas Sistem

No	Pertanyaan	Penilaian	
		Ya	Tidak
1	Apakah aplikasi dapat menunjukkan posisi pengguna dengan akurat ?		-
2	Apakah aplikasi dapat menunjukkan lokasi tempat wisata yang dipilih?		-
3	Apakah aplikasi dapat menampilkan daftar lokasi tempat wisata terdekat dari posisi pengguna ?		-
4	Apakah aplikasi dapat memberikan informasi yang jelas mengenai tempat wisata pada pengguna ?		-
5	Apakah aplikasi dapat membantu pengguna untuk mencari lokasi tempat wisata ? ¹⁴²		-

6	Apakah aplikasi dapat menunjukkan rute dari lokasi pengguna menuju tempat wisata yang dipilih ?		-
7	Apakah fungsi tambah data pada aplikasi berjalan dengan baik ?		-
8	Apakah aplikasi dapat menunjukkan daftar tempat wisata berdasarkan kategori tokonya?		-
9	Apakah fungsi pencarian tempat wisata pada aplikasi berjalan dengan baik ?		-
Total		9	0

2. Pengujian Antarmuka Sistem

No	Pertanyaan	Penilaian				
		SS	S	N	TS	STS
1	Apakah aplikasi memiliki tampilan yang menarik bagi pengguna?	-		-	-	-
2	Apakah aplikasi mudah digunakan oleh pengguna ?	-		-	-	-
3	Apakah aplikasi menggunakan bahasa yang mudah di mengerti ?	-		-	-	-
4	Apakah informasi yang ditampilkan aplikasi dapat ditangkap dengan mudah dan jelas ?	-		-	-	-

5	Apakah penggunaan komposisi warna pada tampilan aplikasi sudah sesuai ?	-	-		-	-
6	Apakah penggunaan ikon menu pada aplikasi sudah sesuai?	-	-		-	-
Total		0	5	1	0	0



LEMBAR ANKET PENGUJIAN APLIKASI

Petunjuk pengisian :

3. Tulislah nama dan pekerjaan anda pada tempat yang telah disediakan.
4. Berilah tanda centang () pada kolom jawaban yang anda pilih, yaitu Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS), Sangat Tidak Setuju (STS)

Nama : Arief Fakhrudin

Pekerjaan : Wiraswasta

3. Pengujian Fungsionalitas Sistem

No	Pertanyaan	Penilaian	
		Ya	Tidak
1	Apakah aplikasi dapat menunjukkan posisi pengguna dengan akurat ?		-
2	Apakah aplikasi dapat menunjukkan lokasi tempat wisata yang dipilih?		-
3	Apakah aplikasi dapat menampilkan daftar lokasi tempat wisata terdekat dari posisi pengguna ?		-
4	Apakah aplikasi dapat memberikan informasi yang jelas mengenai tempat wisata pada pengguna ?		-
5	Apakah aplikasi dapat membantu pengguna untuk mencari lokasi tempat wisata ?		-
6	Apakah aplikasi dapat menunjukkan rute dari lokasi		-

	pengguna menuju tempat wisata yang dipilih ?		
7	Apakah fungsi tambah data pada aplikasi berjalan dengan baik ?		-
8	Apakah aplikasi dapat menunjukkan daftar tempat wisata berdasarkan kategori tokonya?		-
9	Apakah fungsi pencarian tempat wisata pada aplikasi berjalan dengan baik ?		-
Total		9	0

4. Pengujian Antarmuka Sistem

No	Pertanyaan	Penilaian				
		SS	S	N	TS	STS
1	Apakah aplikasi memiliki tampilan yang menarik bagi pengguna?	-		-	-	-
2	Apakah aplikasi mudah digunakan oleh pengguna ?	-		-	-	-
3	Apakah aplikasi menggunakan bahasa yang mudah di mengerti ?	-		-	-	-
4	Apakah informasi yang ditampilkan aplikasi dapat ditangkap dengan mudah dan jelas ?	-		-	-	-
5	Apakah penggunaan komposisi warna pada tampilan aplikasi sudah sesuai ?	-	-		-	-

6	Apakah penggunaan ikon menu pada aplikasi sudah sesuai?	-		-	-	-
Total		0	6	0	0	0



LEMBAR ANKET PENGUJIAN APLIKASI

Petunjuk pengisian :

5. Tulislah nama dan pekerjaan anda pada tempat yang telah disediakan.
6. Berilah tanda centang () pada kolom jawaban yang anda pilih, yaitu Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS), Sangat Tidak Setuju (STS)

Nama : Arie Wijayanti

Pekerjaan : Mahasiswi

5. Pengujian Fungsionalitas Sistem

No	Pertanyaan	Penilaian	
		Ya	Tidak
1	Apakah aplikasi dapat menunjukkan posisi pengguna dengan akurat ?		-
2	Apakah aplikasi dapat menunjukkan lokasi tempat wisata yang dipilih?		-
3	Apakah aplikasi dapat menampilkan daftar lokasi tempat wisata terdekat dari posisi pengguna ?		-
4	Apakah aplikasi dapat memberikan informasi yang jelas mengenai tempat wisata pada pengguna ?		-
5	Apakah aplikasi dapat membantu pengguna untuk mencari lokasi tempat wisata ?		-
6	Apakah aplikasi dapat menunjukkan rute dari lokasi		-

	pengguna menuju tempat wisata yang dipilih ?		
7	Apakah fungsi tambah data pada aplikasi berjalan dengan baik ?		-
8	Apakah aplikasi dapat menunjukkan daftar tempat wisata berdasarkan kategori tokonya?		-
9	Apakah fungsi pencarian tempat wisata pada aplikasi berjalan dengan baik ?		-
Total		9	0

6. Pengujian Antarmuka Sistem

No	Pertanyaan	Penilaian				
		SS	S	N	TS	STS
1	Apakah aplikasi memiliki tampilan yang menarik bagi pengguna?	-		-	-	-
2	Apakah aplikasi mudah digunakan oleh pengguna ?	-		-	-	-
3	Apakah aplikasi menggunakan bahasa yang mudah di mengerti ?	-	-		-	-
4	Apakah informasi yang ditampilkan aplikasi dapat ditangkap dengan mudah dan jelas ?	-		-	-	-
5	Apakah penggunaan komposisi warna pada tampilan aplikasi sudah sesuai ?	-		-	-	-

6	Apakah penggunaan ikon menu pada aplikasi sudah sesuai?	-		-	-	-
Total		0	5	1	0	0



LEMBAR ANKET PENGUJIAN APLIKASI

Petunjuk pengisian :

- Tuliskan nama dan pekerjaan anda pada tempat yang telah disediakan.
- Berilah tanda centang () pada kolom jawaban yang anda pilih, yaitu Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS), Sangat Tidak Setuju (STS)

Nama : Edo Manaf

Pekerjaan : Pegawai Swasta

7. Pengujian Fungsionalitas Sistem

No	Pertanyaan	Penilaian	
		Ya	Tidak
1	Apakah aplikasi dapat menunjukkan posisi pengguna dengan akurat ?		-
2	Apakah aplikasi dapat menunjukkan lokasi tempat wisata yang dipilih?		-
3	Apakah aplikasi dapat menampilkan daftar lokasi tempat wisata terdekat dari posisi pengguna ?		-
4	Apakah aplikasi dapat memberikan informasi yang jelas mengenai tempat wisata pada pengguna ?		-
5	Apakah aplikasi dapat membantu pengguna untuk mencari lokasi tempat wisata ?		-
6	Apakah aplikasi dapat menunjukkan rute dari lokasi		-

	pengguna menuju tempat wisata yang dipilih ?		
7	Apakah fungsi tambah data pada aplikasi berjalan dengan baik ?		-
8	Apakah aplikasi dapat menunjukkan daftar tempat wisata berdasarkan kategori tokonya?		-
9	Apakah fungsi pencarian tempat wisata pada aplikasi berjalan dengan baik ?		-
Total		9	0

8. Pengujian Antarmuka Sistem

No	Pertanyaan	Penilaian				
		SS	S	N	TS	STS
1	Apakah aplikasi memiliki tampilan yang menarik bagi pengguna?	-		-	-	-
2	Apakah aplikasi mudah digunakan oleh pengguna ?	-		-	-	-
3	Apakah aplikasi menggunakan bahasa yang mudah di mengerti ?	-		-	-	-
4	Apakah informasi yang ditampilkan aplikasi dapat ditangkap dengan mudah dan jelas ?	-		-	-	-
5	Apakah penggunaan komposisi warna pada tampilan aplikasi sudah sesuai ?	-		-	-	-

6	Apakah penggunaan ikon menu pada aplikasi sudah sesuai?	-		-	-	-
Total		0	6	0	0	0



LEMBAR ANKET PENGUJIAN APLIKASI

Petunjuk pengisian :

9. Tulislah nama dan pekerjaan anda pada tempat yang telah disediakan.
10. Berilah tanda centang () pada kolom jawaban yang anda pilih, yaitu Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS), Sangat Tidak Setuju (STS)

Nama : Edo Surya Putra

Pekerjaan : Wiraswasta

9. Pengujian Fungsionalitas Sistem

No	Pertanyaan	Penilaian	
		Ya	Tidak
1	Apakah aplikasi dapat menunjukkan posisi pengguna dengan akurat ?		-
2	Apakah aplikasi dapat menunjukkan lokasi tempat wisata yang dipilih?		-
3	Apakah aplikasi dapat menampilkan daftar lokasi tempat wisata terdekat dari posisi pengguna ?		-
4	Apakah aplikasi dapat memberikan informasi yang jelas mengenai tempat wisata pada pengguna ?		-
5	Apakah aplikasi dapat membantu pengguna untuk mencari lokasi tempat wisata ?		-
6	Apakah aplikasi dapat menunjukkan rute dari lokasi		-

	pengguna menuju tempat wisata yang dipilih ?		
7	Apakah fungsi tambah data pada aplikasi berjalan dengan baik ?		-
8	Apakah aplikasi dapat menunjukkan daftar tempat wisata berdasarkan kategori tokonya?		-
9	Apakah fungsi pencarian tempat wisata pada aplikasi berjalan dengan baik ?		-
Total		9	0

10. Pengujian Antarmuka Sistem

No	Pertanyaan	Penilaian				
		SS	S	N	TS	STS
1	Apakah aplikasi memiliki tampilan yang menarik bagi pengguna?	-		-	-	-
2	Apakah aplikasi mudah digunakan oleh pengguna ?	-		-	-	-
3	Apakah aplikasi menggunakan bahasa yang mudah di mengerti ?	-		-	-	-
4	Apakah informasi yang ditampilkan aplikasi dapat ditangkap dengan mudah dan jelas ?	-		-	-	-
5	Apakah penggunaan komposisi warna pada tampilan aplikasi sudah sesuai ?	-		-	-	-

6	Apakah penggunaan ikon menu pada aplikasi sudah sesuai?	-		-	-	-
Total		0	6	0	0	0



LEMBAR ANGKET PENGUJIAN APLIKASI

Petunjuk pengisian :

11. Tulislah nama dan pekerjaan anda pada tempat yang telah disediakan.

12. Berilah tanda centang () pada kolom jawaban yang anda pilih, yaitu Sangat Setuju

(SS), Setuju (S), Netral (N), Tidak Setuju (TS), Sangat Tidak Setuju (STS)

Nama : Erfan Fahmi Hakim

Pekerjaan : Mahasiswa

11. Pengujian Fungsionalitas Sistem

No	Pertanyaan	Penilaian	
		Ya	Tidak
1	Apakah aplikasi dapat menunjukkan posisi pengguna dengan akurat ?		-
2	Apakah aplikasi dapat menunjukkan lokasi tempat wisata yang dipilih?		-
3	Apakah aplikasi dapat menampilkan daftar lokasi tempat wisata terdekat dari posisi pengguna ?		-
4	Apakah aplikasi dapat memberikan informasi yang jelas mengenai tempat wisata pada pengguna ?		-
5	Apakah aplikasi dapat membantu pengguna untuk mencari lokasi tempat wisata ?		-
6	Apakah aplikasi dapat menunjukkan rute dari lokasi		-

	pengguna menuju tempat wisata yang dipilih ?		
7	Apakah fungsi tambah data pada aplikasi berjalan dengan baik ?		-
8	Apakah aplikasi dapat menunjukkan daftar tempat wisata berdasarkan kategori tokonya?		-
9	Apakah fungsi pencarian tempat wisata pada aplikasi berjalan dengan baik ?		-
Total			

12. Pengujian Antarmuka Sistem

No	Pertanyaan	Penilaian				
		SS	S	N	TS	STS
1	Apakah aplikasi memiliki tampilan yang menarik bagi pengguna?	-		-	-	-
2	Apakah aplikasi mudah digunakan oleh pengguna ?	-		-	-	-
3	Apakah aplikasi menggunakan bahasa yang mudah di mengerti ?	-	-		-	-
4	Apakah informasi yang ditampilkan aplikasi dapat ditangkap dengan mudah dan jelas ?	-	-		-	-
5	Apakah penggunaan komposisi warna pada tampilan aplikasi sudah sesuai ?	-		-	-	-

6	Apakah penggunaan ikon menu pada aplikasi sudah sesuai?	-		-	-	-
Total		0	4	2	0	0



LEMBAR ANKET PENGUJIAN APLIKASI

Petunjuk pengisian :

13. Tulislah nama dan pekerjaan anda pada tempat yang telah disediakan.

14. Berilah tanda centang () pada kolom jawaban yang anda pilih, yaitu Sangat Setuju

(SS), Setuju (S), Netral (N), Tidak Setuju (TS), Sangat Tidak Setuju (STS)

Nama : Fiyadillah

Pekerjaan : Mahasiswa

13. Pengujian Fungsionalitas Sistem

No	Pertanyaan	Penilaian	
		Ya	Tidak
1	Apakah aplikasi dapat menunjukkan posisi pengguna dengan akurat ?		-
2	Apakah aplikasi dapat menunjukkan lokasi tempat wisata yang dipilih?		-
3	Apakah aplikasi dapat menampilkan daftar lokasi tempat wisata terdekat dari posisi pengguna ?		-
4	Apakah aplikasi dapat memberikan informasi yang jelas mengenai tempat wisata pada pengguna ?		-
5	Apakah aplikasi dapat membantu pengguna untuk mencari lokasi tempat wisata ?		-
6	Apakah aplikasi dapat menunjukkan rute dari lokasi		-

	pengguna menuju tempat wisata yang dipilih ?		
7	Apakah fungsi tambah data pada aplikasi berjalan dengan baik ?		-
8	Apakah aplikasi dapat menunjukkan daftar tempat wisata berdasarkan kategori tokonya?		-
9	Apakah fungsi pencarian tempat wisata pada aplikasi berjalan dengan baik ?		-
Total		9	0

14. Pengujian Antarmuka Sistem

No	Pertanyaan	Penilaian				
		SS	S	N	TS	STS
1	Apakah aplikasi memiliki tampilan yang menarik bagi pengguna?	-	-	-	-	-
2	Apakah aplikasi mudah digunakan oleh pengguna ?	-	-	-	-	-
3	Apakah aplikasi menggunakan bahasa yang mudah di mengerti ?	-	-	-	-	-
4	Apakah informasi yang ditampilkan aplikasi dapat ditangkap dengan mudah dan jelas ?	-	-	-	-	-
5	Apakah penggunaan komposisi warna pada tampilan aplikasi sudah sesuai ?	-	-	-	-	-

6	Apakah penggunaan ikon menu pada aplikasi sudah sesuai?	-		-	-	-
Total		0	5	1	0	0



LEMBAR ANKET PENGUJIAN APLIKASI

Petunjuk pengisian :

15. Tulislah nama dan pekerjaan anda pada tempat yang telah disediakan.

16. Berilah tanda centang () pada kolom jawaban yang anda pilih, yaitu Sangat Setuju

(SS), Setuju (S), Netral (N), Tidak Setuju (TS), Sangat Tidak Setuju (STS)

Nama : Ilman Fahmi

Pekerjaan : Mahasiswa

15. Pengujian Fungsionalitas Sistem

No	Pertanyaan	Penilaian	
		Ya	Tidak
1	Apakah aplikasi dapat menunjukkan posisi pengguna dengan akurat ?		-
2	Apakah aplikasi dapat menunjukkan lokasi tempat wisata yang dipilih?		-
3	Apakah aplikasi dapat menampilkan daftar lokasi tempat wisata terdekat dari posisi pengguna ?		-
4	Apakah aplikasi dapat memberikan informasi yang jelas mengenai tempat wisata pada pengguna ?		-
5	Apakah aplikasi dapat membantu pengguna untuk mencari lokasi tempat wisata ?		-
6	Apakah aplikasi dapat menunjukkan rute dari lokasi		-

	pengguna menuju tempat wisata yang dipilih ?		
7	Apakah fungsi tambah data pada aplikasi berjalan dengan baik ?		-
8	Apakah aplikasi dapat menunjukkan daftar tempat wisata berdasarkan kategori tokonya?		-
9	Apakah fungsi pencarian tempat wisata pada aplikasi berjalan dengan baik ?		-
Total		9	0

16. Pengujian Antarmuka Sistem

No	Pertanyaan	Penilaian				
		SS	S	N	TS	STS
1	Apakah aplikasi memiliki tampilan yang menarik bagi pengguna?	-		-	-	-
2	Apakah aplikasi mudah digunakan oleh pengguna ?	-		-	-	-
3	Apakah aplikasi menggunakan bahasa yang mudah di mengerti ?	-		-	-	-
4	Apakah informasi yang ditampilkan aplikasi dapat ditangkap dengan mudah dan jelas ?	-		-	-	-
5	Apakah penggunaan komposisi warna pada tampilan aplikasi sudah sesuai ?	-		-	-	-

6	Apakah penggunaan ikon menu pada aplikasi sudah sesuai?	-		-	-	-
Total		0	6	0	0	0



LEMBAR ANKET PENGUJIAN APLIKASI

Petunjuk pengisian :

17. Tulislah nama dan pekerjaan anda pada tempat yang telah disediakan.

18. Berilah tanda centang () pada kolom jawaban yang anda pilih, yaitu Sangat Setuju

(SS), Setuju (S), Netral (N), Tidak Setuju (TS), Sangat Tidak Setuju (STS)

Nama : Isn Oktariani

Pekerjaan : Karyawan Swasta

17. Pengujian Fungsionalitas Sistem

No	Pertanyaan	Penilaian	
		Ya	Tidak
1	Apakah aplikasi dapat menunjukkan posisi pengguna dengan akurat ?		-
2	Apakah aplikasi dapat menunjukkan lokasi tempat wisata yang dipilih?		-
3	Apakah aplikasi dapat menampilkan daftar lokasi tempat wisata terdekat dari posisi pengguna ?		-
4	Apakah aplikasi dapat memberikan informasi yang jelas mengenai tempat wisata pada pengguna ?		-
5	Apakah aplikasi dapat membantu pengguna untuk mencari lokasi tempat wisata ?		-
6	Apakah aplikasi dapat menunjukkan rute dari lokasi		-

	pengguna menuju tempat wisata yang dipilih ?		
7	Apakah fungsi tambah data pada aplikasi berjalan dengan baik ?		-
8	Apakah aplikasi dapat menunjukkan daftar tempat wisata berdasarkan kategori tokonya?		-
9	Apakah fungsi pencarian tempat wisata pada aplikasi berjalan dengan baik ?		-
Total		9	0

18. Pengujian Antarmuka Sistem

No	Pertanyaan	Penilaian				
		SS	S	N	TS	STS
1	Apakah aplikasi memiliki tampilan yang menarik bagi pengguna?	-		-	-	-
2	Apakah aplikasi mudah digunakan oleh pengguna ?	-		-	-	-
3	Apakah aplikasi menggunakan bahasa yang mudah di mengerti ?	-	-		-	-
4	Apakah informasi yang ditampilkan aplikasi dapat ditangkap dengan mudah dan jelas ?	-		-	-	-
5	Apakah penggunaan komposisi warna pada tampilan aplikasi sudah sesuai ?	-		-	-	-

6	Apakah penggunaan ikon menu pada aplikasi sudah sesuai?	-	-		-	-
Total		0	4	2	0	0



LEMBAR ANKET PENGUJIAN APLIKASI

Petunjuk pengisian :

19. Tulislah nama dan pekerjaan anda pada tempat yang telah disediakan.

20. Berilah tanda centang () pada kolom jawaban yang anda pilih, yaitu Sangat Setuju

(SS), Setuju (S), Netral (N), Tidak Setuju (TS), Sangat Tidak Setuju (STS)

Nama : Zainul As'ari

Pekerjaan : Mahasiswa

19. Pengujian Fungsionalitas Sistem

No	Pertanyaan	Penilaian	
		Ya	Tidak
1	Apakah aplikasi dapat menunjukkan posisi pengguna dengan akurat ?		-
2	Apakah aplikasi dapat menunjukkan lokasi tempat wisata yang dipilih?		-
3	Apakah aplikasi dapat menampilkan daftar lokasi tempat wisata terdekat dari posisi pengguna ?		-
4	Apakah aplikasi dapat memberikan informasi yang jelas mengenai tempat wisata pada pengguna ?		-
5	Apakah aplikasi dapat membantu pengguna untuk mencari lokasi tempat wisata ?		-
6	Apakah aplikasi dapat menunjukkan rute dari lokasi		-

	pengguna menuju tempat wisata yang dipilih ?		
7	Apakah fungsi tambah data pada aplikasi berjalan dengan baik ?		-
8	Apakah aplikasi dapat menunjukkan daftar tempat wisata berdasarkan kategori tokonya?		-
9	Apakah fungsi pencarian tempat wisata pada aplikasi berjalan dengan baik ?		-
Total		9	0

20. Pengujian Antarmuka Sistem

No	Pertanyaan	Penilaian				
		SS	S	N	TS	STS
1	Apakah aplikasi memiliki tampilan yang menarik bagi pengguna?	-		-	-	-
2	Apakah aplikasi mudah digunakan oleh pengguna ?	-		-	-	-
3	Apakah aplikasi menggunakan bahasa yang mudah di mengerti ?	-		-	-	-
4	Apakah informasi yang ditampilkan aplikasi dapat ditangkap dengan mudah dan jelas ?	-		-	-	-
5	Apakah penggunaan komposisi warna pada tampilan aplikasi sudah sesuai ?	-		-	-	-

6	Apakah penggunaan ikon menu pada aplikasi sudah sesuai?	-		-	-	-
Total		0	5	1	0	0

