

**SISTEM INFORMASI GEOGRAFIS UNTUK PENCARIAN
LOKASI LAHAN PERTANIAN BERBASIS *ANDROID***

SKRIPSI

Oleh :

MUHAMMAD NURIL EFENDI

NIM. 08650130



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN)
MAULANA MALIK IBRAHIM MALANG**

2013

**SISTEM INFORMASI GEOGRAFIS UNTUK PENCARIAN
LOKASI LAHAN PERTANIAN BERBASIS *ANDROID***

SKRIPSI

Diajukan Kepada:

Fakultas Sains dan Teknologi

Universitas Islam Negeri (UIN)

Maulana Malik Ibrahim Malang

**Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

Oleh :

MUHAMMAD NURIL EFENDI

NIM. 08650130

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN)
MAULANA MALIK IBRAHIM MALANG**

2013

**SISTEM INFORMASI GEOGRAFIS UNTUK PENCARIAN
LOKASI LAHAN PERTANIAN BERBASIS *ANDROID***

SKRIPSI

Oleh:

Muhammad Nuril Efendi

NIM. 08650021

Telah disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Suhartono, M. Kom
NIP. 19680519 200312 1 001

Dr. Ahmad Barizi, MA
NIP. 19731212 199803 1 001

Tanggal, 26 September 2013

Mengetahui

Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdiان
NIP. 19740424 200901 1 008

**SISTEM INFORMASI GEOGRAFIS UNTUK PENCARIAN LOKASI
LAHAN PERTANIAN BERBASIS *ANDROID***

SKRIPSI

Oleh:

Muhammad Nuril Efendi

NIM. 08650021

Telah Dipertahankan di Depan Dewan Penguji Skripsi dan
Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)

Tanggal, 26 September 2013

Susunan Dewan Penguji	Tanda Tangan
1. Penguji Utama : <u>Roro Inda Melani, M.Sc</u> NIP. 19780925 200501 2 008	()
2. Ketua Penguji : <u>A'la Syauqi, M.Kom</u> NIP. 19771201 200801 1 007	()
3. Sekretaris Penguji : <u>Dr. Suhartono, M. Kom</u> NIP. 19680519 200312 1 001	()
4. Anggota Penguji : <u>Dr. Ahmad Barizi, MA</u> NIP. 19731212 199803 1 001	()

**Mengetahui dan Mengesahkan,
Ketua Jurusan Teknik Informatika**

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

SURAT PERNYATAAN ORISINALITAS PENELITIAN

Saya yang bertanda tangan di bawah ini:

Nama : Muhammad Nuril Efendi
NIM : 08650130
Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika
Judul Penelitian : Sistem Informasi Geografis Untuk Pencarian Lokasi Lahan
Pertanian Berbasis *Android*

Menyatakan dengan sebenar-benarnya bahwa hasil penelitian saya ini tidak terdapat unsur-unsur penjiplakan karya penelitian atau karya ilmiah yang pernah dilakukan atau dibuat oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka. Apabila ternyata hasil penelitian ini terbukti terdapat unsur-unsur penjiplakan, maka saya bersedia untuk mempertanggung jawabkan, serta diproses sesuai peraturan yang berlaku.

Malang, 26 September 2013

Yang Membuat Pernyataan,

Muhammad Nuril Efendi
NIM. 08650130

LEMBAR PERSEMBAHAN

Bismillaahirrahmaanirrahîm...

*Kakak Siti Ni'mah dan adik Isnaini Likah Tercinta
kehadiran kalian menjadi motivasiku
untuk selalu menjadi yang terbaik*

*Istri Tercinta Nuril Nuzulia
yang sedang menempuh Program Pascasarjana S2
Jurusan Pendidikan Guru Madrasah Ibtidaiyah
Terima kasi banyak telah memberiku motivasi dan
membantuku menyelesaikan sekripsi ini.
Kau adalah sumber inspirasiku.
Semoga Alloh senantiasa menjagamu dimanapun kau
berada.*

MOTTO

*”Sesungguhnya sesudah kesulitan itu ada kemudahan.
Maka apabila kamu Telah selesai (dari sesuatu urusan),
kerjakanlah dengan sungguh-sungguh (urusan) yang lain.
Dan Hanya kepada Tuhanmulah hendaknya kamu berharap.”*
(Qs. Alam-Nasyrah/ 94 : 6-8)

KATA PENGANTAR

Assalamu alaikum Wr. Wb.

Puji syukur penulis haturkan kehadiran Ilahi Rabbi yang telah memberi Rahmat, Taufiq dan Hidayah-Nya, sehingga penulisan skripsi dengan judul “Sistem Informasi Geografis Untuk Pencarian Lokasi Lahan Pertanian Berbasis *Android*” ini dapat penulis selesaikan tanpa halangan yang berarti.

Untaian shalawat serta salam semoga selalu mengalir kepada junjungan kita, nabi Muhammad SAW, berkat pengorbanan dan kasih beliau, kita semua bisa merasakan indahnya hidup di bawah naungan agama yang damai, yaitu agama Islam.

Penulisan ini diajukan sebagai salah satu persyaratan dalam menyelesaikan program Strata Satu (S1) Sarjana Komputer Teknik Informatika Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang. Selain itu, penulisan ini juga disusun sebagai bentuk partisipasi penulis dalam mengembangkan hasanah keilmuan dan perwujudan ilmu yang telah didapat selama menjadi mahasiswa.

Penyelesaian skripsi ini tidak terlepas dari peran dan dukungan beberapa pihak terkait yang telah banyak memberikan motivasi dan bantuan. Oleh karena itu, rangkaian ungkapan terima kasih penulis sampaikan yang sedalam-dalamnya kepada :

1. Prof. Dr. H. Mudjia Rahardjo, M.Si selaku Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
2. Dr. drh. Bayyinatul Muchtaromah, M.Si, selaku Dekan Fakultas Saintek Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
3. Dr. Cahyo Crysdiyan selaku Ketua Jurusan Teknologi Informatika Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
4. Dr. Suhartono, M.Kom selaku dosen pembimbing, yang telah membimbing dan mengarahkan penulis dalam menyusun skripsi ini.
5. Dr. Ahmad Barizi, M.A selaku dosen pembimbing agama, yang telah membimbing dan mengarahkan penulis dalam menyusun skripsi ini.
6. Bapak dan Ibu dosen Jurusan Teknologi Informatika Fakultas Saintek, yang telah banyak memberikan ilmu kepada penulis sejak berada di bangku kuliah.
7. Keluarga besar Dinas Pertanian Kota Malang yang telah banyak memberikan pengalaman berharga bagi penulis sebagai bekal menyelesaikan skripsi ini.
8. Bapak (Alm) Muslik dan Ibu Muslikah tersayang, kakak, adikku dan seluruh keluarga besar di Lamongan yang telah banyak memberikan doa, motivasi dan dorongan dalam penyelesaian skripsi ini.
9. Semua sahabat yang telah membantu penulis hingga terselesaikannya skripsi ini, khususnya kepada Aprillia Dewi Kreswanti dan Hari Santoso. Sahabat Teknik Informatika UIN Maliki Malang angkatan 2008 semoga

Allah SWT memberikan balasan yang setimpal atas jasa dan bantuan yang telah diberikan.

10. Dan kepada seluruh pihak yang mendukung penulisan skripsi ini, yang tidak dapat disebutkan satu persatu, penulis ucapkan terima kasih yang sebesar-besarnya.

Penulis menyadari penuh dengan kelemahan yang dimilikinya, sehingga dalam menyelesaikan skripsi ini masih terdapat banyak kesalahan dan kekurangan. Oleh karena itu, penulis mengharapkan akan adanya saran dan kritik dari semua kalangan guna menyempurnakan penulisan ini. Akhirnya, mudah-mudahan penelitian ini dapat bermanfaat bagi kita semua, lebih-lebih kepada penulis. Amiin.

Wassalamualaikum Wr. Wb.

Malang, 26 September 2013

Penulis

Muhammad Nuril Efendi
NIM. 08650130

DAFTAR ISI

HALAMAN JUDUL	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN.....	iv
LEMBAR PERNYATAAN	v
PERSEMBAHAN.....	vi
MOTTO	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	xi
DAFTAR TABEL	xv
DAFTAR GAMBAR.....	xvi
DAFTAR LAMPIRAN	xviii
ABSTRAK	xix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	9
1.3 Batasan Masalah.....	10
1.4 Tujuan Penelitian.....	10
1.5 Manfaat Penelitian.....	11
1.6 Metode Penelitian.....	11
1.7 Sistematika Penelitian	12
BAB II TINJAUAN PUSTAKA.....	16
2.1 Pertanian.....	16
2.2 Sistem Informasi Geografis.....	27
2.3 GPS	34
2.4 Android	35
2.4.1 The Dalvik Virtual Machine (DVM)	35
2.4.2 Android SDK (Software Development Kit)	35

2.4.2.1	Application dan Widgets.....	35
2.4.2.2	Application Frameworks.....	36
2.4.2.3	Libraries	36
2.4.2.4	Andoid Run Time.....	37
2.4.2.5	Linux Kernel	37
2.4.3	Fundamental Aplikasi.....	38
2.4.3.1	Activities	38
2.4.3.2	Service.....	39
2.4.3.3	Broadcast Receiver	39
2.4.3.4	Content Provider	40
2.4.4	Emulator	40
2.4.5	Droiddraw	41
2.4.6	Komponen-komponen Android.....	42
2.4.6.1	User Interface (UI).....	42
2.4.6.2	AndroidManifest.xml.....	43
2.4.6.3	Activity.....	45
2.4.6.4	Siklus Activity.....	46
2.4.6.5	Widget.....	49
2.4.6.6	Kumpulan Widgets	50
2.4.6.7	Message Box	52
2.4.6.8	More View	53
2.5	Algoritma Dijkstra	54
2.6	Eclipse Integrated Development Environment (IDE).....	59
2.7	Google Maps.....	61
2.7.1	Cara Kerja Google Maps	62
2.7.2	Google Maps API.....	62
2.8	MySQL	68
2.8.1	Mengenal SQL (Structured Query Language).....	69
2.8.1.1	DDL (Data Definition Language)	70
2.8.1.2	DML (Data Manipulation Language)	70
2.8.1.3	DCL (Data Control Language).....	71

BAB III METODE PENELITIAN DAN PERANCANGAN SISTEM.....	72
3.1 Tahap- Tahap Pembuatan	72
3.2 Analisa Kebutuhan.....	75
3.2.1 Software.....	76
3.2.2 Hardware	77
3.3 Spesifikasi Aplikasi	78
3.4 Tahap Pengerjaan.....	79
3.4.1 Pengumpulan Data.....	80
3.5 Perancangan Sistem.....	80
3.6 Flow Chart	81
3.6.1 Flow Chart Menu Tampilan Peta.....	81
3.6.2 Flow Chart Menu Penambahan lahan.....	82
3.6.3 Flow Chart Menu Cari Jalur	83
3.7 DFD (Data Flow Diagram).....	86
3.7.1 DFD (Data Flow Diagram) Level 0.....	86
3.7.2 DFD (Data Flow Diagram) Level 1.....	87
3.8 Struktur Basis Data	89
3.9 Desain Interface SIG Pertanian	91
 BAB IV HASIL KEGIATAN PKLI	 92
4.1 Implementasi Sistem.....	92
4.1.1 Implementasi SIG Pertanian Provider	92
4.1.2 Implementasi SIG Pertanian Client	93
4.1.3 Ruang Lingkup Perangkat Keras	93
4.1.4 Ruang Lingkup Perangkat Lunak	94
4.1.4.1 Ruang Lingkup Perangkat Lunak Pertanian <i>Provide</i>	94
4.1.4.2 Ruang Lingkup Perangkat Lunak Pertanian <i>Client</i>	94
4.2 Implementasi Interface SIG Pertanian.....	95
4.2.1 Halaman Splash	95
4.2.2 Halaman Utama	95
4.2.3 Login.....	97

4.2.4 Halaman Kategori Wilayah	99
4.2.5 Halaman Menu	100
4.2.6 Halaman Tampilan Peta.....	101
4.2.7 Halaman Penambahan Lahan	102
4.2.8 Halaman Rute	103
4.2.9 Tentang Program	108
4.2.10 Logout.....	108
4.3 Impelemntasi Algoritma	109
4.3.1 Pengambilan Data.....	109
4.3.2 Ekstraksi JSON.....	113
4.3.3 Penghitungan Algoritma Dan Drawing	115
4.4 Uji Coba Sistem.....	117
4.4.1 Proses uji coba	117
4.3.2 Analisa Hasil Uji Coba.....	117
4.5 SIG Pencarian Lokasi lahan pertanian dalam Pandangan Islam	121
BAB V PENUTUP	125
5.1 Kesimpulan	125
5.2 Saran	126
DAFTAR PUSTAKA	127
LAMPIRAN-LAMPIRAN	129

DAFTAR TABEL

Tabel 2.1 <i>Lifecycle Activity</i>	48
Tabel 3.1 <i>Login</i>	89
Tabel 3.2 Kategori wilayah	89
Tabel 3.3 Tipe Lahan	90
Tabel 3.4 Lokasi Lahan Pertanian	90
Tabel 4.1 Uji Coba Pada Smartphone ber Sistem Operasi Android	117
Tabel 4.2 Hasil Rekapitulasi Kuesioner	119



DAFTAR GAMBAR

Gambar 2.1 Contoh Aplikasi GIS	33
Gambar 2.2 Arsitektur <i>Android</i>	38
Gambar 2.3 Emulator	41
Gambar 2.4 Droiddraw.....	41
Gambar 2.5 <i>Lifecycle Activity</i>	47
Gambar 2.6 Posisi <i>Padding</i> Dari Sebuah <i>Widgets</i>	51
Gambar 2.7 Contoh keterhubungan antar titik dalam algoritma Dijkstra.....	55
Gambar 2.8 Contoh kasus Dijkstra - Langkah 1	56
Gambar 2.9 Contoh kasus Dijkstra - Langkah 2	57
Gambar 2.10 Contoh kasus Dijkstra - Langkah 3	57
Gambar 2.11 Contoh kasus Dijkstra - Langkah 4	58
Gambar 2.12 Contoh kasus Dijkstra - Langkah 5	58
Gambar 2.13 Jendela Kerja Eclipse	60
Gambar 3.1 Desain Sistem Secara Umum	72
Gambar 3.2 Diagram Alir Pembuatan sistem	79
Gambar 3.3 Flow Chart Menu Tampilan Peta	82
Gambar 3.4 Flow Chart Menu Penambahan lahan	83
Gambar 3.5 Flow Chart Menu Cari Lahan Menggunakan Algoritma Dijkstra	84
Gambar 3.6 DFD level 0	86
Gambar 3.7 DFD level 1	88
Gambar 3.8 Desain <i>Interface</i> SIG Pertanian.....	91
Gambar 4.1 Halaman <i>Splash</i>	95
Gambar 4.2 Halaman Utama.....	96
Gambar 4.3 <i>Form Login</i>	98
Gambar 4.4 Kategori Wilayah	99
Gambar 4.5 Menu Tampilan	100
Gambar 4.6 Tampilan Petan.....	101
Gambar 4.7 Tambah Lahan.....	102
Gambar 4.8 Rute Terpendek	103

Gambar 4.9 Tentang Program 108
Gambar 4.10 Logout 109



DAFTAR LAMPIRAN

Lampiran 1. Kuesioner Pengujian Sistem.....	129
---	-----



ABSTRAK

Efendi, Muhammad Nuril. 2013. **Sistem Informasi Geografis Untuk Pencarian Lokasi Lahan Pertanian Berbasis *Android***. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pembimbing : (1) Dr. Suhartono, M.Kom (2) Dr. Ahmad Barizi, M.A

Kata Kunci : *Sistem Informasi Geografis, Pertanian, Android, Dijkstra.*

Sistem Informasi pada umumnya memberikan informasi dalam bentuk *website*, selebaran, atau spanduk yang dipasang di pinggir jalan, sedangkan permintaan dari pasar teknologi sekarang adalah menginginkan sistem informasi yang bekerja pada telepon seluler berbasis *Android*. Pengembangan aplikasi mobile saat ini masih terbatas. Terutama mengenai aplikasi yang dapat mencari informasi lokasi lahan pertanian dan menampilkan jalur terpendek dari tempat lahan pertanian yang dituju, terutama di Kota Malang yang saat ini untuk mencari lokasi lahan pertanian agak susah karena wilayah Kota Malang yang sebagian wilayahnya beralih fungsi menjadi perkantoran, perumahan serta pusat perbelanjaan.

Berdasarkan latar belakang tersebut penelitian ini bertujuan untuk membuat suatu Sistem Informasi Geografis berbasis *Android*. Sistem ini akan menghasilkan Pencarian Lokasi Lahan Pertanian dengan memberikan informasi pada setiap lokasinya dengan menggunakan Algoritma *Dijkstra*. Sehingga user lebih mudah dalam mendapatkan dan memahami informasi pertanian yang disampaikan melalui *Mobile Android*.

Dari hasil penilaian 25 pengguna terhadap aplikasi SIG Pertanian menunjukkan penilaian yang baik terhadap aspek komunikasi visual, aspek perangkat lunak dan aspek desain sistem. Oleh karena itu, aplikasi SIG Pertanian ini layak diimplementasikan dan juga dapat membantu mempermudah pemantauan lahan pertanian milik petani. Sedangkan dari hasil pencarian menggunakan Algoritma *Dijkstra* yang penerapannya pada SIG Pertanian berbasis *Android* dapat diukur melalui kemampuan pencarian dan juga keakuratan terdapat jarak tempuh pada peta dengan membandingkan pada jarak tempuh menggunakan alat ukur manual dapat diperoleh keakuratan Algoritma *Dijkstra* mencapai 98,70 %. Dari hasil ini diperoleh tingkat keakuratan yang tinggi sehingga dapat disimpulkan penerapan Algoritma *Dijkstra* pada SIG Pertanian berbasis *Android* layak untuk dikembangkan.

BAB I

PENDAHULUAN

1.1 Latar Belakang

﴿ وَهُوَ الَّذِي أَنْشَأَ جَنَّاتٍ مَّعْرُوشَاتٍ وَغَيْرِ مَّعْرُوشَاتٍ وَالنَّخْلَ وَالزَّرْعَ مُخْتَلِفًا أُكْلُهُمْ
 وَالزَّيْتُونَ وَالرُّمَانَ مُتَشَابِهًا وَغَيْرَ مُتَشَابِهٍ ۚ كُلُوا مِنْ ثَمَرِهِ إِذَا أَثْمَرَ وَءَاتُوا حَقَّهُ يَوْمَ
 حَصَادِهِ ۚ وَلَا تُسْرِفُوا ۚ إِنَّهُ لَا يُحِبُّ الْمُسْرِفِينَ ۝﴾

Artinya:

“Dan Dialah yang menjadikan kebun-kebon yang berjunjung dan yang tidak berjunjung, pohon korma, tanam-tanaman yang bermacam-macam buahnya, zaitun dan delima yang serupa (bentuk dan warnanya) dan tidak sama (rasanya). Makanlah dari buahnya (yang bermacam-macam itu) bila dia berbuah, dan tunaikanlah haknya di hari memetik hasilnya (dengan disedekahkan kepada fakir miskin); dan janganlah kamu berlebih-lebihan. Sesungguhnya Allah tidak menyukai orang yang berlebih-lebihan”.(QS. Al An’am/ 6: 141)

Kata *Az-zar'* ialah tanaman yang tumbuh ditanam manusia, mencakup segala tumbuhan yang ditanam, khususnya yang menjadi makanan pokok. Seperti gandum dan kedelai. Jenis-jenis tumbuhan ini telah disebutkan secara berturut-turut dari yang paling rendah kedudukannya sebagai makanan biasa dan makanan pokok manusia sampai kepada yang paling tinggi dan umum, karena biji-bijian merupakan tumbuhan yang menjadi bahan pokok, sebagai makanan yang mengenyangkan. (A. Mustafa Al-Maragi, 1992, hal. 35)

Dari Q.S Al-An’am/ 6: 141 dapat dijelaskan bahwa Allah memberikan nikmat dan karuniaNya kepada hambaNya yang berupa berbagai jenis

tanaman yang dapat dimanfaatkan oleh manusia dengan tidak berlebihan. Akan tetapi banyak manusia dan kaum yang membantah tentang keesaan Allah dan beranggapan bahwa semua nikmat yang dia dapat adalah bukan semata-mata berasal dari Allah melainkan hasil dari usahanya sendiri. Sementara itu orang-orang yang beriman menyadari bahwa semua nikmat yang diterimanya berasal dari Allah dan merupakan tanda dari kekuasaan-Nya, sehingga setiap usaha yang dilakukan untuk mendapatkan nikmat tersebut selalu dilandasi atas iman dan dengan berpedoman kepada kitab-Nya serta selalu bersyukur atas hasil yang diperolehnya.

Selanjutnya Allah melarang makan berlebih-lebihan, karena hal itu sangat berbahaya bagi kesehatan dan dapat menimbulkan bermacam-macam penyakit yang mungkin membahayakan jiwa. Allah Yang Maha Pengasih kepada hambaNya tidak menyukai hambaNya yang berlebih-lebihan itu. Dengan demikian dapat dipahami bagaimana kasih sayang Allah kepada hambaNya. Dia melengkapkan segala kebutuhan manusia dengan tanaman dan binatang bahkan menjadikan segala apa yang di langit dan di bumi untuk kepentingan makhlukNya, sebagaimana yang tersebut dalam FirmanNya:

أَلَمْ تَرَوْا أَنَّ اللَّهَ سَخَّرَ لَكُمْ مَّا فِي السَّمَوَاتِ وَمَا فِي الْأَرْضِ وَأَسْبَغَ عَلَيْكُمْ نِعْمَهُ ظَهْرَةَ وَبَاطِنَةً
 وَمِنَ النَّاسِ مَن يُجَادِلُ فِي اللَّهِ بِغَيْرِ عِلْمٍ وَلَا هُدًى وَلَا كِتَابٍ مُّنِيرٍ ﴿٢٠﴾

Artinya:

“Tidakkah kamu perhatikan Sesungguhnya Allah Telah menundukkan untuk (kepentingan)mu apa yang di langit dan apa yang di bumi dan menyempurnakan untukmu nikmat-Nya lahir dan batin. dan di antara manusia ada yang membantah tentang (keesaan) Allah tanpa ilmu pengetahuan atau petunjuk dan tanpa Kitab yang memberi penerangan”. (QS. Luqman/ 31: 20)

Kata (سخر) *sakhhara* berarti menundukkan sesuatu sehingga melakukan apa yang dikehendaki oleh yang menundukkannya. Persis seperti pena yang ditundukkan oleh seorang penulis. Ia akan menulis sesuai kehendak penulisnya. Yang menundukkan alam raya adalah Allah SWT. penundukannya untuk manusia. Allah menundukkan dengan menciptakan hukum-hukum alam, lalu manusia diilhami-Nya pengetahuan sehingga mampu menggunakan hukum-hukum alam itu untuk menjadikan alam dapat melakukan apa yang dikehendaki manusia atas izin Allah SWT. Selanjutnya karena penundukan Allah itu dimaksudkan-Nya untuk kepentingan manusia, maka Allah memberi kewenangan dan kemampuan untuk mengelola alam raya. Dia Yang Maha Kuasa itu memerintahkan manusia untuk mengelola sesuai konsep yang dikehendaki-Nya. Namun dalam saat yang sama, konsep itu merupakan ujian buat manusia. Dia dapat melaksanakannya dan untuk itu dia mendapat ganjaran, atau mengabaikannya dan ini mengakibatkan kesengsaraan paling tidak di akhir nanti. Sekali lagi yang menundukkan adalah Allah, bukan manusia. (M. Quraish Shihab, 2002, hal. 21)

Dari Q.S. Luqman/ 31: 20 dapat disimpulkan bahwa Allah telah menundukkan segala yang ada di langit dan di bumi sebagai bukti dari sifat Rahman-Nya, dan sekaligus sebagai penyempurna nikmat yang diberikan kepada semua manusia, baik yang berupa nikmat lahir maupun batin. Akan tetapi semua nikmat tersebut tidak datang dengan sendirinya melainkan dengan usaha. Salah satu bentuk usaha yang dapat dilakukan adalah melalui proses befikir, mempelajari, dan merenungkan segala bentuk ciptaan-Nya

yang ada dimuka bumi yang berwujud lingkungan alam yang ada disekitar kita. Keutamaan orang-orang yang berilmu dan beriman sekaligus, diungkapkan Allah dalam ayat-ayat seperti berikut ini :

أَمَّنْ هُوَ قَنِيتٌ ءَانَاءَ اللَّيْلِ سَاجِدًا وَقَائِمًا يَحْذَرُ الْآخِرَةَ وَيَرْجُوا رَحْمَةَ رَبِّهِ ۗ قُلْ هَلْ يَسْتَوِي
الَّذِينَ يَعْلَمُونَ وَالَّذِينَ لَا يَعْلَمُونَ ۗ إِنَّمَا يَتَذَكَّرُ أُولُو الْأَلْبَابِ ﴿١٠١﴾

Artinya :

“Apakah kamu Hai orang musyrik yang lebih beruntung) ataukah orang yang beribadat di waktu-waktu malam dengan sujud dan berdiri, sedang ia takut kepada (azab) akhirat dan mengharapkan rahmat Tuhannya? Katakanlah: "Adakah sama orang-orang yang mengetahui dengan orang-orang yang tidak mengetahui?" Sesungguhnya orang yang berakallah yang dapat menerima pelajaran”. (QS. Az Zumar / 39:9)

Pendapat al-Maraghy dalam tafsirnya yang dikutip oleh Abuddin Nata (Abuddin Nata, 2002: 27) menyebutkan : *“Katakanlah hai Rasul kepada kaummu, adakah sama orang-orang yang mengetahui bahwa ia akan mendapatkan pahala karena ketaatan pada Tuhan-nya, dengan orang-orang yang tidak mengetahui hal yang demikian itu”.* Ungkapan pertanyaan ini menunjukkan bahwa yang pertama (orang-orang yang mengetahui) akan dapat mencapai derajat kebaikan; sedangkan yang kedua (orang-orang yang tidak mengetahui) akan mendapat kehinaan.

يَتَأْتِيهَا الَّذِينَ ءَامَنُوا إِذَا قِيلَ لَكُمْ تَفَسَّحُوا فِي الْمَجَالِسِ فَافْسَحُوا يَفْسَحِ اللَّهُ لَكُمْ ۗ وَإِذَا قِيلَ انشُرُوا فَانشُرُوا يَرْفَعِ اللَّهُ الَّذِينَ ءَامَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ ۗ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ ﴿١٠٢﴾

Artinya :

“Hai orang-orang beriman apabila kamu dikatakan kepadamu: "Berlapang-lapanglah dalam majlis", Maka lapangkanlah niscaya Allah akan memberi kelapangan untukmu. dan apabila dikatakan: "Berdirilah kamu", Maka berdirilah, niscaya Allah akan meninggikan orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu pengetahuan beberapa

derajat. dan Allah Maha mengetahui apa yang kamu kerjakan”. (QS. Al Mujadilah / 58 :11).

Maksudnya dari ayat tersebut adalah Allah SWT akan mengangkat derajat (martabat) orang – orang yang melaksanakan perintahNya dan RasulNya dan orang – orang yang berilmu pengetahuan. Selanjutnya ayat ini mendorong kita mengadakan kegiatan dibidang ilmu pengetahuan, dengan cara mengunjungi dan menghadiri majelis ilmu. Konsekuensi dari usaha mencari ilmu, akan diperolehnya derajat yang tinggi dari Allah SWT. Sedangkan dalam Hadis Nabi yang diriwayatkan oleh Ibnu Majah, menjelaskan betapa pentingnya mencari ilmu sehingga menjadi sebuah keharusan.

“Telah menceritakan kepadaku Hisyam ibn Ammar, telah menceritakan kepadaku Hafs ibn Sulaiman, telah menceritakan kepadaku Katsir ibn Syindzir dari Muhammad ibn Sirin dari Anas ibn Malik, ia berkata : Telah bersabda Rasulullah SAW : “Mencari ilmu itu wajib bagi setiap muslim dan orang yang memberikan ilmu (kepercayaan) kepada yang bukan ahlinya bagaikan mengalungi babi hutan dengan intan atau permata atau emas.” (Sunan Ibn Majah no. 220)

Hadits riwayat Ibn Majah tersebut di atas, bahwa belajar adalah suatu keharusan dan juga menekankan bahwa jika suatu ilmu (amanat, wewenang, kepercayaan) diberikan kepada orang yang bukan ahlinya maka bagaikan mengalungi babi hutan dengan intan, permata atau emas. Kalimat perumpamaan ini sebagai simbol bahwa perbuatan tersebut sia-sia (tidak ada manfaatnya) bahkan bisa mengakibatkan kehancuran. Menurut Al Ghazali ditinjau dari segi akal, Ilmu Pengetahuan dan Teknologi adalah sesuatu yang palin utama, karena dengan ilmu manusia akan sampai kepada Allah SWT dan

menjadi dekat denganNya. Dan pada akhirnya akan memperoleh kebahagiaan abadi dan kenikmatan yang kekal. Ilmu menimbulkan kemuliaan dunia dan akhirat . Dalam kaitan ini kita dapat mengatakan, bahwa ajaran Islamlah yang amat peduli terhadap perkembangan ilmu pengetahuan dengan berbagai aspeknya. Pada ayat tersebut terkandung juga motivasi yang amat kuat agar orang giat menuntut ilmu pengetahuan, yaitu dengan memberikan kedudukan yang tinggi dalam pandangan Allah SWT.

Untuk mempermudah proses tersebut kita membutuhkan sarana yang berisikan informasi mengenai tempat-tempat yang memiliki kelebihan dan bermanfaat dari tempat-tempat lainnya, dalam hal ini adalah Pertanian yang semuanya merupakan ciptaan Allah yang ditundukkan untuk manusia supaya menjadikan manusia untuk bersyukur atas karunia yang diberikan-Nya, serta menjaga dan memanfaatkannya.

Sistem Informasi pada umumnya memberikan informasi dalam bentuk *website*, selebaran, atau spanduk yang dipasang di pinggir jalan, sedangkan permintaan dari pasar Teknologi sekarang adalah menginginkan sistem informasi yang bekerja pada telepon seluler berbasis *Android*.

Hal ini juga dilihat peneliti terjadi pada Sistem Informasi Geografis yang hanya menampilkan informasinya pada *website*. Oleh karena itu Peneliti membuat Sistem Informasi Geografis untuk pencarian lokasi lahan pertanian berbasis *Android*. Maka dari itu peneliti mencoba merancang sebuah aplikasi Sistem Informasi Geografis yang berjalan pada telepon seluler khususnya

platform berbasis *Android* untuk menambah pengetahuan teknologi informasi khususnya dalam bidang pertanian.

Sistem Informasi Geografis ini hanya menekankan pada pencarian lokasi lahan pertanian saja dengan menggunakan algoritma *Dijkstra*. Peneliti menggunakan algoritma *Dijkstra* ini dengan tujuan pada saat pencarian lokasi lahan pertanian dapat memberikan kemudahan kepada pengguna yaitu jalur lokasi tujuan yang terdekat.

Algoritma *Dijkstra*, (dinamai menurut penemunya, seorang ilmuwan komputer, *Edsger Dijkstra*), adalah sebuah algoritma rakus (*greedy algorithm*) yang dipakai dalam memecahkan permasalahan jarak terpendek (*shortest path problem*) untuk sebuah graf berarah (*directed graph*) dengan bobot-bobot sisi (*edge weights*) yang bernilai tak-negatif. Misalnya, bila *vertices* dari sebuah graf melambangkan kota-kota dan bobot sisi (*edge weights*) melambangkan jarak antara kota-kota tersebut, maka algoritma *Dijkstra* dapat digunakan untuk menemukan jarak terpendek antara dua kota. (R. Munir, 2010, hal. 10)

Pemilihan telepon seluler berbasis *Android* untuk salah satu pengembangan aplikasi selain bersifat open source dan lebih mudah dalam pengoperasiannya, sifat dari telepon seluler yang fleksibel menjadi salah satu alasannya. Sekarang ini kebutuhan masyarakat akan smartphone sudah semakin meningkat. Hingga saat ini tercatat perkembangan smartphone berbasis *Android* meningkat drastis dengan melihat data dari Telkomsel pada desember 2011 yang sudah mencapai 900 ribu user. Penggunaanya juga bervariasi dari mulai anak-anak sampai orang dewasa. Penggunaan telepon

seluler selain sebagai media komunikasi, juga bisa dijadikan sebagai sarana informasi. Beragam aplikasi telah banyak berjalan dalam telepon seluler khususnya platform *Android*. (Muhammad Zaenal Arifin, 2012)

Kota Malang merupakan salah satu daerah otonom dan merupakan kota besar kedua di Jawa Timur setelah Kota Surabaya. Sebagai kota besar, Malang tidak lepas dari permasalahan sosial dan lingkungan yang semakin buruk kualitasnya. Kota yang pernah dianggap mempunyai tata kota yang terbaik di antara kota-kota Hindia Belanda ini, kini banyak dikeluhkan warganya seperti kemacetan dan kesemrawutan lalu lintas, suhu udara yang mulai panas, sampah yang berserakan atau harus merelokasi pedagang kaki lima yang memenuhi alun-alun kota. Namun terlepas dari berbagai permasalahan tata kotanya, pariwisata Kota Malang mampu menarik perhatian tersendiri. Dari segi geografis, Malang diuntungkan oleh keindahan alam daerah sekitarnya membuat para pelancong menjadikan kota ini sebagai tempat singgah dan sekaligus tempat belanja. Perdagangan ini mampu mengubah konsep pariwisata Kota Malang dari kota peristirahatan menjadi kota wisata belanja. (Ditjen Cipta Karya, 2010)

Namun karena semakin berkembangnya Kota Malang membuat daerah yang sebelumnya banyak ditumbuhi tanaman-tanaman di daerah sekitar Kota Malang, banyak lahan persawah yang kemudian sekarang berubah menjadi daerah tempat tinggal atau lokasi perumahan yang membuat semakin sempit masyarakat sekita untuk membudidayakan tanaman-tanaman pada tahun-tahun sekarang ini.

Lahan pertanian sebagai salah satu kawasan resapan air di Kota Malang saat ini tersisa 1.300 hektare dari luas Kota Malang 110,6 kilometer persegi. Luas lahan pertanian ini menyusut 250 hektare selama empat tahun terakhir dari luas semula yang mencapai 1.550 hektare. Hal ini diakibatkan lahan pertanian semakin berkurang karena berubah fungsi menjadi kawasan perkantoran, perumahan serta pusat perbelanjaan. Dinas Pertanian telah mengusulkan ke Badan Perencanaan dan Pembangunan Daerah menetapkan lahan pertanian menjadi lahan abadi atau sabuk hijau. Saat ini, lahan pertanian di Kota Malang hanya berada di empat kecamatan yakni Kecamatan Lowokwaru, Blimbing, Kedungkandang dan Sukun. Dan hampir luas lahan pertanian di Kecamatan Klojen yang menjadi pusat kota habis tak tersisa. (www.greenersmagz.com)

Berangkat dari uraian di atas maka peneliti tertarik untuk mengkaji lebih dalam tentang Sistem Informasi Geografis berbasis *Android* dalam Pencarian Lokasi Lahan Pertanian. Dari sini peneliti mengadakan penelitian dengan mengambil tema yang berjudul “**Sistem Informasi Geografis Untuk Pencarian Lokasi Lahan Pertanian Berbasis *Android***”.

1.2 Rumusan Masalah

Dalam melaksanakan perencanaan dan pembuatan sistem yang akan dibuat pada proyek akhir ini, permasalahan yang ada adalah sebagai berikut:

- a. Bagaimana desain Sistem Informasi Geografis pada *Android* ?

- b. Bagaimana cara menampilkan lokasi Lahan Pertanian berbasis *Android* dengan algoritma *Dijkstra* ?

1.3 Batasan Masalah

Agar ruang lingkup proyek akhir ini tidak menyimpang dari tujuan proyek akhir, maka dibutuhkan beberapa batasan masalah antara lain:

- a. Lokasi pencarian lahan pertanian yang digunakan adalah di Kota Malang.
- b. Data yang digunakan pada proyek akhir ini adalah data lokasi lahan Pertanian berdasarkan ketersediaan data dari Dinas Pertanian.
- c. Perangkat yang digunakan adalah *Smartphone Android*.
- d. Metode yang digunakan adalah teknik pencarian dengan menggunakan algoritma *Dijkstra*.
- e. Rute ditampilkan pada peta menggunakan jalur lintasan kendaraan mobil.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk membuat suatu Sistem Informasi Geografis berbasis *Android*. Sistem ini akan menghasilkan Pencarian Lokasi Lahan Pertanian dengan memberikan informasi pada setiap lokasinya dengan menggunakan Algoritma *Dijkstra*. Sehingga user lebih mudah dalam mendapatkan dan memahami informasi pertanian yang disampaikan melalui *Smartphone Android*.

1.5 Manfaat Penelitian

Kegunaan yang dapat dihasilkan dari hasil penelitian dalam tugas akhir ini adalah :

1. Program aplikasi yang dibuat juga akan sangat bermanfaat bagi petani di Kota Malang bahkan seluruh Indonesia dan dapat dijadikan bahan untuk penelitian lebih lanjut di bidang yang berkaitan.
2. Hasil penelitian ini diharapkan dapat digunakan sebagai langkah awal untuk membangun Sistem Informasi Geografis berbasis *Android*, yang bisa diaplikasikan pada lahan pertanian.
3. Dari hasil penelitian ini juga diharapkan dapat diperoleh pemahaman yang lebih baik terhadap Sistem Informasi Geografis berbasis *Android* dengan menggunakan Algoritma *Dijkstra*.

1.6 Metode Penelitian

Tahapan-tahapan yang dilakukan selama penelitian dan pembangunan Sistem Informasi Geografis Untuk Pencarian Lokasi Lahan Pertanian Berbasis *Android* adalah sebagai berikut:

1. Studi Literatur

Pada tahap ini dilakukan pendalaman buku-buku literatur yang berhubungan dengan pembuatan aplikasi berbasis *Android* dan SIG.

2. Pengumpulan Data

Kegiatan mengumpulkan data-data lokasi seluruh Pertanian di Kota Malang dengan menggunakan *Google Maps* yang akan digunakan dalam pembuatan aplikasi.

3. Perancangan Perangkat Lunak

Proses pengumpulan data akan dibuat menjadi perangkat lunak SIG berbasis *Android*.

4. Integrasi Keseluruhan Perangkat Lunak

Dari perangkat lunak yang telah dibuat kemudian diintegrasikan dengan data yang ada. Lalu dijalankan pada sebuah telepon seluler berbasis *Android*.

5. Pengujian dan Analisa Perangkat Lunak

Pengujian dan analisa didasarkan pada cara kerja aplikasi serta dari sudut pandang pengguna.

6. Pembuatan Laporan Proyek Akhir

Tahap ini merupakan tahap terakhir yaitu melakukan penulisan laporan dan detail tentang proyek akhir yang dibuat

1.7 Sistematika Penulisan

BAB I: PENDAHULUAN

Bab ini mengantarkan pembaca untuk dapat menjawab pertanyaan apa yang diteliti, untuk apa dan mengapa penelitian ini dilakukan yang termuat dalam sub bab-sub bab berikut:

1.1 Latar Belakang

Alasan yang melatari dilakukannya penelitian ini yakni Sistem Informasi Geografis Untuk Pencarian Lokasi Lahan Pertanian Berbasis *Android*. dikemukakan dalam sub bab ini.

1.2 Rumusan Masalah

Sub bab ini berisi fokus permasalahan yang dibahas dalam penelitian ini, yaitu bagaimana desain Sistem Informasi Geografis pada *Android* dan bagaimana cara menampilkan lokasi Lahan Pertanian berbasis *Android* dengan algoritma *Dijkstra*.

1.3 Batasan Masalah

Penelitian ini dibatasi dalam ruang lingkup tertentu yang dijelaskan dalam sub bab ini.

1.4 Tujuan Penelitian

Sub bab ini menjelaskan tujuan atau hasil akhir dari penelitian ini yaitu implementasi teknologi untuk membuat suatu Sistem Informasi Geografis berbasis *Android*.

1.5 Manfaat Penelitian

Manfaat dan harapan dari hasil penelitian ini berupa sistem yang berguna dalam membangun Sistem Informasi Geografis Berbasis *Android*, yang bisa diaplikasikan pada lahan Pertanian yang disampaikan dalam sub bab ini.

1.6 Metode Penelitian

Runtutan proses yang diterapkan dalam penelitian ini dijelaskan dalam sub bab ini.

1.7 Sistematika Penulisan

Sub bab ini menjelaskan orientasi atau maksud dari masing-masing bab atau sub bab.

BAB II: TINJAUAN PUSTAKA

Bab ini menjelaskan konsep dan teori dasar yang mendukung penulisan tugas akhir ini seperti Sistem Informasi Geografis, GPS (*Global Positioning System*), *Android*, Algoritma *Dijkstra*, *Eclipse Integrated Development Environment (IDE)*, *Google Maps*, *MySql*.

BAB III: METODE PENELITIAN

Bab ini menjelaskan mengenai analisis dan perancangan Aplikasi Sistem Informasi Geografis Untuk Pencarian Lokasi Lahan Pertanian Berbasis *Android* secara keseluruhan.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi pengujian terhadap hasil pengujian dari aplikasi yang telah dibangun.

BAB V PENUTUP

Bab ini berisi kesimpulan dan saran terhadap seluruh kegiatan tugas akhir yang telah dilakukan.

DAFTAR PUSTAKA

Seluruh bahan rujukan atau referensi dalam penulisan skripsi ini, dicantumkan dalam bab ini.

LAMPIRAN

Data atau keterangan lain yang berfungsi untuk melengkapi uraian yang telah disajikan dalam bagian utama di tempatkan di bagian ini.



BAB II

TINJAUAN PUSTAKA

2.1 Pertanian

﴿ وَهُوَ الَّذِي أَنْشَأَ جَنَّاتٍ مَّعْرُوشَاتٍ وَغَيْرِ مَّعْرُوشَاتٍ وَالنَّخْلَ وَالزَّرْعَ مُخْتَلِفًا أُكْلُهُمْ وَالزَّيْتُونَ وَالرُّمَانَ مُتَشَابِهًا وَغَيْرَ مُتَشَابِهٍ كُلُوا مِنْ ثَمَرِهِ إِذَا أَثْمَرَ وَآتُوا حَقَّهُ يَوْمَ حَصَادِهِ وَلَا تُسْرِفُوا إِنَّهُ لَا يُحِبُّ الْمُسْرِفِينَ ﴾

Artinya:

“Dan Dialah yang menjadikan kebun-kebon yang berjunjung dan yang tidak berjunjung, pohon korma, tanam-tanaman yang bermacam-macam buahnya, zaitun dan delima yang serupa (bentuk dan warnanya) dan tidak sama (rasanya). Makanlah dari buahnya (yang bermacam-macam itu) bila dia berbuah, dan tunaikanlah haknya di hari memetik hasilnya (dengan disedekahkan kepada fakir miskin); dan janganlah kamu berlebih-lebihan. Sesungguhnya Allah tidak menyukai orang yang berlebih-lebihan”. (QS. Al An’am/ 6: 141)

Kata *Az-zar'* ialah tanaman yang tumbuh ditanam manusia, mencakup segala tumbuhan yang ditanam, khususnya yang menjadi makanan pokok. Seperti gandum dan kedelai. Jenis-jenis tumbuhan ini telah disebutkan secara berturut-turut dari yang paling rendah kedudukannya sebagai makanan biasa dan makanan pokok manusia sampai kepada yang paling tinggi dan umum, karena biji-bijian merupakan tumbuhan yang menjadi bahan pokok, sebagai makanan yang mengenyangkan. (A. Mustafa Al-Maragi, 1992, hal. 35).

Allah SWT menciptakan manusia dan mengistimewakannya dari segenap makhluk-Nya yang lain dengan nikmat akal yang dengannya dia dapat

mengatur, meneliti dan berpikir tentang alam semesta yang ada di sekelilingnya, yang tak pernah mengenal akhir dan tak pernah diketahui permulaannya. Manusia dapat berpikir tentang benda yang ada di sekitarnya yang diciptakan oleh sang pencipta pertama, berupaya memanfaatkannya dan mempelajarinya lalu mendapatkan makanan, obat-obatan, pakaian, minuman dan tempat tinggal. Manusia mampu mengkaji hakikat dan dalil-dalilnya yang menunjukkan adanya Sang Pencipta alam semesta, Sang Pemberi kehidupan dan berbagai kenikmatan ini, kekuasaan-Nya dan ilmu-Nya.

وَالَّتَيْنِ وَالزَّيْتُونَ ﴿١﴾ وَطُورِ سَيْنِينَ ﴿٢﴾ وَهَذَا الْبَلَدِ الْأَمِينِ ﴿٣﴾ لَقَدْ خَلَقْنَا الْإِنْسَانَ فِي أَحْسَنِ تَقْوِيمٍ ﴿٤﴾ ثُمَّ رَدَدْنَاهُ أَسْفَلَ سَافِلِينَ ﴿٥﴾ إِلَّا الَّذِينَ ءَامَنُوا وَعَمِلُوا الصَّالِحَاتِ فَلَهُمْ أَجْرٌ غَيْرُ مَمْنُونٍ ﴿٦﴾ فَمَا يُكَذِّبُكَ بَعْدُ بِالدِّينِ ﴿٧﴾ أَلَيْسَ اللَّهُ بِأَحْكَمَ الْحَاكِمِينَ ﴿٨﴾

Artinya:

“Demi (buah) Tin dan (buah) Zaitun, dan demi bukit Sinai, dan demi kota (Mekah) ini yang aman, Sesungguhnya Kami telah menciptakan manusia dalam bentuk yang sebaik-baiknya . kemudian Kami kembalikan Dia ke tempat yang serendah-rendahnya (neraka), kecuali orang-orang yang beriman dan mengerjakan amal saleh; Maka bagi mereka pahala yang tiada putus-putusnya. Maka Apakah yang menyebabkan kamu mendustakan (hari) pembalasan sesudah (adanya keterangan-keterangan) itu? Bukankah Allah hakim yang seadil-adilnya”.(QS. At Tiin/ 95: 1-8).

Kata tin dalam al-qur’an hanya disebut satu kali, yaitu dalam surat ini, ada ahli tafsir yang menyebutkan bahwa tin adalah jenis buah yang terdapat di Timur Tengah. Kata zaitun disebut empat kali dalam Al-qur’an. Tidak semua ahli tafsir sependapat bahwa yang di maksud Tin dan Zaitun adalah nama buah. Ada yang berpendapat bahwa Tin adalah nama bukit tempat Nabi Ibrahim a.s menerima wahyu, sedangkan Zaitun adalah nama bukit di daerah

Yerusalem tempat nabi Isa menerima wahyu. Jadi, Tin dan Zaitun adalah dua tempat yang di anggap bersejarah.

Tin dan Zaitun ialah dua tempat, yang di masing-masing tempat itu Allah telah membangkitkan Nabi-nabi utusan-Nya, Rasul-rasul yang terkemuka, mempunyai syariat yang besar-besar. Pertama tempat yang di sana banyak tumbuh Tin dan Zaitun. Itulah Baitul-Maqdis. Di sanalah Tuhan mengutus Isa bin Maryam ,alaihi-salam.

Menurut Zainal Abidin Bagir, 2006, hal. 3 “Maka firman Tuhan demi buah tin, demi buah zaitun, demi Bukit Thurisinina, demi negeri yang aman ini,” hal tersebut adalah sumpah kemuliaan yang dianugerahkan Tuhan kepada kedua tempat yang mulia lagi agung, yang di sana sinar Allah dan petunjuk-Nya dan kedua tempat itu diturunkan ketiga kitab-Nya: Taurat, Injil dan Al-Qur’an, sebagaimana yang telah disebutkan dalam Taurat: “Datang Allah dari Torsina, telah terbit di Seir dan gemerlapan cahayanya dari gunung Paran.” Sekedar itu kita salinkan dari Ibnu Taimiyah.

Para ahli tafsir berbeda pendapat tentang sumpah Allah SWT dengan Tin dan Zaitun. Sebagian dari mereka berpendapat bahwa sumpah itu berkenaan dengan keduanya secara dzatiah yakni Tin dan Zaitun, karena banyaknya kegunaan masing-masing keduanya disebut disebabkan oleh adanya peristiwa agung yang sangat berpengaruh pada kehidupan manusia (Nabi Adam), bernaung di bawah pohon Tin di surga. Adapun ketika tampak aurat istrinya, keduanya segera menutupi aurat masing-masing dengan daun pohon Tin.

Hampir seluruh ahli tafsir sependapat kalau yang dimaksud Thur Sinin pada ayat ke 2 dan 3 adalah bukit Tursina atau yang lebih dikenal dengan nama bukit Sinai, yaitu bukit yang berada di Palestina, tempat Nabi Musa menerima wahyu, sementara yang dimaksud Baladil Amiin adalah kota mekkah, tempat Nabi Muhammad menerima wahyu. Dengan empat ayat diatas, Allah bersumpah dengan empat tempat penting, yaitu Tin, Zaitun, Tursina (bukit sanai) dan kota Mekkah (Baladil Amiin), dimana tempat tersebut Nabi Ibrahim a.s, Musa a.s, Isa a.s dan Nabi Muhammad saw menerima wahyu untuk memberikan bimbingan dan pencerahan hidup pada umat manusia untuk menjaga manusia agar tetap berada dalam kemuliaannya

Tumbuhan merupakan salah satu makhluk hidup ciptaan Allah yang memiliki banyak sekali manfaat. Tumbuh-tumbuhan dapat memunculkan beberapa zat untuk dimanfaatkan oleh makhluk hidup lainnya, misalnya mulai beberapa vitamin-vitamin, minyak dan masih banyak lainnya. Dalam firman-Nya Allah menjelaskan:

وَهُوَ الَّذِي أَنْزَلَ مِنَ السَّمَاءِ مَاءً فَأَخْرَجْنَا بِهِ نَبَاتَ كُلِّ شَيْءٍ فَأَخْرَجْنَا مِنْهُ خَضِرًا مَخْرُجًا مِنْهُ
حَبًّا مُتَرَاكِبًا وَمِنَ النَّخْلِ مِن طَلْعِهَا قِنْوَانٌ دَانِيَةٌ وَجَنَّاتٍ مِّنْ أَعْنَابٍ وَالزَّيْتُونَ وَالرُّمَّانَ
مُشْتَبِهًا وَغَيْرَ مُتَشَبِهٍ ۗ انظُرُوا إِلَى ثَمَرِهِ إِذَا أَثْمَرَ وَيَنْعِهِ ۗ إِنَّ فِي ذَٰلِكُمْ لَآيَاتٍ لِّقَوْمٍ يُؤْمِنُونَ



Artinya :

“Dan Dialah yang menurunkan air hujan dari langit, lalu Kami tumbuhkan dengan air itu segala macam tumbuh-tumbuhan Maka Kami keluarkan dari tumbuh-tumbuhan itu tanaman yang menghijau. Kami keluarkan dari tanaman yang menghijau itu butir yang banyak; dan dari

mayang korma mengurai tangkai-tangkai yang menjulai, dan kebun-kebun anggur, dan (kami keluarkan pula) zaitun dan delima yang serupa dan yang tidak serupa. perhatikanlah buahnya di waktu pohonnya berbuah dan (perhatikan pulalah) kematangannya. Sesungguhnya pada yang demikian itu ada tanda-tanda (kekuasaan Allah) bagi orang-orang yang beriman”.(QS. Al An'am / 6:99).

Tumbuhan mengalami proses pertumbuhan yang sangat rumit. Mulai dari berkecambah dengan melakukan penyerapan air dari dalam tanah tumbuhan pun memulai perkembangannya. Biji yang tadinya tumbuh menjadi kecambah kulitnya pun mulai robek karena perkembangannya. Selanjutnya tumbuhan mulai mengeluarkan akar dan menembus kedalam tanah untuk mencari makanan dan masih panjang lagi perjalanan tumbuhan menjalani proses pertumbuhannya. (Imaduddin Abdulrahim.1997, hal. 96).

Semua proses pertumbuhan. Mulai dari permukaan yang mendapatkan siraman air, pergerakan, perkembangan dan pertumbuhan yang dialami oleh tanaman mulai sejak awal sampai dengan proses selanjutnya sebenarnya telah terangkum dalam kata didalam Al-Quran, seperti dalam kalimat *ihazzat* yang berarti “bergerak”, *wa robot* yang memiliki arti “bertambah atau berkembang”, serta *wa anbatat* yang artinya “menumbuhkan”. Kata-kata yang telah disebutkan dalam al-quran ini sangatlah sesuai dengan apa yang telah dikemukakan dalam penelitian-penelitian ilmu pengetahuan modern.

هُوَ الَّذِي أَنْزَلَ مِنَ السَّمَاءِ مَاءً لَكُمْ مِنْهُ شَرَابٌ وَمِنْهُ شَجْرٌ فِيهِ تُسِيمُونَ ﴿٦٩﴾ يُنْبِتُ لَكُمْ بِهِ الزَّرْعَ وَالزَّيْتُونَ وَالنَّخِيلَ وَالْأَعْنَابَ وَمِنْ كُلِّ الثَّمَرَاتِ ۗ إِنَّ فِي ذَلِكَ لَآيَةً لِّقَوْمٍ يَتَفَكَّرُونَ ﴿٧٠﴾

Artinya :

“Dia-lah, yang telah menurunkan air hujan dari langit untuk kamu, sebahagiannya menjadi minuman dan sebahagiannya (menyuburkan) tumbuh-tumbuhan, yang pada (tempat tumbuhnya) kamu menggembalakan ternakmu. Dia menumbuhkan bagi kamu dengan air hujan itu tanam-tanaman; zaitun, korma, anggur dan segala macam buah-buahan. Sesungguhnya pada yang demikian itu benar-benar ada tanda (kekuasaan Allah) bagi kaum yang memikirkan”.(QS. An Nahl / 16: 10-11).

Di setiap tempat kita dapat menemui berbagai jenis tumbuhan. Entah itu di taman, ladang, pedesaan, perkotaan atau dimanapun itu. Sebagian tumbuh-tumbuhan dapat hidup dimanapun tempatnya. Akan tetapi ada juga beberapa jenis tumbuhan yang hanya dapat tumbuh ditempat tertentu saja. Ada tumbuhan yang hanya bisa tumbuh di daerah tropis, ada pula yang hanya cocok tumbuh didaerah subtropis.

Tumbuhan memiliki banyak spesies serta jenis yang beragam. Dan sama pula dengan makhluk hidup lainnya. Di seluruh penjuru dunia ini terdapat banyak sekali jenis tumbuh-tumbuhan, mulai dari yang terkecil sampai yang terbesar. Dalam sebuah penelitian telah terdapat 350.000 tumbuh-tumbuhan yang telah terdaftar dari seluruh permukaan bumi. Menurut Abduh, diperlukan pengamatan terhadap jenis tumbuh-tumbuhan dan binatang yang memiliki kekuatan memenuhi kebutuhan-kebutuhan, untuk memelihara wujud hidupnya dengan mempergunakan alat-alat dan anggota-anggotanya yang terletak di badannya.(Jamaluddin Mahran, 2005, hal. 31).

Tumbuhan di bumi ini diciptakan oleh Allah berpasangan, ada yang jantan dan ada pula yang betina.

سُبْحٰنَ الَّذِيْ خَلَقَ الْاَزْوَاجَ كُلَّهَا مِمَّا تُنْبِتُ الْاَرْضُ وَمِنْ اَنْفُسِهِمْ وَمِمَّا لَا يَعْلَمُوْنَ ﴿٦٠﴾

Artinya :

“Maha suci Tuhan yang telah menciptakan pasangan-pasangan semuanya, baik dari apa yang ditumbuhkan oleh bumi dan dari diri mereka maupun dari apa yang tidak mereka ketahui”.(QS. Yaasiin / 36: 36).

Proses kehidupan tumbuhan mulai dari pertumbuhan awal sampai menghasilkan buah tersusun dari berbagai sel-sel. Mulai dari sel untuk menyimpan makanan yang telah diserap, sel pertumbuhan serta sel-sel lainnya.

Semua sel pada tumbuhan dibatasi oleh dinding-dinding sel yang terbuat dari selulosa. Selulosa yang masih muda dinding selnya sangatlah tipis sedangkan semakin tua selulosanya maka sel dinding sel semakin tebal. Itulah penyebab mengapa tumbuhan yang masih muda memiliki sifat yang lunak, lain halnya dengan tumbuhan yang tumbuh semakin tua maka semakin keras pula tumbuhannya itu. (Muhammad Utsman Najati, 2005, hal. 50).

Dalam setiap sel tumbuhan hijau daun mengandung klorofil untuk menyerap energi matahari. Klorofil menyerap energi matahari dan digunakan sebagai makanan. Energi yang telah terserap oleh klorofil akan tersimpan dalam tumbuhan tersebut. Yang sangat menabjubkan bahwa tumbuhan hijau dapat menyimpan energi hingga jutaan tahun dalam bentuk fosil.

Banyak tumbuh-tumbuhan yang oleh Allah disebutkan dalam Al-quran dan tumbuhan tersebut memiliki banyak manfaat dan khasiat. Misalnya tumbuhan kurma, jahe, pohon tin dan masih banyak yang lainnya.

Pertanian adalah proses menghasilkan bahan pangan, ternak, serta produk-produk agro industri dengan cara memanfaatkan sumber daya

tumbuhan dan hewan. Terkait dengan pertanian, usaha tani (*farming*) adalah sekumpulan kegiatan yang dilakukan dalam budi daya (tumbuhan maupun hewan). Petani adalah sebutan bagi mereka yang menyelenggarakan usaha tani, sebagai contoh "petani tembakau" atau "petani ikan". Khusus untuk pembudidaya hewan ternak (*livestock*) disebut sebagai peternak. Ilmuwan serta pihak-pihak lain yang terlibat dalam perbaikan metode pertanian dan aplikasinya juga dianggap terlibat dalam pertanian. (Akhmad Suhadak, 2010, hal. 27)

Bagian terbesar penduduk dunia bermata pencaharian dalam bidang-bidang di lingkup pertanian, namun pertanian hanya menyumbang 4% dari PDB (Produk Domestik Bruto) dunia. Sejarah Indonesia sejak masa kolonial sampai sekarang tidak dapat dipisahkan dari sektor pertanian dan perkebunan, karena sektor - sektor ini memiliki arti yang sangat penting dalam menentukan pembentukan berbagai realitas ekonomi dan sosial masyarakat di berbagai wilayah Indonesia. Berdasarkan data BPS (Badan Pusat Statistik) tahun 2002, bidang pertanian di Indonesia menyediakan lapangan kerja bagi sekitar 44,3% penduduk meskipun hanya menyumbang sekitar 17,3% dari total pendapatan domestik bruto.

Kelompok ilmu-ilmu pertanian mengkaji pertanian dengan dukungan ilmu-ilmu pendukungnya. Inti dari ilmu-ilmu pertanian adalah biologi dan ekonomi. Karena pertanian selalu terikat dengan ruang dan waktu, ilmu-ilmu pendukung, seperti ilmu tanah, meteorologi, permesinan pertanian, biokimia, dan statistika, juga dipelajari dalam pertanian. Usaha tani (*farming*) adalah

bagian inti dari pertanian karena menyangkut sekumpulan kegiatan yang dilakukan dalam budidaya.

Usaha pertanian memiliki dua ciri penting:

- 1) Selalu melibatkan barang dalam volume besar.
- 2) Proses produksi memiliki resiko yang relatif tinggi.

Dua ciri khas ini muncul karena pertanian melibatkan makhluk hidup dalam satu atau beberapa tahapnya dan memerlukan ruang untuk kegiatan itu serta jangka waktu tertentu dalam proses produksi. Beberapa bentuk pertanian modern (misalnya budidaya alga, *hidroponika*) telah dapat mengurangi ciri-ciri ini tetapi sebagian besar usaha pertanian dunia masih tetap demikian. (Akhmad Suhadak, 2010, hal. 27)

Di dalam kitab Shohih Muslim dibawakan hadis yang diriwayatkan dari sahabat Anas rodhiyallohu ‘anhu dia berkata: Bahwasanya ketika sampai di Madinah Nabi shallallohu ‘alaihi wa sallam melewati suatu kaum (dari kalangan sahabat anshor) yang sedang mengawinkan pohon kurma, maka beliau berkata: *”Sekiranya kalian tidak melakukannya niscaya itu lebih baik.”* Anas melanjutkan: *”kemudian (mereka tidak melakukannya) sehingga hasilnya jelek (gagal).”* Tatkala Nabi shallallohu ‘alaihi wa sallam kembali melewati mereka, beliau bertanya kepada mereka: *”Bagaimana dengan pohon-pohon kurma kalian?”* Mereka berkat: *”Bukankah anda yang mengatakan begini dan begitu (mereka mengikuti perkataan Nabi shallallohu ‘alaihi wa sallam tersebut meskipun hasilnya jelek).* Maka Nabi shallallohu ‘alaihi wa sallam bersabda: *”Kalian lebih tahu dengan urusan dunia kalian”.*

Dari uraian di atas dapat diketahui bahwa Islam menyerahkan pengembangan ilmu dan teknologi pertanian kepada umat manusia. Karena Ilmu dan Teknologi Pertanian adalah urusan dunia. Merupakan suatu hal yang bijak dan tepat apabila suatu perkara diserahkan kepada ahlinya. Maka pada masalah-masalah ilmu dan teknologi pertanian diserahkan kepada ahlinya berupa ilmuwan, peneliti dan orang yang berkompeten di bidang tersebut. (Abuddin Nata, 2002, hal. 27-28).

Dalam Islam, jika pertanian merupakan satu-satunya bidang yang seseorang boleh lakukan untuk mencari nafkah bagi menyara diri dan keluarganya, maka hukum bertani itu adalah fardu ‘ain baginya.

Menurut Dr. Zainal Azam Abd. Rahman seorang cendekiawan Islam dalam tulisan beliau dalam akhbar Berita Harian bertarikh 6 Januari 2005, kegiatan pertanian menjadi fardu kifayah kerana manfaatnya lebih besar daripada manfaat pribadi. Kebanyakan fuqaha' Islam berpendapat bahawa pertanian adalah lebih afdal atau utama pada pandangan Islam dan suatu gagasan berbanding lain-lain jenis perniagaan dan perancangan projek-projek “Mega-Mega” kerana manfaat pertanian lebih meluas dan menjangkau kehidupan rayat justeru kepentingannya tidak dapat dinafikan sebagai bidang yang membekalkan makanan kepada umat. Sebagaimana firman Allah dalam surah Abasa ayat 27 – 32 yang bermaksud :

فَأَنْبَتْنَا فِيهَا حَبًّا ۖ ﴿٢٧﴾ وَعِنَبًا وَقَضْبًا ۖ ﴿٢٨﴾ وَزَيْتُونًا وَنَخْلًا ۖ ﴿٢٩﴾ وَحَدَائِقَ غُلْبًا ۖ ﴿٣٠﴾ وَفَيْكِهَةً ۖ ﴿٣١﴾ وَأَبًّا ۖ ﴿٣٢﴾ مَتَاعًا لَّكُمْ وَلِأَنْعَمِكُمْ ۗ ﴿٣٣﴾

Artinya :

“Lalu Kami tumbuhkan biji-bijian di bumi itu, anggur dan sayur-sayuran, zaitun dan kurma, kebun-kebun (yang) lebat, dan buah-buahan serta rumput-rumputan, untuk kesenanganmu dan untuk binatang-binatang ternakmu”.(QS. Abasa/ 80:27-32).

Tentulah menjadi masalah yang besar sekiranya sebuah negara itu banyak bergantung kepada sumber luar untuk mendapatkan bahan makanan. Ini kerana berlaku di negara peperangan dan negara berpeluang bencana alam, menyebabkan stok makanan sukar didapati. Pandangan itu tepat, jika ditinjau dari keadaan yang berlaku pada hari ini, disetengah negara dimana sekatan-sekatan ekonomi, peparangan dan kemusnahan disebabkan oleh bencana alam menyebabkan kekurangan makanan dan kesusahan dimana-mana. Nabi Muhammad saw bersabda:

"Andainya kiamat tiba dan pada tangan seseorang dari pada kamu ada sebatang anak kurma, maka hendaklah dia tanpa berlengah-lengah lagi menanamkannya." (Hadis riwayat Imam Ahmad).

Demikianlah pentingnya kegiatan pertanian hingga pada zaman akhir, bidang ini tidak boleh diabaikan kerana ia adalah sumber terpenting bagi kehidupan manusia sebagai penyumbang bekal makanan. (Pelita Irsyad, 2002, hal. 10)

Allah *Subhana Wa Ta'ala* menjanjikan insentif istimewa kepada pengusaha sektor pertanian sesuai dengan kedudukannya sebagai sektor yang sangat digalakkan, kita dapati ada Allah Swt telah menjanjikan insentif yang lumayan bagi petani dan pengusaha sektor ini, sama ada ia dilihat dari sudut

kebendaan atau kerohanian. Bagi umat Islam, bidang pertanian adalah antara cara mudah bagi mendapat pahala dan ganjaran daripada Allah, selain menerima manfaat atau pendapatan halal. (Zainal Azam, 2012, hal. 23)

Semua usaha pertanian pada dasarnya adalah kegiatan ekonomi sehingga memerlukan dasar-dasar pengetahuan yang sama akan pengelolaan tempat usaha, pemilihan benih/bibit, metode budidaya, pengumpulan hasil, distribusi produk, pengolahan dan pengemasan produk, dan pemasaran. Apabila seorang petani memandang semua aspek ini dengan pertimbangan efisiensi untuk mencapai keuntungan maksimal maka ia melakukan pertanian intensif (*intensive farming*). Usaha pertanian yang dipandang dengan cara ini dikenal sebagai agribisnis. Program dan kebijakan yang mengarahkan usaha pertanian ke cara pandang demikian dikenal sebagai intensifikasi. Karena pertanian industrial selalu menerapkan pertanian intensif, keduanya sering kali disamakan.

2.2 Sistem Informasi Geografis

هُوَ الَّذِي جَعَلَ الشَّمْسَ ضِيَاءً وَالْقَمَرَ نُورًا وَقَدَرَهُ مَنَازِلَ لِتَعْلَمُوا عَدَدَ السِّنِينَ وَالْحِسَابَ مَا خَلَقَ اللَّهُ ذَلِكَ إِلَّا بِالْحَقِّ يُفَصِّلُ الْآيَاتِ لِقَوْمٍ يَعْلَمُونَ ﴿٥﴾

Artinya:

“Dia-lah yang menjadikan matahari bersinar dan bulan bercahaya dan ditetapkan-Nya manzilah-manzilah (tempat-tempat) bagi perjalanan bulan itu, supaya kamu mengetahui bilangan tahun dan perhitungan (waktu). Allah tidak menciptakan yang demikian itu melainkan dengan hak. Dia menjelaskan tanda-tanda (kebesaran-Nya) kepada orang-orang yang Mengetahui.” (QS. Yunus/10 : 5)

Ada kata-kata yang digunakan dalam al-Quran untuk matahari, seperti *siraj* dan *dhiya'*, yang berarti cahaya yang kuat dan sangat. Mengenai bulan, kata-kata yang digunakan adalah *nur* dan *munir* yang berarti cahaya yang redup. Kemudian Al-Quran mengatakan secara tidak langsung bahwa proses penciptaan ini dan perputaran matahari serta bulan hendaknya tidak dianggap sebagai masalah remeh. Selanjutnya ayat diatas mengatakan, Allah tidak menciptakan yang demikian itu melainkan dengan kebenaran. Awan dan angin, bulan dan matahari bergerak dilangit agar supaya anda bisa mencari rezeki anda itu.

Di akhir ayat, Al-Quran menekankan bahwa Allah menjelaskan ayat-ayat Nya bagi mereka yang memahaminya meskipun mereka yang berpikiran picik dan tak sadar mengabaikan semua ayat itu, tanpa memahami sedikitpun dariNya. (Sayyid Quthb, 2004, hal. 32).

Di ayat lain Allah berfirman:

وَالشَّمْسِ وَضُحَاهَا ﴿١﴾ وَالْقَمَرِ إِذَا تَلَّهَا ﴿٢﴾

Artinya :

“Demi matahari dan cahayanya di pagi hari, dan bulan apabila mengiringinya,”. (QS. Asy Shams / 91: 1-2).

Dari kedua ayat tersebut dapat dijelaskan kata “*mengiringi*” adalah terjemahan dari “*tahala*” dengan asal kata “*tala*” yang berarti “*mengikuti*”, atau “*bergantung pada*”. “*Tala*” ini memiliki akar kata “*talaw*” yang berarti “*membaca dan memperdengarkannya*”, dimana kata “*talaw*” ini digunakan tidak kurang dari 60 kali dalam Al Qur’an.

Bulan adalah sesuatu yang “mengikuti” matahari. Sedangkan jika lihat di ayat lain dijelaskan bahwa orbit bulan berbeda dengan matahari, dan orbit bulan berbentuk seperti batang kurma yang tua, dimanabulan bersama-sama dengan bumi mengelilingi matahari. Ayat ini ternyata mengandung kebenaran ilmiah, dimana dibuktikan dengan ilmu pengetahuan saat ini yang menyebutkan bahwa bulan cenderung “mengikuti” matahari dikarenakan gaya tarik matahari.

Di ayat lain Allah berfirman:

وَهُوَ الَّذِي جَعَلَ لَكُمُ النُّجُومَ لِتَهْتَدُوا بِهَا فِي ظُلُمَاتِ الْبَرِّ وَالْبَحْرِ قَدْ فَصَّلْنَا الْآيَاتِ لِقَوْمٍ يَعْلَمُونَ ﴿١٧﴾

Artinya:

“Dan dialah yang menjadikan bintang-bintang bagimu, agar kamu menjadikannya petunjuk dalam kegelapan di darat dan di laut. Sesungguhnya kami Telah menjelaskan tanda-tanda kebesaran (kami) kepada orang-orang yang Mengetahui.” (QS. Al An'am/ 6: 97)

Allah SWT telah menciptakan bintang-bintang untuk keperluan manusia yang dengan itu bisa mengambil manfaat dari cahayanya, kemunculannya, dan posisinya di kegelapan malam ketika manusia berada di tengah padang pasir dan lautan sehingga bisa menemukan jalan dalam pengembaraan. Sebagian berada di depan kita dan sebagian lagi berada di belakang, di kiri dan di kanan. Bintang-bintang itu menjadi petunjuk bagi orang-orang yang bepergian menuju kota-kota, mencari arah Ka`bah,

mengenal waktu di malam hari, dan menemukan arah jalan di padang pasir dan lautan. (Andi Rosadisastra, 2007, hal. 16).

إِنَّ عِدَّةَ الشُّهُورِ عِنْدَ اللَّهِ اثْنَا عَشَرَ شَهْرًا فِي كِتَابِ اللَّهِ يَوْمَ خَلَقَ السَّمَوَاتِ وَالْأَرْضَ مِنْهَا أَرْبَعَةٌ حُرْمٌ ذَلِكَ الدِّينُ الْقَيِّمُ فَلَا تَظْلِمُوا فِيهِنَّ أَنْفُسَكُمْ وَقَتِلُوا الْمُشْرِكِينَ كَافَّةً كَمَا يُقْتَلُونَكُمْ كَافَّةً وَعَلِمُوا أَنَّ اللَّهَ مَعَ الْمُتَّقِينَ ﴿٣٦﴾

Artinya:

“Sesungguhnya bilangan bulan pada sisi Allah adalah dua belas bulan, dalam ketetapan Allah di waktu Dia menciptakan langit dan bumi, di antaranya empat bulan haram. Itulah (ketetapan) agama yang lurus, Maka janganlah kamu Menganiaya diri kamu dalam bulan yang empat itu, dan perangilah kaum musyrikin itu semuanya sebagaimana merekapun memerangi kamu semuanya, dan ketahuilah bahwasanya Allah beserta orang-orang yang bertakwa”.(QS. At Taubah / 9: 36).

Refleksi peredaran bumi mengelilingi matahari adalah perubahan kedudukan tahunan matahari di langit, yang menimbulkan pola perubahan musim tahunan. Fenomena regular (periodik) tahunan akibat perubahan kedudukan matahari itu, menyebabkan perubahan musim di belahan bumi utara dan belahan bumi selatan, dari musim panas, gugur, dingin, semi, panas lagi, atau perubahan musim di Indonesia dari musim penghujan ke musim kemarau. (Quraish Shihab, 2007, hal. 39).

Begitu seringnya pengulangan kedudukan matahari yang selama ribuan tahun itu, manusia menghafalkan pola musim, iklim dan fenomena alam lainnya, akhirnya nenek moyang kita pun juga membuat kalender tahunan bukan berdasarkan kalender Syamsiah (Masehi) atau kalender Komariah (Hijrah/Islam) tetapi berdasarkan kejadian-kejadian alam yaitu seperti musim penghujan, kemarau, musim berbunga, dan letak bintang di jagat raya, serta

pengaruh bulan purnama terhadap pasang surutnya air laut. Ketetapan waktu (kalender) yang dibuat nenek moyang kita akan hal di atas disebut dengan nama Pranata Mangsa.

Pranata mangsa (bahasa Jawa, berarti "penentuan musim") adalah semacam penanggalan yang berkaitan dengan musim menurut pemahaman suku Jawa, khususnya dari kalangan petani dan nelayan. Pemahaman yang mirip seperti ini juga dikenal oleh suku-suku lainnya di Indonesia, seperti suku Sunda dan suku Bali (dikenal sebagai Kerta Masa), atau di beberapa tradisi Eropa, seperti pada bangsa Jerman (dikenal sebagai Bauernkalender, atau "penanggalan untuk petani"). Pranata Mangsa berasal dari dua kata, yaitu Pranata yang berarti aturan, dan Mangsa yang berarti musim atau waktu. Jadi Pranata Mangsa adalah aturan waktu yang digunakan para petani sebagai penentuan atau mengerjakan sesuatu pekerjaan. Hal ini dipelopori oleh seorang raja Pakubuwono VII dan dimulai sejak 22 Juni 1856. Contohnya melaksanakan usaha tani seperti bercocok tanam, atau melaut sebagai nelayan, merantau mungkin juga berperang. Biasanya digunakan oleh para petani pedesaan, yang didasarkan pada naluri saja, dari leluhur yang sebetulnya belum tentu dimengerti asal-usul dan bagaimana uraian satu-satu kejadian di dalam setahun. Walau begitu bagi para petani tetap dipakai dan sebagai patokan untuk mengolah pertanian. Pranata mangsa adalah aturan waktu musim, yang berdasar pada solar kalender. Mungkin kalender Pranata Mangsa ini termasuk dari 40 sistem kalender yang oleh sebuah studi tahun 1987 digunakan di dunia dan dikenal dalam pergaulan internasional. Tapi kalau kita

spesifikasikan lagi, maka hanya dikategorikan ke dalam tiga mazhab besar, yaitu sistem kalender masehi (syamsiah) atau solar calendar, kalender qomariah (lunar calendar), dan lunisolar.

Seiring dengan berjalannya waktu dan kemajuan dalam bidang teknologi navigasi, manusia telah menciptakan bererapa alat navigasi sehingga mereka tidak lagi bergantung kepa posisi bintang-bintang untuk menentukan arah, diantara adalah peta, kompas, dan *GPS*. Akan tetapi bukan berarti bintang-bintang tersebut sudah tidak berguna, sebab masih banyak para ilmuan yang mencoba mempelajari guna mengetahui manfaat lain dari bintang-bintang tersebut. Sehingga dari pengembangan ilmu pengetahuan tersebut muncullah suatu cabang keilmuan yang dikenal dengan sistem informasi geografis. (Abdurrahman R Effendi dan Gina Puspita, 2007).

GIS (Geographic Information System) atau Sistem Informasi Berbasis Pemetaan dan Geografi adalah sebuah alat bantu manajemen berupa informasi berbantuan komputer yang berkait erat dengan sistem pemetaan dan analisis terhadap segala sesuatu serta peristiwa-peristiwa yang terjadi di muka bumi. Teknologi *GIS* mengintegrasikan operasi pengolahan data berbasis database yang biasa digunakan saat ini, seperti pengambilan data berdasarkan kebutuhan, serta analisis statistik dengan menggunakan visualisasi yang khas serta berbagai keuntungan yang mampu ditawarkan melalui analisis geografis melalui gambar-gambar petanya. Komponen *GIS* adalah sistem komputer, data *geospatial* dan pengguna, seperti diperlihatkan pada Gambar berikut:



Gambar 2.1 Contoh Aplikasi GIS. (Muhammad Zaenal Arifin, 2012)

Data yang diolah pada *GIS* ada 2 macam yaitu data geospasial atau yang biasanya disebut data spasial dan data non-spasial (atribut). Jika pada gambar diatas data atribut tidak digambarkan karena memang dalam *GIS* yang dipentingkan adalah tampilan data secara spasial. Tetapi sebenarnya pada *GIS* kadang-kadang juga melibatkan data atribut baik secara langsung maupun secara tidak langsung. (Muhammad Zaenal Arifin, 2012)

Mobile GIS mempunyai beberapa kelebihan dan kekurangan yang diantaranya sebagai berikut:

- 1) Kelebihan *Mobile GIS*
 - a) Bersifat *Mobile*, bisa digunakan dimana saja.
 - b) Lebih efisien, dan bisa dipadukan dengan *GPS*.
 - c) Penggunaan lebih mudah, ringan dan mudah dibawa.
- 2) Kekurangan *Mobile GIS*
 - a) User interface ditampilkan dalam layer kecil.
 - b) Pengaksesan terhadap data *GIS* dan fungsifungsinya tidak selengkap dan sekomplek di *desktop*.

2.3 GPS

GPS (Global Positioning System) adalah sistem satelit navigasi dan penentuan posisi yang dimiliki dan dikelola oleh Amerika Serikat. Sistem ini didesain untuk memberikan posisi dan kecepatan tiga-dimensi serta informasi mengenai waktu, secara kontinyu di seluruh dunia tanpa bergantung waktu dan cuaca, bagi banyak orang secara simultan. Saat ini *GPS* sudah banyak digunakan orang di seluruh dunia dalam berbagai bidang aplikasi yang menuntut informasi tentang posisi, kecepatan, percepatan ataupun waktu yang teliti. *GPS* dapat memberikan informasi posisi dengan ketelitian bervariasi dari beberapa millimeter (orde nol) sampai dengan puluhan meter.

Beberapa kemampuan *GPS* antara lain dapat memberikan informasi tentang posisi, kecepatan, dan waktu secara cepat, akurat, murah, dimana saja di bumi ini tanpa tergantung cuaca. Hal yang perlu dicatat bahwa *GPS* adalah satu-satunya sistem navigasi ataupun sistem penentuan posisi dalam beberapa abad ini yang memiliki kemampuan handal seperti itu.

Ketelitian dari *GPS* dapat mencapai beberapa mm untuk ketelitian posisinya, beberapa cm/s untuk ketelitian kecepatannya dan beberapa nanodetik untuk ketelitian waktunya. Ketelitian posisi yang diperoleh akan tergantung pada beberapa faktor yaitu metode penentuan posisi, geometri satelit, tingkat ketelitian data, dan metode pengolahan datanya.

Secara umum produk dari *GPS* adalah posisi, kecepatan, dan waktu. Selain itu ada beberapa produk lainnya seperti percepatan, azimuth, parameter attitude, *TEC (Total Electron Content)*, *WVC (Water Vapour Content)*, Polar

motion parameters, serta beberapa produk yang perlu dikombinasikan dengan informasi eksternal dari sistem lain, produknya antara lain tinggi *ortometrik*, *undulasi geoid*, dan *defleksi vertikal*. (Muhammad Zaenal Arifin, 2012)

2.4 Android

Definisi Android *adalah* sebuah sistem operasi untuk perangkat Smartphone berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. (Safaat, 2011).

2.4.1 The Dalvik Virtual Machine (DVM)

Android berjalan di dalam *Dalvik Virtual Machine* (DVM) bukan di Java Virtual Machine (JVM), DVM adalah "*register bases*" sementara JVM adalah "*stack based*". DVM menggunakan kernel linux untuk menangani fungsionalitas tingkat rendah termasuk keamanan, threading, dan proses serta manajemen memori. ini memungkinkan untuk menulis Aplikasi C/C+ sama halnya seperti pada OS Linux kebanyakan.

2.4.2 Android SDK (Software Development Kit)

Definisi Android SDK adalah *tools API*(*Applikation Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman java. Android merupakan *subset* perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci. Secara garis besar Arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut (Safaat, 2011):

2.4.2.1 Application dan Widgets

Application dan *Witgets* ini adalah lapisan dimana kita berhubungan dengan aplikasi saja, dimana biasanya kita *download*

aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di lapisan terdapat aplikasi inti termasuk klien *email*, program SMS, kalender, peta, *browser*, kontak, dan lain- lain.

2.4.2.2 Application Frameworks

Android adalah "*Open Development platform*" yaitu android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi resources, menjalankan *service background*, mengatur alarm, dan menambahkan status notifications, dan sebagainya.

2.4.2.3 Libraries

Libraries ini adalah layer dimana fitur- fitur Android berada, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya. Berjalan diatas Kernel, layer ini meliputi berbagai library C/C++ inti seperti Libe dan SSL, serta:

1. *Libraries media* untuk pemutaran media audio dan video.
2. *Libraries* untuk manajemen tampilan.
3. *Libraries Graphics* mencakup SGL, dan openGL untuk grafis 2D dan 3D.
4. *Libraries LiveWebcore* mencakup modern web browser dengan engine embedee web tampilan.
5. *Libraries 3D* yang mencakup implementasi OpenGL ES LO APTs.

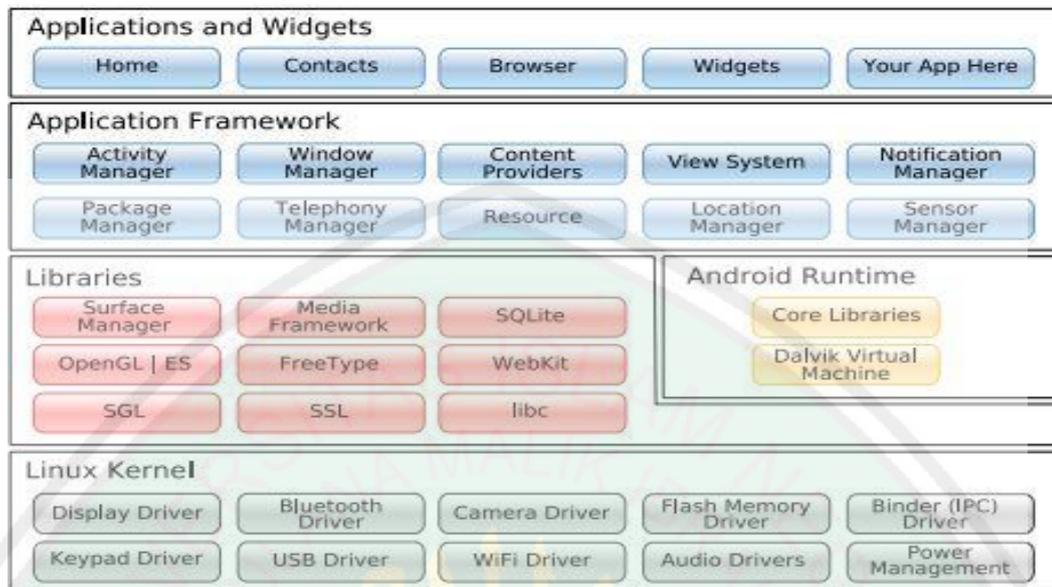
2.4.2.4 Andoid Run Time

Layer yang membuat aplikasi android dapat dijalankan dimana dalam prosesnya menggunakan implementasi linux. Di dalam Android Run Time dibagi menjadi dua bagian yaitu:

1. *Core Libraries*: sebuah libraries yang berfungsi untuk menterjemahkan bahasa java/c yang ditangani oleh *Core Libraries*.
2. *Dalvik Virtual Machine*: virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi- fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat linux kernel untuk melakukan threading dan manajemen tingkat rendah.

2.4.2.5 Linux Kernel

Linux kernel berisi file- file sistem yang mengatur sistem *processing, memory, resource, drivers*, dan sistem- sistem operasi android lainnya. linux kernel yang digunakan android adalah linux kernel release 2.2.



Gambar 2.2 Arsitektur *Android* (Safaat, 2011)

2.4.3 Fundamental Aplikasi

Aplikasi android ditulis dalam bahasa pemrograman java. Kode java dikompilasi bersama dengan data file resource yang dibutuhkan oleh aplikasi, dimana prosesnya di *package oleh tools* yang dinamakan "apt tools" kedalam paket android sehingga menghasilkan file dengan ekstensi apk. Ada enam komponen aplikasi android, yaitu (Safaat, 2011):

2.4.3.1 Activities

Suatu activity akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi android bisa jadi hanya memiliki satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut. Untuk pindah dari satu *activity ke activity* lain kita dapat melakukannya dengan satu even, misalnya click tombol, memilih opsi atau menggunakan *triggers*

tertentu. Secara hirarki sebuah windows activity dinyatakan dengan method `activity.setContentTampilan()`. `ContentTampilan` adalah objek yang berada pada root hirarki.

2.4.3.2 Service

Service tidak memiliki *graphic user interface* (GUI), tetapi service berjalan secara *background*, sebagai contoh dalam memainkan musik, service mungkin memainkan musik atau mengambil data dari jaringan, tetapi setiap service harus berada dalam kelas induknya. Misalnya, media player sedang memutar lagu dari list yang ada, aplikasi ini akan memiliki dua atau lebih activity yang memungkinkan pengguna untuk memilih lagu misalnya, atau menulis sms sambil *player* sedang jalan. Untuk menjaga musik tetap dijalankan, activity *player* dapat menjalankan service. Service dijalankan pada thread utama dari proses aplikasi.

2.4.3.3 Broadcast Receiver

Broadcast *receiver* berfungsi menerima dan bereaksi untuk menyampaikan notifikasi zona waktu berubah, baterai low, gambar telah selesai diambil oleh kamera, atau perubahan pustaka bahasa yang digunakan. Aplikasi juga dapat menginisiasi *broadcast* misalnya memberikan informasi pada aplikasi lain bahwa ada data yang telah diunduh ke perangkat dan siap untuk digunakan.

Broadcast *receiver* tidak memiliki *user interface* (UI), tetapi memiliki sebuah activity untuk merespon informasi yang mereka terima, atau mungkin menggunakan *Notification Manager* untuk

memberitahu kepada pengguna, seperti lampu latar atau *vibrating* (getaran) perangkat, dan lain sebagainya.

2.4.3.4 Content Provider

Content provider membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam file sistem seperti database SQLite. *Content provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*, misalnya ketika kita menggunakan aplikasi yang membutuhkan peta (Map), atau aplikasi yang membutuhkan untuk mengakses data kontak dan navigasi maka disinilah fungsi *content provider*.

2.4.4 Emulator

Emulator atau lebih tepatnya piranti lunak emulator memungkinkan suatu program atau piranti lunak yang dibuat pada awalnya oleh suatu sistem komputer (arsitektur dan sistem operasi) dan untuk dijalankan dalam sistem itu (atau dijalankan dalam suatu sistem yang didedikasikan), dapat dijalankan dalam sistem komputer yang sama sekali berbeda. Sebagai contoh suatu program Windows dapat dijalankan di sistem operasi Linux dengan menggunakan piranti lunak emulator Wine. Ada pula program yang mengemulasikan suatu komputer dalam komputer, misalnya VMware. Contoh lain adalah program-program emulator untuk menjalankan permainan komputer yang awalnya hanya bisa dijalankan pada konsolnya masing-masing, misalnya Nintendo, Atari, PlayStation, Xbox dan lain-lain.

Dalam hal ini emulator yang digunakan adalah emulator salah satu *tools* yang terdapat pada eclipse indigo. Emulator ini berfungsi untuk

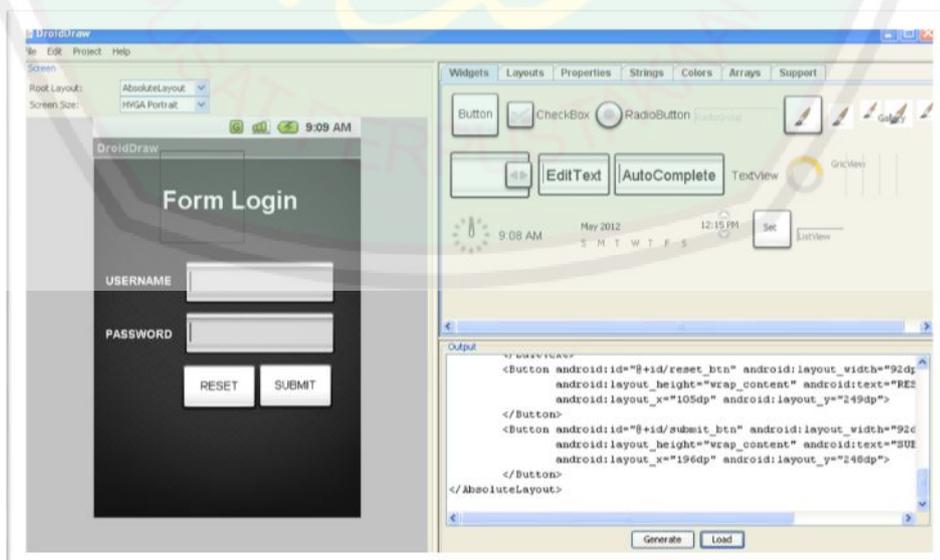
mensimulasikan device di dalam computer, sehingga aplikasi yang dibuat bisa dijalankan dan diuji coba di komputer tanpa harus di-*install* ke dalam *smartphone* terlebih dahulu.



Gambar 2.3 Emulator

2.4.5 Droiddraw

Droiddraw adalah tools desain aplikasi android yang bersifat *free* dan dapat di-*download* di <http://www.droiddraw.org>. dengan tools ini dapat dengan mudah mendesain tampilan android.



Gambar 2.4 Droiddraw

2.4.6 Komponen-komponen Android

2.4.6.1 User Interface (UI)

Secara umum arsitektur *user interface* (UI) pada aplikasi android adalah user interface yang meliputi *activity* dan *user interface* yang terdiri dari komponen. Semua yang berhubungan dengan user interface pada aplikasi android biasanya berada pada lokasi `res/layout/filename.xml` di mana coding java untuk memanggilnya yang dikenal dengan `R.layout.filename`.

Komponen- komponennya adalah sebagai berikut (Safaat, 2011):

1. <ViewGroup>

Kumpulan tampilan dimana kita dapat menentukan tata letak komponen tampilan secara berbeda, seperti `LinearLayout`, `RelativeLayout`, `FrameLayout`, serta Tabulasi.

2. <View>

Sama seperti <ViewGroup> tetapi <View> lebih dikenal dengan “individual UI component”, atributnya juga minimal terdiri dari tiga yang sama persis dengan atribut yang dimiliki oleh <ViewGroup>.

3. <RequestFocus>

Element kosong yang bias kita definisikan di dalam <View>

4. <include>

Memasukkan file *layout* ke dalam *layout*. Atributnya sama dengan <ViewGroup> dan <View> tetapi ada satu tambahan atribut yaitu <resource> yang berfungsi untuk menentukan file *layout*-nya.

2.4.6.2 AndroidManifest.xml

File AndroidManifest.xml diperlukan oleh setiap aplikasi android, file ini berada pada *folder root* aplikasi, file ini mendeskripsikan variable global dari paket aplikasi yang kita gunakan.

Komponen- komponennya adalah sebagai berikut:

1. <manifest>

Titik root utama dari AndroidManifest.xml, berisi atribut package aplikasi serta paket *activity* dalam program android kita.

2. <user-permission>

Menjelaskan tentang *user permission/security permission* yang harus diberikan agar aplikasi dapat berjalan sebagaimana mestinya misalnya apabila kita menggunakan resources yang tersedia dari system, seperti ketika mengirim sms, dan sebagainya.

3. <permission>

Menjelaskan batasan tentang *user permission/security permission*.

4. <instrumentation>

Mendeklarasikan komponen instrument yang tersedia untuk menguji fungsionalitas dari paket aplikasi yang kita gunakan dalam aplikasi android kita.

5. <application>

Elemen *root* yang berisi deklarasi aplikasi android kita.

6. <intent filter>

Mendeklarasikan intent yang dibutuhkan oleh aplikasi android yang kita gunakan, atribut- atribut bias diberikan disini men *-supply* label, *icon*, data dan informasi yang kita gunakan dalam aplikasi android.

7. <action>

Berisi tentang *action type* yang didukung oleh komponen-komponen yang berada dalam aplikasi android.

8. <category>

Mendeklarasikan kategori- kategori yang didukung oleh aplikasi android.

9. <data>

Mendeklarasikan tipe MIME, URL, *authority* penggunaan URL, serta penentuan *path* yang digunakan dalam URL.

10. <meta-data>

Mendeklarasikan meta data yang kita butuhkan sebagai tambahan data yang ada kita gunakan dalam aplikan android.

11. <receiver>

Mendeklarasikan dimana aplikasi kita diberikan informasi mengenai sesuatu perubahan atau aksi yang terjadi, seperti menerima SMS.

12. <service>

Mendeklarasikan komponen yang dapat berjalan sebagai service (berjalan di *background*)

13. <provider>

Mendeklarasikan komponen-komponen yang mengelola data dan mempublikasikannya untuk dikelola/ dipakai oleh aplikasi lain.

2.4.6.3 Activity

Activity merupakan *public class* dalam aplikasi android. Setiap *activity* merupakan sesuatu yang unik atau *single*, yang ditujukan untuk menghandle macam-macam hal yang bisa dilakukan oleh *user*. Umumnya, *activity* berhubungan dengan user dimana *activity* menciptakan windows atau UI yang mana ditampilkan dengan concepts *setContentTampilan(Tampilan)*. Ada 2 *method* yang pasti dimiliki oleh satu *activity* yaitu:

1. *onCreate* untuk menginstalasi suatu *activity*, biasanya dipanggil dengan perintah *setContentTampilan(int)* untuk *resource* yang kita deferensikan di layout UI, dan perintah *findTampilanById(int)* untuk memanggil widget yang dibutuhkan UI untuk berinteraksi dengan aplikasi.

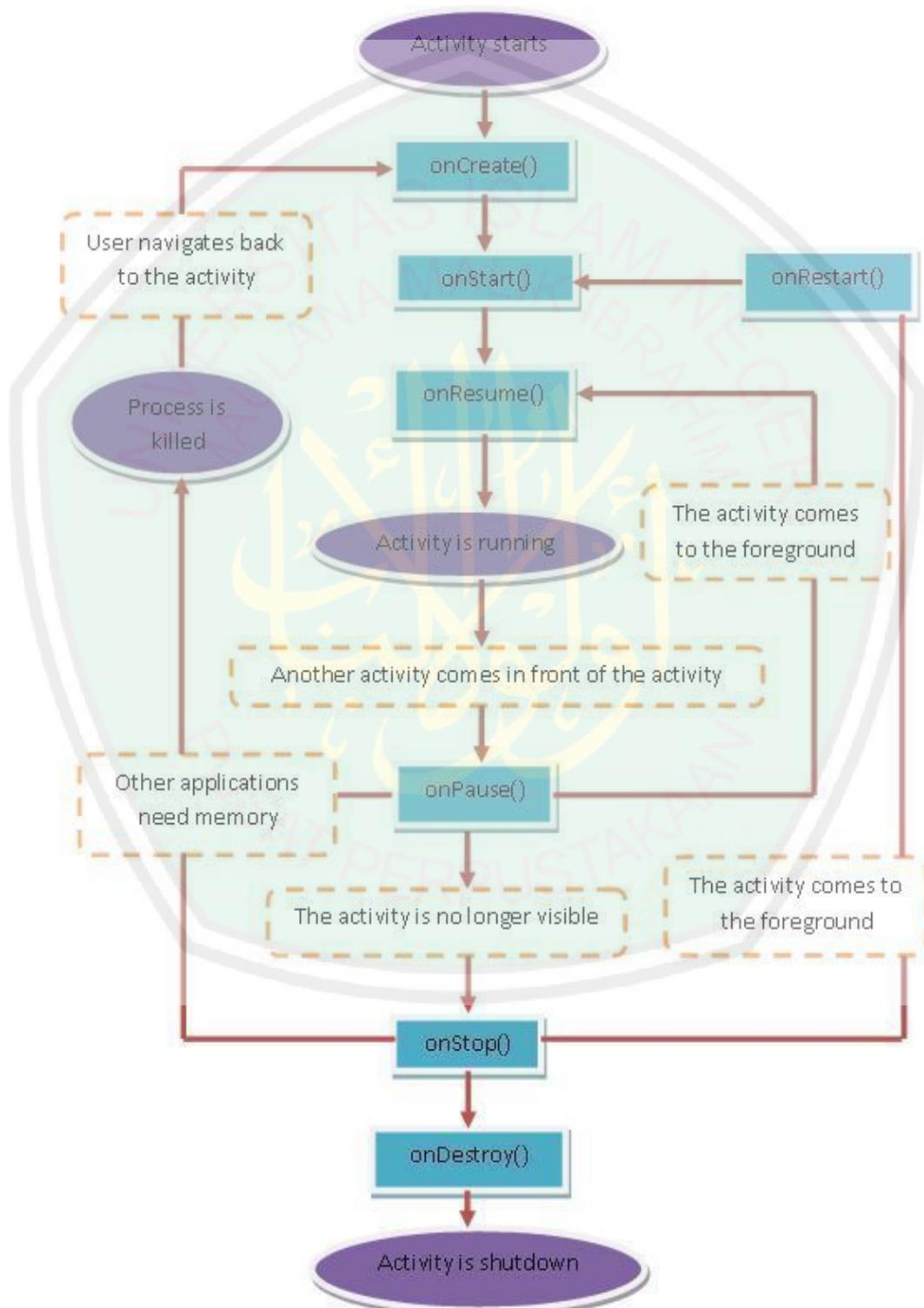
2. *OnPause* untuk menyatakan ketika *user* meninggalkan suatu *activity*.

2.4.6.4 Siklus Activity

Activity aplikasi android dikelola dengan system yang dikenal dengan *activity stack*. Ketika suatu *activity start*, *activity* diletakkan pada *stack* yang paling atas dan *activity- activity* yang sudah jalan berada dibawahnya dan akan terus berada pada posisi atas *stack* sampai muncul *activity* baru. Pada dasarnya *activity* memiliki empat keadaan, yaitu (Safaat, 2011):

1. *Active* atau *running*, jika *activity* berada pada posisi atas *stack*.
2. *Pause*, jika *activity* tidak dipakai atau dibutuhkan pada suatu saat tertentu, tetapi *activity* itu masih ada atau *visible*, ketika *activity* baru yang ditangani oleh system *activity* yang lama disebut *pause* dan masih berada di *memory*.
3. *Stopped*, jika *activity* sudah tidak dipakai dan digantikan oleh *activity* lain, *activity* yang sudah *stopped* tidak akan pernah dipanggil lagi dan secara permanen *memory* pun tidak menyimpan info mengenai *activity* ini.
4. *Restart*, jika *activity* *pause* atau *stopped*, system dapat *drop activity* ini dari *system memory*, dan ketika user membutuhkan *activity* tersebut, *activity* akan kembali keadaan awal, artinya *activity* tersebut mengalami proses *restart*.

Berikut ini adalah gambaran mengenai activity lifecycle atau siklus dari *activity* di dalam android system.



Gambar 2.5 Lifecycle Activity (Safaat, 2011)

Berdasarkan gambar *lifecycle activity* di atas, secara umum dapat kita jelaskan bahwa *looping* (perulangan) yang mungkin dialami oleh *activity* ada tiga perulangan, yaitu:

1. *Entire lifetime*, yaitu *activity* yang terjadi mulai dari `onCreate()` sampai dengan `onDestroy()`. Biasanya *activity* ini akan dibuat setup *global* ketika mendefinisikannya.
2. *Visible lifetime*, yaitu *activity* yang terjadi mulai dari `onCreate` sampai dengan `onStop()`.
3. *Foreground lifetime*, yaitu *activity* yang terjadi diantara `onResume` dan `onPause()`.

Tabel 2.1 *Lifecycle Activity* (Safaat, 2011)

Method	Description	Next
onCreate()	Ketika sebuah <i>activity</i> dibuat, pada method inilah kita melakukan inisialisasi seperti <code>create tampilan</code> , <code>list data</code> , dan lain sebagainya. Method onCreate() selalu diikuti oleh onStart() .	onStart()
onRestart()	Ketika sebuah <i>activity</i> dihentikan, dan merupakan prioritas untuk memanggil <i>activity</i> itu kembali. Method ini selalu diikuti oleh onStart()	onStart()
onStart()	Ketika sebuah <i>activity</i> dipanggil sebelum diperlihatkan ke <i>user</i> , method ini diikuti oleh onResume() atau onStop()	onResume() or onStop()
onResume()	Ketika sebuah <i>activity</i> start atau memulai melakukan interaksi dengan <i>user</i> , pada saat ini <i>activity</i> berada pada posisi teratas dari <i>activity stack</i> yang mana <i>user</i> akan melakukan input. Selalu diikuti oleh onPause()	onPause()
Method	Description	Next
onPause()	Ketika sebuah <i>activity</i> lainnya dipanggil atau dimulai, method ini digunakan ketika data tidak harus disimpan kedalam system secara	onResume() or onStop()

	permanen, method ini diikuti oleh onResume() atau onStop() .	
onStop()	Ketika sebuah activity tidak lagi dibutuhkan atau tidak terlihat lagi oleh <i>user</i> , method ini diikuti oleh onRestart() atau onDestroy()	onRestart() or onDestroy()
onDestroy()	Ketika sebuah activity secara permanen tidak lagi dibutuhkan (dihancurkan), dapat juga dilakukan dengan fungsi finish() atau isFinishing() .	-

2.4.6.5 Widget

Paket widget pada dasarnya merupakan visualisasi dari elemen *User Interface* (UI) yang digunakan pada layar aplikasi android dimana kita dapat merancang sendiri sesuai kebutuhan. *Widget* di dalam android ditampilkan dengan konsep Tampilan. Di mana aplikasi android pada umumnya menggunakan *widget* sebagai layout XML. Berikut ini adalah *file- file* yang umumnya kita butuhkan apabila kita membuat *widget*:

1. File java, berupa *file* java dimana merupakan *file* yang mengimplementasikan aksi dari *widget*. Jika didefinisikan suatu widget beserta posisinya dilayar yang didefinisikan dari file XML, kita harus melakukan *coding* di *file* java dimana kita dapat mengambil semua nilai atribut dari tata letak *file* XML yang didefinisikan.
2. File XML, sebuah file XML yang berada di *res/values/* yang mendefinisikan komponen elemen- elemen XML yang digunakan untuk inialisasi *widget* serta atribut yang mendukungnya (*file- file* yang dikenal dengan *main.xml*)

3. Layout XML(optional), *file* ini tidak harus ada, *file* XML berada di dalam *res/layout/* yang menggambarkan atau penambahan keterangan pada layout *widget*.

2.4.6.6 Kumpulan Widgets

Kumpulan widgets yang memiliki struktur tersendiri- tersendiri biasanya disebut dengan *containers*. Jika kita berjalan dalam sebuah tampilan dimana terdapat beberapa *widget* kita harus menentukan *layout* yang digunakan. Seperti kebanyakan GUI yang memiliki *layout management* untuk menampilkan *containers*, seperti *layout* *BoxLayout* di *java/swing*, *android* juga memiliki beberapa *layout management* yaitu sebagai berikut (Safaat, 2011):

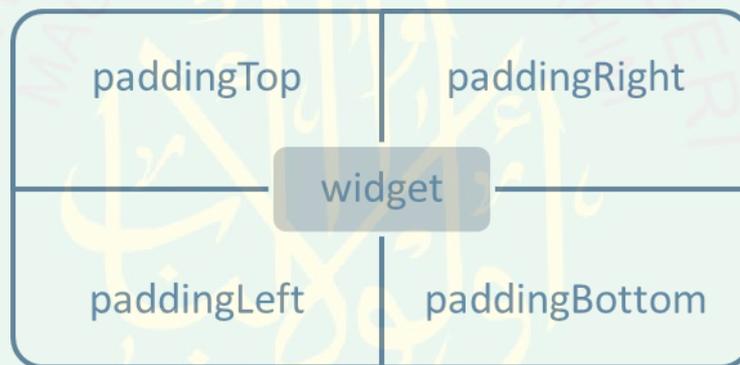
1. Linear

Penggunaannya mirip dengan model *box* dari *java/swing*, yaitu menampilkan *widgets* atau *containers* dalam posisi baris atau kolom, satu persatu. Konfigurasi *LinearLayout* mempunyai 5 konten yaitu:

- a. *Orientation*, mengindikasikan *layout* ditampilkan berdasarkan baris atau kolom.
- b. *Fill Model*, bentuk tampilan *element/widgets* terhadap *space* yang tersedia.
- c. *Weight*, mengatur settingan *android.layout_width* (untuk baris) dan *android.layout_height* (untuk kolom).
- d. *Gravity*, secara default pada tampilan dengan *LinearLayout* adalah “left and top aligned”. Jika kita

menginginkan tampilan yang berbeda kita bias mengatur gravity sesuai kebutuhan. Nilai- nilai gravity adalah `left`, `center_horizontal`, `right` for `left aligned`, `centered` dan `right-aligned`.

- e. `Padding`, `padding` memiliki empat nilai standard yaitu `paddingLeft`, `paddingRight`, `paddingTop` dan `paddingBottom`.



Gambar 2.6 Posisi *Padding* Dari Sebuah *Widgets* (Safaat, 2011)

2. *Relative*

RelativeLayout adalah *layout* dimana jika kita membutuhkan penempatan widget A dibawah kiri dari widget B atau dibawah sebelah kanan, dan penempatan lainnya.

RelativeLayout standar nya mempunyai dua layout, yaitu:

f. *Relative to Container*

Terdiri dari beberapa position, yaitu:

- 1) `android:layout_alignParentTop`
- 2) `android:layout_alignParentBottom`

- 3) `android:layout_alignParentLeft`
- 4) `android:layout_alignParentHorizontal`
- 5) `android:layout_centerHorizontal`
- 6) `android:layout_centerVertical`
- 7) `android:layout_centerInParent`

g. Relative to widget

Terdiri dari beberapa position, yaitu:

- 1) `android:layout_above`
- 2) `android:layout_below`
- 3) `android:layout_toLeftOf`
- 4) `android:layout_toRightOf`

3. *Tabul* yang lebih kita kenal dengan *layout table* di HTML.
4. *Scrollwork* adalah membuat *layout* yang bias dilakukan *scroll* baik *scroll* naik maupun *scroll* turun.

2.4.6.7 Message Box

Message box yang dikenal di android terbagi dua jenis yaitu:

1. Toasts, adalah *message* atau pesan yang muncul dalam waktu sebentar dan kemudian hilang tanpa mempunyai *user interaction* dengan *user*, biasa ini merupakan focus dari *activity* yang *active*.
2. Alert, adalah *dialog box* atau *message* yang mempunyai *user interaction* dengan *user*, biasa terdiri dari 3 bagian yaitu:

- a. `setMessage()` berisi tulisan pesan atau message
- b. `SetTitle()` atau `setIcon()`, konfigurasi title atau icon dari dialog box.
- c. `SetPositiveButton()`, `setNeutralButton()`, `setNegativeButton()`, berisi posisi serta metode clicknya.

2.4.6.8 More View

Pada *bagian more view* ini akan dijelaskan tentang *layout user interface* yang belum dijelaskan pada bagian *user interface* dan bagian *widget*. *Layout* tersebut adalah sebagai berikut:

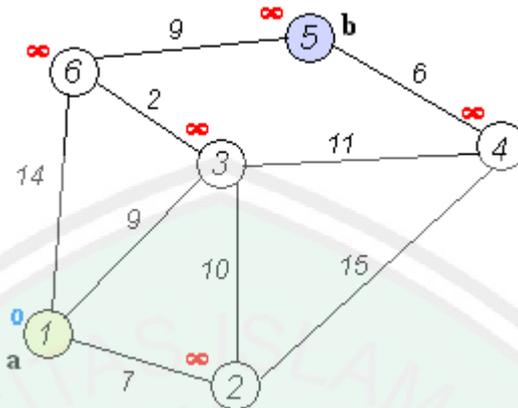
1. *TimePicker view*, *Picker view* adalah tampilan *layout* tanggal dan waktu. Fungsi *picker tampilan* dibagi menjadi dua bagian yaitu *timepicker* dan *datepicker view*. *Time picker* ini memperbolehkan *user* untuk seleksi waktu dalam sehari yaitu 24 jam (pm/am).
2. *Date picker view* adalah *layout* yang memperbolehkan *user* untuk melakukan seleksi tanggal (tanggal, bulan, tahun).
3. *Spinner view* adalah *layout* yang menampilkan list dari item pada saat tertentu dan memperbolehkan *user* untuk memilihnya.
4. *Gallery view* adalah *layout* yang menampilkan item- item dalam bentuk *image*.

5. *ImageSwitcher* view hampir sama dengan *ImageView*, hanya perbedaan cara menampilkan perubahan atau pergantian gambar yang ditampilkan pada layar.
6. Context Menu dan Options Menu, perbedaan *context menu* dan *options menu* adalah *context menu* biasanya menampilkan menu dengan aksi meng-click menu tersebut, sedangkan *option menu* adalah *layout* yang menampilkan menu dengan aksi meng-click/ atau menekan tombol menu.
7. *AnalogClock* dan *DigitalClock* adalah *layout* untuk menampilkan jam analog. Sedangkan *digitalclock* adalah *layout* yang menampilkan jam digital.

2.5 Algoritma Dijkstra

Algoritma *Dijkstra*, (dinamai menurut penemunya, seorang ilmuwan komputer, *Edsger Dijkstra*), adalah sebuah algoritma rakus (*greedy algorithm*) yang dipakai dalam memecahkan permasalahan jarak terpendek (*shortest path problem*) untuk sebuah graf berarah (*directed graph*) dengan bobot-bobot sisi (*edge weights*) yang bernilai tak-negatif. (Rinaldi Munir, 2010)

Algoritma ini bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Misalkan titik menggambarkan gedung dan garis menggambarkan jalan, maka algoritma Dijkstra melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik.



Gambar 2.7 Contoh keterhubungan antar titik dalam algoritma Dijkstra

Pertama-tama tentukan titik mana yang akan menjadi node awal, lalu beri bobot jarak pada node pertama ke node terdekat satu per satu, Dijkstra akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap. Inilah urutan logika dari algoritma Dijkstra:

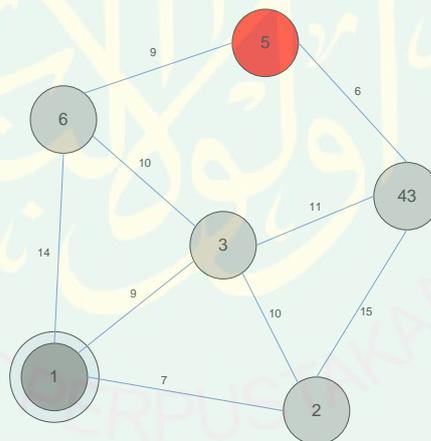
1. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga terhadap node lain (belum terisi)
2. Set semua node “Belum terjamah” dan set node awal sebagai “Node keberangkatan”
3. Dari node keberangkatan, pertimbangkan node tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan. Sebagai contoh, jika titik keberangkatan A ke B memiliki bobot jarak 6 dan dari B ke node C berjarak 2, maka jarak ke C melewati B menjadi $6+2=8$. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.
4. Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah terjamah sebagai “Node terjamah”. Node terjamah

tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.

5. Set “Node belum terjamah” dengan jarak terkecil (dari node keberangkatan) sebagai “Node Keberangkatan” selanjutnya dan lanjutkan dengan kembali ke step 3.

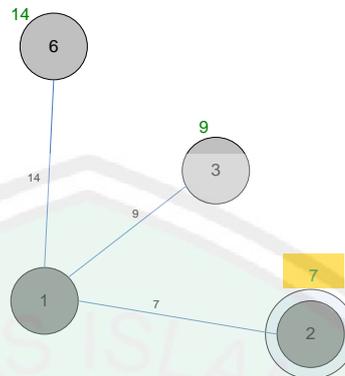
Dibawah ini penjelasan langkah per langkah pencarian jalur terpendek secara rinci dimulai dari node awal sampai node tujuan dengan nilai jarak terkecil.

1. Node awal 1, Node tujuan 5. Setiap edge yang terhubung antar node telah diberi nilai



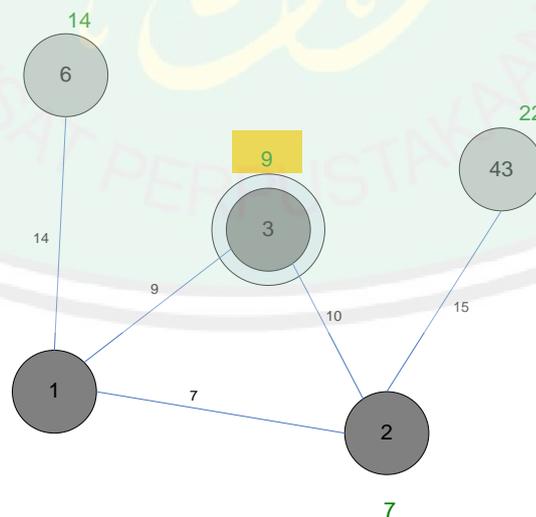
Gambar 2.8 Contoh kasus Dijkstra - Langkah 1

2. Dijkstra melakukan kalkulasi terhadap node tetangga yang terhubung langsung dengan node keberangkatan (node 1), dan hasil yang didapat adalah node 2 karena bobot nilai node 2 paling kecil dibandingkan nilai pada node lain, nilai = 7 (0+7).



Gambar 2.9 Contoh kasus Dijkstra - Langkah 2

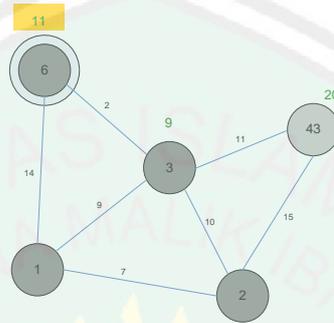
- Node 2 diset menjadi node keberangkatan dan ditandai sebagai node yang telah terjamah. Dijkstra melakukan kalkulasi kembali terhadap node-node tetangga yang terhubung langsung dengan node yang telah terjamah. Dan kalkulasi dijkstra menunjukkan bahwa node 3 yang menjadi node keberangkatan selanjutnya karena bobotnya yang paling kecil dari hasil kalkulasi terakhir, nilai 9 ($0+9$).



Gambar 2.10 Contoh kasus Dijkstra - Langkah 3

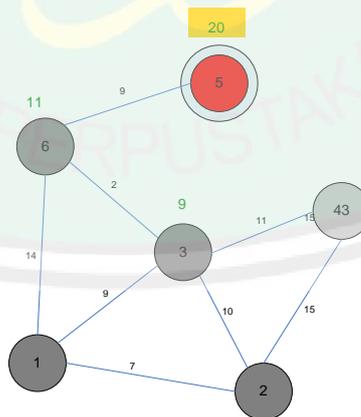
- Perhitungan berlanjut dengan node 3 ditandai menjadi node yang telah terjamah. Dari semua node tetangga belum terjamah yang terhubung

langsung dengan node terjamah, node selanjutnya yang ditandai menjadi node terjamah adalah node 6 karena nilai bobot yang terkecil, nilai 11 (9+2).



Gambar 2.11 Contoh kasus Dijkstra - Langkah 4

- Node 6 menjadi node terjamah, dijkstra melakukan kalkulasi kembali, dan menemukan bahwa node 5 (node tujuan) telah tercapai lewat node 6. Jalur terpendeknya adalah 1-3-6-5, dan nilai bobot yang didapat adalah 20 (11+9). Bila node tujuan telah tercapai maka kalkulasi dijkstra dinyatakan selesai.

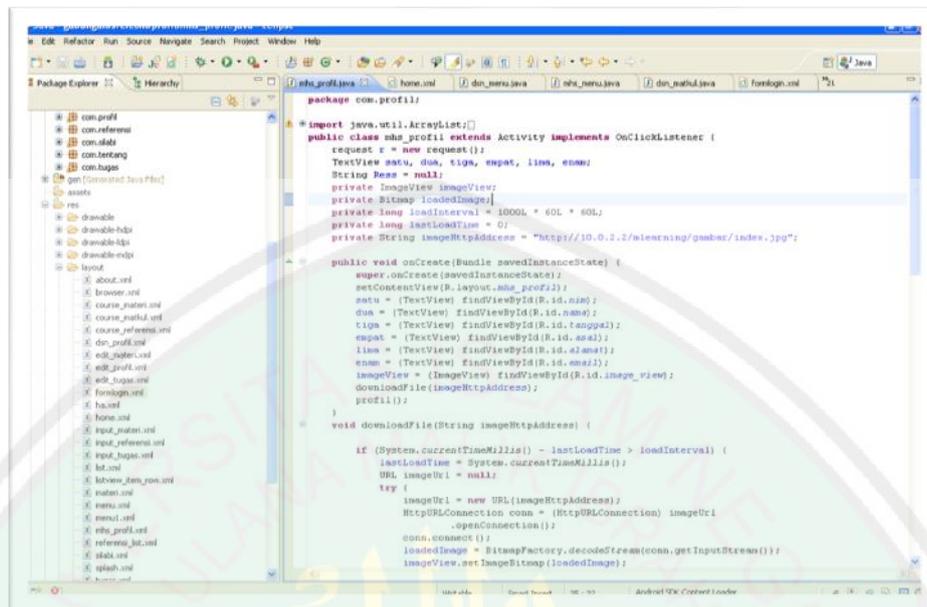


Gambar 2.12 Contoh kasus Dijkstra - Langkah 5

2.6 Eclipse Integrated Development Environment (IDE)

Integrated Development Environment (IDE) adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tujuan dari *IDE* adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak. *Eclipse* adalah sebuah *IDE (Integrated Development Environment)* untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*). Berikut ini adalah sifat-sifat dari Eclipse antara lain:

- 1) *Multi-platform*: Target sistem operasi *Eclipse* adalah *Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X*.
- 2) *Multi-language*: *Eclipse* dikembangkan dengan bahasa pemrograman *Java*, akan tetapi *Eclipse* mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti *C/C++, Cobol, Python, Perl, PHP* dan sebagainya.
- 3) *Multi-role*: Selain sebagai *IDE* untuk pengembangan aplikasi, *Eclipse* dapat digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, uji perangkat lunak, pengembangan web dan sebagainya.



Gambar 2.13 Jendela Kerja Eclipse

Eclipse pada saat ini merupakan salah satu *IDE* populer dikarenakan gratis dan *open source*, yang berarti setiap orang dapat melihat dan memodifikasi *source code* perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan plugin. *Eclipse* awalnya dikembangkan oleh perusahaan *IBM (International Business Machines)* untuk menggantikan perangkat lunak *IBM Visual Age for Java 4.0*. Produk ini diluncurkan oleh *IBM* pada tanggal 5 November 2001, yang menginvestasikan sebanyak *US\$ 40* juta untuk pengembangannya. Semenjak itu konsorsium *Eclipse Foundation* mengambil alih untuk pengembangan *Eclipse* lebih lanjut dan pengaturan organisasinya. Sejak tahun 2006 *Eclipse Foundation* secara rutin merilis versi *Eclipse* setiap tahun. Setiap rilis kemudian diberi kode sesuai dengan berbagai namabintang dari planet Jupiter. Diantaranya adalah sebagai berikut :

- 1) Callisto (26 Juni 2006).
- 2) Europa (27 Juni 2007).
- 3) Ganymede (25 Juni 2008).
- 4) Galileo (24 Juni 2009).
- 5) Helios (23 Juni 2010, sedang tahap perencanaan).

Di samping itu *Eclipse* juga didistribusikan dalam beberapa proyek sesuai kebutuhan spesifik :

- 1) Sebagai Java IDE (fungsi utama).
- 2) C++ IDE.
- 3) Java mobile/embedded device IDE.
- 4) Web development, dan sebagainya. (Muhammad Zaenal Arifin, 2012)

2.7 Google Maps

Google Maps merupakan sebuah layanan peta dunia virtual berbasis web yang disediakan oleh Google. Layanan ini gratis dan dapat ditemukan di <http://maps.google.com> (Riyanto, 2010, hal. 9)

Google Maps menawarkan peta yang dapat digeser (*panned*), diperbesar (*zoom in*), diperkecil (*zoom out*), dapat diganti dalam beberapa mode (*map, satelit, hybrid*, dan lain-lain), fitur pencarian rute (*routing*), petunjuk arah dari suatu objek peta ke objek yang lain, (*direction*), dan juga pencarian tempat (*Place*) bisnis di Amerika, Kanada, Jepang, Hongkong, Cina, Inggris, Irlandia (hanaya pusat kota), dan beberapa bagian Eropa. Sampai saat ini, Google Maps masih berada dalam tahap beta, dan masih terus

dikembangkan dengan data yang selalu diperbarui secara berkala. (Riyanto, 2010, hal. 9)

2.7.1 Cara Kerja Google Maps

Google Maps dibuat dengan menggunakan kombinasi dari gambar peta, database, serta obyek-obyek interaktif yang dibuat dengan bahasa pemrograman *HTML*, *Javascript* dan *AJAX*, serta beberapa bahasa pemrograman lainnya. Gambar-gambar yang muncul pada peta merupakan hasil komunikasi dengan database pada *web server Google* untuk menampilkan gabungan dari potonganpotongan gambar yang diminta. Keseluruhan citra yang ada diintegrasikan ke dalam database pada *Google Server*, yang nantinya akan dapat dipanggil sesuai kebutuhan permintaan. Bagian- bagian gambar map merupakan gabungan dari potongan gambar-gambar bertipe *PNG* yang disebut *tile* yang berukuran 256 x 256pixel.

2.7.2 Google Maps API

API atau *Application Programming Interface* merupakan suatu dokumentasi yang terdiri dari *interface*, fungsi, kelas, struktur dan sebagainya untuk membangun sebuah perangkat lunak. Dengan adanya *API* ini, maka memudahkan programmer untuk “membongkar” suatu *software* untuk kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak yang lain. *API* dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya yang

memungkinkan programmer menggunakan sistem *function*. Proses ini dikelola melalui *Operating System*.

Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya dapat saling berhubungan dan berinteraksi. Bahasa pemrograman yang digunakan oleh *Google Maps* yang terdiri dari *HTML*, *Javascript* dan *AJAX* serta *XML*, memungkinkan untuk menampilkan peta *Google Maps* di *website* lain. *Google* juga menyediakan layanan *Google Maps API* yang memungkinkan para pengembang untuk mengintegrasikan *Google Maps* ke dalam *website* masing-masing dengan menambahkan data point sendiri. Dengan menggunakan *Google Maps API*, *Google Maps* dapat ditampilkan pada *web site eksternal*. Agar aplikasi *Google Maps* dapat muncul di *website* tertentu, diperlukan adanya *API Key*. *API Key* merupakan kode unik yang digenerasikan oleh *Google* untuk suatu *website* tertentu, agar server *Google Maps* dapat mengenali. (Siswanto. 2013)

Saat penelitian ini dibuat, *Google Maps* sudah rilis API versi 2 untuk jenis peta pada handheld dan smartphone, seperti Maps untuk Apple dan Android. Sedangkan untuk Maps versi web kini sudah mencapai API versi 3 dengan format data yang lebih lengkap dari sebelumnya.

Dalam penelitian ini, data *Google Maps* diperoleh dari server *Google APIS* ([http:// maps.googleapis.com](http://maps.googleapis.com)). Data yang diperoleh berupa format *JSON* dengan informasi-informasi yang bisa diolah

dengan aplikasi berbasis web, desktop, maupun *Smartphone*. Untuk mendapatkan data, maka client harus mengirimkan request ke Google APIS dengan format dan parameter tertentu. Format-format yang disediakan google adalah mode, alternatives, maupun tipe data yang diinginkan. Berikut ini adalah format standar informasi jalan dari google maps.

```
{ "routes" :
  [{ "bounds" :
    { "northeast" : { "lat" : -7.9468333, "lng" :
112.6607378}, "southwest" : { "lat" : -7.949940700000001,
"lng" : 112.6534805} },
    "copyrights" : "Map data ©2013 Google",
    "legs" : [{
      "distance" : { "text" : "1.0 km", "value" :
1043 },
      "duration" : { "text" : "2 mins", "value" :
129 },
      "end_address" : "Jalan Batubara, Blimbing,
Malang City, East Java 65126, Indonesia",
      "end_location" : { "lat" : -7.9482577, "lng"
: 112.6534805 },
      "start_address" : "Jalan Simpang Sulfat
Utara, Blimbing, Malang City, East Java 65126, Indonesia",
      "start_location" : { "lat" : -
7.949940700000001, "lng" : 112.6605367 },
      "steps" : [
        { "distance" : { "text" : "90 m", "value"
: 90 },
          "duration" : { "text" : "1 min", "value"
: 14 },
          "end_location" : { "lat" : -
7.949223699999999, "lng" : 112.6607378 },
          "html_instructions" : "Head
\u003cb\u003enorthwest\u003c/b\u003e",
          "polyline" : { "points" :
"bvoo@k_snT?BABA@A@C@C?EACAe@Qa@Ok@O" },
          "start_location" : { "lat" : -
7.949940700000001, "lng" : 112.6605367 },
          "travel_mode" : "DRIVING"},
        { "distance" : { "text" : "0.8
km", "value" : 778 },
```

```

        "duration" : {"text" : "2
mins","value" : 94 },
        "end_location" : {"lat" : -
7.9468333,"lng" : 112.6541389 },
        "html_instructions" : "Turn
\u003cb\u003eleft\u003c/b\u003e onto \u003cb\u003eJl.
Laksamana Adi Sucipto\u003c/b\u003e",
        "maneuver" : "turn-left",
        "polyline" : {"points" :
"rqoo@s`snTmI@AFSr@Ol@_@dBYjAMd@Ov@GTKn@ABGd@o@hEOLAMv@M^Qd@
Wh@?@s@bBEFUL@GLMXa@bAe@fa" },
        "start_location" : {"lat" : -
7.949223699999999,"lng" : 112.6607378 },
        "travel_mode" : "DRIVING"},
        { "distance" : {"text" : "0.2
km","value" : 175 },
        "duration" : {"text" : "1 min","value"
: 21 },
        "end_location" : {"lat" : -
7.9482577,"lng" : 112.6534805 },
        "html_instructions" : "Turn
\u003cb\u003eleft\u003c/b\u003e onto \u003cb\u003eJl.
Batubara\u003c/b\u003e\u003cdiv style=\"font-
size:0.9em\"\u003eDestination will be on the
right\u003c/div\u003e",
        "maneuver" : "turn-left",
        "polyline" : {"points" :
"tboo@kwqnTnAd@ZHnA\\l@TD@^RLH" },
        "start_location" : {"lat" : -
7.9468333,"lng" : 112.6541389 },
        "travel_mode" : "DRIVING"} ],
        "via_waypoint" : [[]] ],
        "overview_polyline" : {"points" :
"bvoo@k_snTEJQ?i@SmA_@s@vCwAnG]nB_AvGMv@M^i@nAs@dBq@|AgAjCjB
n@|Br@d@TLH" },
        "summary" : "Jl. Laksamana Adi Sucipto",
        "warnings" : [],
        "waypoint_order" : []}
    ],
    "status" : "OK"}

{ "routes" :
  [{ "bounds" : {"northeast" :
    { "lat" : -7.9468333, "lng" :
112.6607378},"southwest" : { "lat" : -7.949940700000001,
"lng" : 112.6534805} },
    "copyrights" : "Map data \u00a92013 Google",
    "legs" : [{
      "distance" : {"text" : "1.0 km","value" :
1043 },

```

```

        "duration" : {"text" : "13 mins","value" :
790 },
        "end_address" : "Jalan Batubara, Blimbing,
Malang City, East Java 65126, Indonesia",
        "end_location" : {"lat" : -7.9482577,"lng"
: 112.6534805 },
        "start_address" : "Jalan Simpang Sulfat
Utara, Blimbing, Malang City, East Java 65126, Indonesia",
        "start_location" : {"lat" : -
7.949940700000001,"lng" : 112.6605367 },
        "steps" : [
            { "distance" : {"text" : "90
m","value" : 90 },
              "duration" : {"text" : "1 min","value"
: 75 },
              "end_location" : {"lat" : -
7.9492236999999999,"lng" : 112.6607378 },
              "html_instructions" : "Head
\u003cb\u003enorthwest\u003c/b\u003e",
              "polyline" : {"points" :
"bvoo@k_snT?BABA@A@C@C?C?EACAe@Qa@Ok@O" },
              "start_location" : {"lat" : -
7.949940700000001,"lng" : 112.6605367 },
              "travel_mode" : "WALKING"},
            { "distance" : {"text" : "0.8
km","value" : 778 },
              "duration" : {"text" : "10
mins","value" : 584 },
              "end_location" : {"lat" : -
7.9468333,"lng" : 112.6541389 },
              "html_instructions" : "Turn
\u003cb\u003eleft\u003c/b\u003e onto \u003cb\u003eJl.
Laksamana Adi Sucipto\u003c/b\u003e",
              "maneuver" : "turn-left",
              "polyline" : {"points" :
"rqoo@s`snTm1@AFSr@Ol@_@dBjAMd@Ov@GTKn@ABGd@o@hEOLAMv@M^Qd@
Wh@?@s@bBEFUL@GLMXa@bAe@fA" },
              "start_location" : {"lat" : -
7.9492236999999999,"lng" : 112.6607378 },
              "travel_mode" : "WALKING"},
            { "distance" : {"text" : "0.2
km","value" : 175 },
              "duration" : {"text" : "2
mins","value" : 131 },
              "end_location" : {"lat" : -
7.9482577,"lng" : 112.6534805 },
              "html_instructions" : "Turn
\u003cb\u003eleft\u003c/b\u003e onto \u003cb\u003eJl.
Batubara\u003c/b\u003e\u003cdiv style=\\"font-

```

```

size:0.9em\"\\u003eDestination will be on the
right\\u003c/div\\u003e",
      "maneuver" : "turn-left",
      "polyline" : {"points" :
"tboo@kwqnTnAd@ZHnA\\l@TD@^RLH" },
      "start_location" : {"lat" : -
7.9468333,"lng" : 112.6541389 },
      "travel_mode" : "WALKING"} ],
      "via_waypoint" : []} ],
      "overview_polyline" : {"points" :
"bvoo@k_snTEJQ?i@SmA_@s@vCwAnG]nB_AvGMv@M^i@nAs@dBq@|AgAjCjB
n@|Br@d@TLH" },
      "summary" : "Jl. Laksamana Adi Sucipto",
      "warnings" : ["Walking directions are in beta.Use
caution - This route may be missing sidewalks or pedestrian
paths." ],
      "waypoint_order" : []
    } ],
    "status" : "OK"}

```

Keterangan dari informasi diatas adalah sebagai berikut :

1. *Northeast dan Southwest* : Merupakan lokasi tampilan aktif pada Google Map yang berkaitan dengan batas frame lokasi pada tampilan;
2. *Copyright* : Hak cipta kepemilikan Google;
3. *Legs* : Jalur yang memungkinkan dari suatu titik ke titik lain pada peta. Legs memiliki beberapa jalur sesuai dengan jumlah jalur atau jalan yang mungkin dilalui;
4. *Distance* : Jarak pada jalan atau antara dua titik jalan;
5. *Duration* : Durasi atau lama waktu yang ditempuh dari satu titik ke titik yang lain atau lama waktu yang ditempuh pada suatu jalan;
6. *End_location* : Titik akhir jalan pada peta;
7. *Html_instruction* : Instruksi ketika beradari di titik akhir atau belokan pada jalan sesuai dengan nama jalan yang dilalui;
8. *Maneuver* : Instruksi ketika pada belokan untuk belok kiri atau kanan;

9. *Polyline* : Merupakan jenis data atau gambar berupa garis, point, atau luasan;
10. *Start_location* : Titik awal pada peta atau jalan;
11. *Travel_mode* : Jenis perjalanan yang bias dilalui;
12. *Overview_polyline* : Penjelasan tentang polyline yang tampak pada peta, termasuk titik, garis, atau luasan;
13. *Summary* : Informasi nama jalan yang dilalui;
14. *Warning* : Keterangan untuk berhati-hati dari Google, bisa karena jenis versi peta yang kurang sempurna, pencarian yang kurang tepat, atau perubahan pada kondisi riil di jalan.
15. *Waypoint_order* : Urutan penggunaan jalan dari *start_location* hingga *end_location*;
16. *Status* : informasi pengiriman data sukses atau tidak;

Untuk mengolah data tersebut dalam algoritma Dijkstra, peneliti hanya membutuhkan titik awal (*start_location*) dan titik akhir (*end_location*) sebagai poin atau node, *polyline* sebagai Edge-nya, dan *distance* sebagai bobot (panjang jalan) atau jarak dari titik awal jalan ke titik akhir jalan.

2.8 MySQL

MySQL adalah Sebuah program *database server* yang mampu menerima dan mengirimkan datanya sangat cepat, *multi user* serta

menggunakan perintah dasar SQL (*Structured Query Language*). (Haris Saputro, 2012)

MySQL merupakan dua bentuk lisensi, yaitu *Free Software* dan *Shareware*. MySQL yang biasa kita gunakan adalah *MySQL Free Software* yang berada dibawah Lisensi GNU/GPL (*General Public License*).

MySQL Merupakan sebuah *database server* yang *free*, artinya kita bebas menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya. MySQL pertama kali dirintis oleh seorang *programmer database* bernama Michael Widenius . Selain *database server*, MySQL juga merupakan program yang dapat mengakses suatu *database* MySQL yang berposisi sebagai *Server*, yang berarti program kita berposisi sebagai *Client*. Jadi MySQL adalah sebuah *database* yang dapat digunakan sebagai *Client* maupun *server*. Database MySQL merupakan suatu perangkat lunak *database* yang berbentuk *database relational* atau disebut *Relational Database Management System* (*RDBMS*) yang menggunakan suatu bahasa permintaan yang bernama SQL (*Structured Query Language*).

2.8.1 Mengetahui SQL (*Structured Query Language*)

SQL (*Structured Query Language*) adalah sebuah bahasa permintaan database yang terstruktur. Bahasa SQL ini dibuat sebagai bahasa yang dapat merelasikan beberapa tabel dalam *database* maupun merelasikan antar *database*.

SQL dibagi menjadi tiga bentuk Query, yaitu :

2.8.1.1 DDL (*Data Definition Language*)

DDL adalah sebuah metode *Query SQL* yang berguna untuk mendefinisikan data pada sebuah *Database*, *Query* yang dimiliki DDL adalah :

1. *CREATE* : Digunakan untuk membuat *Database* dan Tabel
2. *Drop* : Digunakan untuk menghapus Tabel dan *Database*
3. *Alter* : Digunakan untuk melakukan perubahan struktur tabel yang telah dibuat, baik menambah *Field* (*Add*), mengganti nama *Field* (*Change*) ataupun menamakannya kembali (*Rename*), dan menghapus *Field* (*Drop*).

2.8.1.2 DML (*Data Manipulation Language*)

DML adalah sebuah metode *Query* yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari *Query* DML ini untuk melakukan pemanipulasian *database* yang telah dibuat.

Query yang dimiliki DML adalah :

1. *INSERT* : Digunakan untuk memasukkan data pada Tabel *Database*
2. *UPDATE* : Digunakan untuk pengubahan terhadap data yang ada pada tabel *Database*
3. *DELETE* : Digunakan untuk Penhapusan data pada tabel *Database*.

2.8.1.3 DCL (*Data Control Language*)

DCL adalah sebuah metode *Query SQL* yang digunakan untuk memberikan hak otorisasi mengakses *Database*, mengalokasikan *space*, pendefinisian *space*, dan pengauditan penggunaan *database*. *Query* yang dimiliki DCL adalah :

1. *GRANT* : Untuk mengizinkan *user* mengakses tabel dalam *Database*.
2. *REVOKE* : Untuk membatalkan izin hak *user*, yang ditetapkan oleh perintah *GRANT*.
3. *COMMIT* : Menetapkan penyimpanan *Database*.
4. *ROLLBACK* : Membatalkan penyimpanan *Database*. (Haris Saputra. 2012)

BAB III

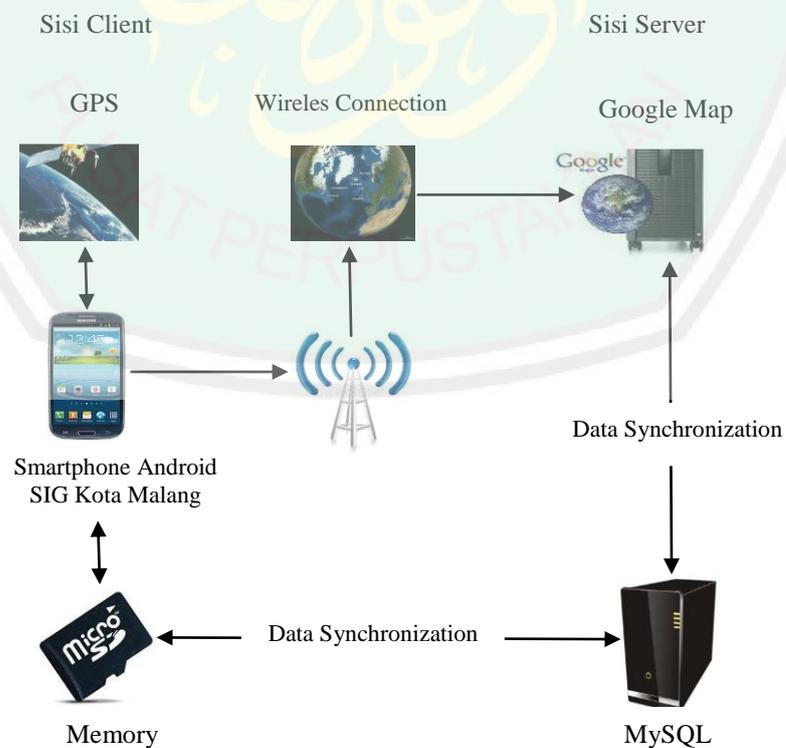
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisa Sistem

Pada proyek akhir mengenai “Sistem Informasi Geografis Untuk Pencarian Lokasi Lahan Pertanian Berbasis Android” diarahkan pada dua sisi yaitu sebagai berikut :

1. SIG pertanian *provider*, yaitu penyedia layanan yang berbasis *web server* (http).
2. SIG pertanian *client*, yaitu aplikasi berbasis sistem operasi Android yang mengakses layanan dari *provider*.

Desain umum yang digambarkan, adalah sebagai berikut:



Gambar 3.1 Desain Sistem Secara Umum

Arsitektur desain sistem aplikasi disesuaikan dengan arsitektur aplikasi *server*, yaitu *Google Maps* yang menyediakan peta secara *online* dan MySQL yang menyediakan data lahan Pertanian Kota Malang dengan bantuan PHP sebagai pemroses inputan dari pengguna. Dari arsitektur terpusat (*client-server*) tersebut, komponen yang terdapat pada sisi *client* berupa pengguna akhir perangkat bergerak *Android* yang dapat menampilkan peta atau memberikan hasil analisa operasi SIG seperti pencarian Lahan Pertanian. Sedangkan, komponen pada sisi *server* menyediakan data geospasial dan melakukan operasi SIG berdasarkan permintaan dari sisi *client*.

Terdapat berbagai jenis jaringan komunikasi nirkabel (GPRS/3G, CDMA/EVDO, ISP) diantara sisi *client* dan sisi *server* yang berfungsi untuk memfasilitasi pertukaran geodata dan layanan. Berdasarkan Gambar 3.2 menunjukan desain sistem secara umum dan menggambarkan 7 komponen dasarnya :

1. GPS/LBS/Wifi

Sistem penentuan posisi yang dapat memberikan informasi georeferensi koordinat (x, y, dan zelevasi) untuk *Smartphone Android*. Terdapat 2 sistem utama, yaitu sistem penentuan posisi lokal (LBS/Wifi) yang mengandalkan sistem posisi triangulasi sinyal radio atau sinyal telepon seluler dari beberapa BTS dan global (GPS) yang menggunakan sinyal satelit untuk menghitung posisi unit GPS.

2. Smartphone Android

Handphone atau Tablet berplatform *Android* yang telah dilengkapi dengan perangkat komunikasi nirkabel yang dapat digunakan untuk pertukaran data dan telekomunikasi ke *server*. Smartphone *Android* menjalankan aplikasi SIG lahan Pertanian Kota Malang yang dapat melakukan proses *geocoding*, pencarian *route*, atau menampilkan peta beserta informasi lokasi dari lahan Pertanian kepada pengguna akhir.

3. Memory

Tempat penyimpanan sementara data *cache geospasial* seperti *tiles* dan basis data Lahan pertanian. Data akan terisi setelah dilakukannya pengunduhan dan sinkronisasi dari *server*.

4. Wireless Communication

Mekanisme komunikasi yang menghubungkan *Smartphone Android* dengan *server*. Hubungan ini dapat berupa komunikasi nirkabel secara seperti melalui *Wifi* atau sinyal telepon selular. Komunikasi ini merupakan komunikasi dua arah dimana saat *Smartphone Android* meminta informasi dari *server*, dan *server* akan merespon permintaan dengan mengirimkan informasi terbaru ke *client* atau penerima.

5. Google Maps

Google Maps adalah layanan aplikasi peta *online* yang disediakan oleh *Google* secara gratis. Layanan peta *Google Maps* secara resmi dapat diakses melalui situs <http://maps.google.com>.

6. MySQL

MySQL merupakan suatu perangkat lunak *database* yang berbentuk database relasional atau disebut *Relational Database Management System* (RDBMS) yang menggunakan bahasa SQL (*Structured Query Language*).

3.2 Analisa Kebutuhan

Sistem yang akan dibuat dapat memberikan penyelesaian masalah berupa daftar lokasi lahan pertanian dan peta penunjuk jalan dengan jarak tempuh terpendek. Dalam membangun sistem ini, parameter-parameter yang dibutuhkan untuk data masukan sistem meliputi:

1. Data kategori wilayah lahan pertanian yang terpilih, hasil masukan (*input*) pengguna.
2. Data posisi lahan pertanian hasil masukan (*input*) pengguna, yaitu data lokasi lahan pertanian atau lokasi akhir (*target*) terkait lahan pertanian tersebut.
3. Data posisi pengguna saat ini berada, hasil masukan dari perangkat GPS pada *smartphone*.

Setelah pengguna memilih kategori wilayah lahan pertanian maka akan ditampilkan daftar lahan pertanian yang dimaksud. Pengguna dapat memilih satu diantara sekian daftar tersebut. Kemudian akan ditampilkan lahan pertanian tersebut serta penunjuk jalan. Penunjuk jalan yang dimaksud akan menunjukkan jalur tempuh terpendek dari tempat pengguna berada

menuju lokasi lahan pertanian. Untuk mendapatkan jalur terpendek akan digunakan algoritma pencarian jalur terpendek dijkstra. Penunjuk jalan ini akan ditampilkan dalam bentuk peta. Peta yang digunakan dalam sistem ini adalah google map.

Sedangkan komponen yang dibutuhkan terbagi menjadi dua macam, yaitu komponen *software* dan *hardware*.

3.2.1 Software

Software yang dibutuhkan untuk proses pembuatan aplikasi SIG ini antara lain sebagai berikut:

a. *Windows XP*

Windows XP digunakan karena lebih banyak *support* dan *compatible* dengan *software* lain yang dibutuhkan dalam pembuatan aplikasi SIG ini.

b. *Java Development Kit (JDK)*

JDK merupakan paket *platform Java* yang terdiri dari berbagai macam *library*, *JVM*, *compiler* dan *debugger*.

c. *Java Runtime Environment (JRE)*

Supaya sebuah program *Java* dapat dijalankan, maka file berekstensi *.java* harus dikompilasi menjadi file *bytecode*. *JRE* berfungsi untuk mengeksekusi *file bytecode* yang memungkinkan pemakai untuk menjalankan program *Java* di berbagai *platform*.

d. *Android SDK*

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform Android* menggunakan bahasa pemrograman *Java*.

e. *Eclipse*

Eclipse adalah sebuah *IDE (Integrated Development Environment)* untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform (platform-independent)*.

3.2.2 *Hardware*

Dalam pembuatan aplikasi *mobile learning*, *hardware* yang dibutuhkan antara lain:

a. *Komputer*

Komputer yang digunakan untuk membangun aplikasi *mobile learning* mempunyai spesifikasi sebagai berikut:

- 1) AMD Dual-Core A4-3305M
- 2) RAM 2 GB
- 3) Hardisk 500GB
- 4) VGA AMD Radeon HD 6480G

b. *Smartphone*

Selain menggunakan emulator android yang diintegrasikan dengan *Eclipse*, peneliti juga menguji aplikasi

mobile learning ini pada *smartphone Android Samsung Galaxy GIO GT-S5660* dengan spesifikasi sebagai berikut:

- 1) Prosesor Adreno 200, 800 MHz
- 2) Cipset Qualcomm QCT MSM7227-1 Turbo
- 3) Memori internal 158 MB storage
- 4) RAM 278 MB
- 5) Micro SD 2 GB
- 6) Ukuran Layar 3.2 inches (~180 ppi pixel density) 320 x 480 pixels.
- 7) Body 110.5 x 57.5 x 12.2 mm (4.35 x 2.26 x 0.48 in)

3.3 Spesifikasi Aplikasi

Aplikasi terbentuk dari dua perangkat utama, yaitu *client* dan *server*. *Server* berfungsi untuk membangun, menyimpan, dan mengolah peta dan data-data atributnya. Sisi *server* memiliki spesifikasi sebagai berikut:

1. Melakukan digitasi peta dan mengolah hasil digitasi peta.
2. Melakukan pengelolaan data atribut Lahan Pertanian

Spesifikasi pada sisi *client* adalah sebagai berikut:

1. Aplikasi yang dibuat pada *Smartphone Android* menggunakan JDK, *Eclipse Indigo* dan *Android SDK*
2. Menggunakan *Teknologi Java*.

Sedangkan sistem aplikasi pencarian lokasi lahan pertanian dengan menggunakan algoritma *dijkstra* akan memiliki kemampuan sebagai berikut:

1. Memberikan informasi lokasi lahan pertanian yang akan dituju di daerah kota malang berdasarkan beberapa kriteria yang dipilih oleh pengguna.
2. Memberikan informasi penunjuk jalan dengan jarak tempuh terpendek dari tempat pengguna sekarang berada ke lokasi lahan Pertanian yang dituju pengguna.

3.4 Tahap Pengerjaan

Berikut ini adalah langkah- langkah untuk membuat aplikasi Sistem Informasi Geografis Untuk Pencarian Lokasi Lahan Pertanian:



Gambar 3.2 Diagram Alir Pembuatan sistem

3.4.1 Pengumpulan Data

Informasi lahan Pertanian diperoleh dari Dinas Pertanian. Data-data yang dibutuhkan adalah berupa data spasial atau peta wilayah dan data lokasi lahan Pertanian. Adapun data – data tersebut meliputi:

1. Peta Kota Malang
2. Data lahan Pertanian Kota Malang

3.5 Perancangan Sistem

Dalam perancangan dan pembuatan sistem terdapat beberapa fungsi sistem utama, antara lain:

1. Menu, berisi 4 pilihan menu yaitu:
 - a. Tampilan Peta, dimana akan ditampilkan 4 pilihan tampilan peta yaitu normal, satellite, hybrid, terrain.
 - b. Penambahan Lahan, admin dapat menambah point atau titik lahan pada peta beserta data lahan tersebut.
 - c. Sembunyikan lahan, menyembunyikan point atau titik lahan pada peta.
 - d. Cari Jalur, pengguna atau user bisa melakukan pencarian rute terpendek dari tempatnya berada ke beberapa lahan dengan komoditas yang sama.
2. Kategori wilayah, berisi 5 kecamatan di Kota Malang yaitu Kecamatan Sukun, Kecamatan Klojen, Kecamatan Blimbing, Kecamatan Lowokwaru, Kecamatan Kedung Kandang. Ketika pengguna memilih

salah satu kecamatan maka peta akan menampilkan kecamatan yang dipilih.

3. Peta, yang di dalamnya terdapat:
 - a. Lokasi, digunakan untuk mengetahui posisi dari lahan yang tersedia.
 - b. Zoom, digunakan untuk memperbesar atau mengecilkan tampilan dari peta.

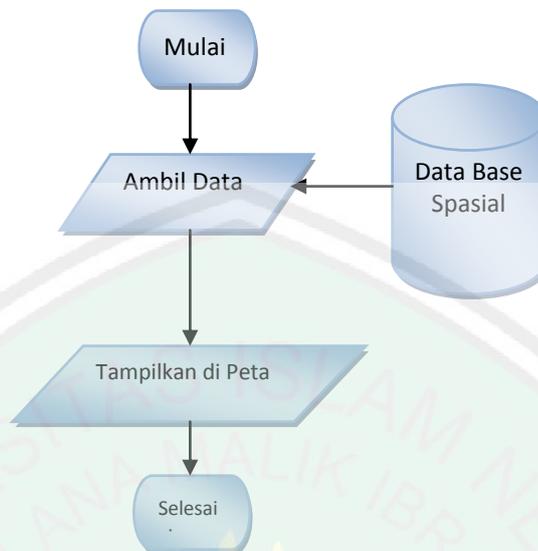
3.6 Flow Chart

Aplikasi ini dimulai dari start, kemudian akan menampilkan *splash screen*. *Splash screen* secara otomatis akan di direct ke Tampilan Utama setelah beberapa detik. Menu utama ini terdiri dari beberapa komponen, yaitu Menu Tampilan Peta, Menu Penambahan lahan, Menu Sembunyikan Lahan, Menu Cari Jalur, Menu Kategori Wilayah, Menu Peta.

3.6.1 Flow Chart Menu Tampilan Peta

Untuk menampilkan tampilan peta yang akan dipilih oleh pengguna. Pada tampilan peta ini sudah disediakan oleh *Map Server* yang kali ini peneliti mengambil data dari Google Maps.

Diagram flow chart dari menu tampilan ini bisa dilihat pada Gambar 3.3. Apabila digambarkan dalam bentuk flow chart adalah sebagai berikut:

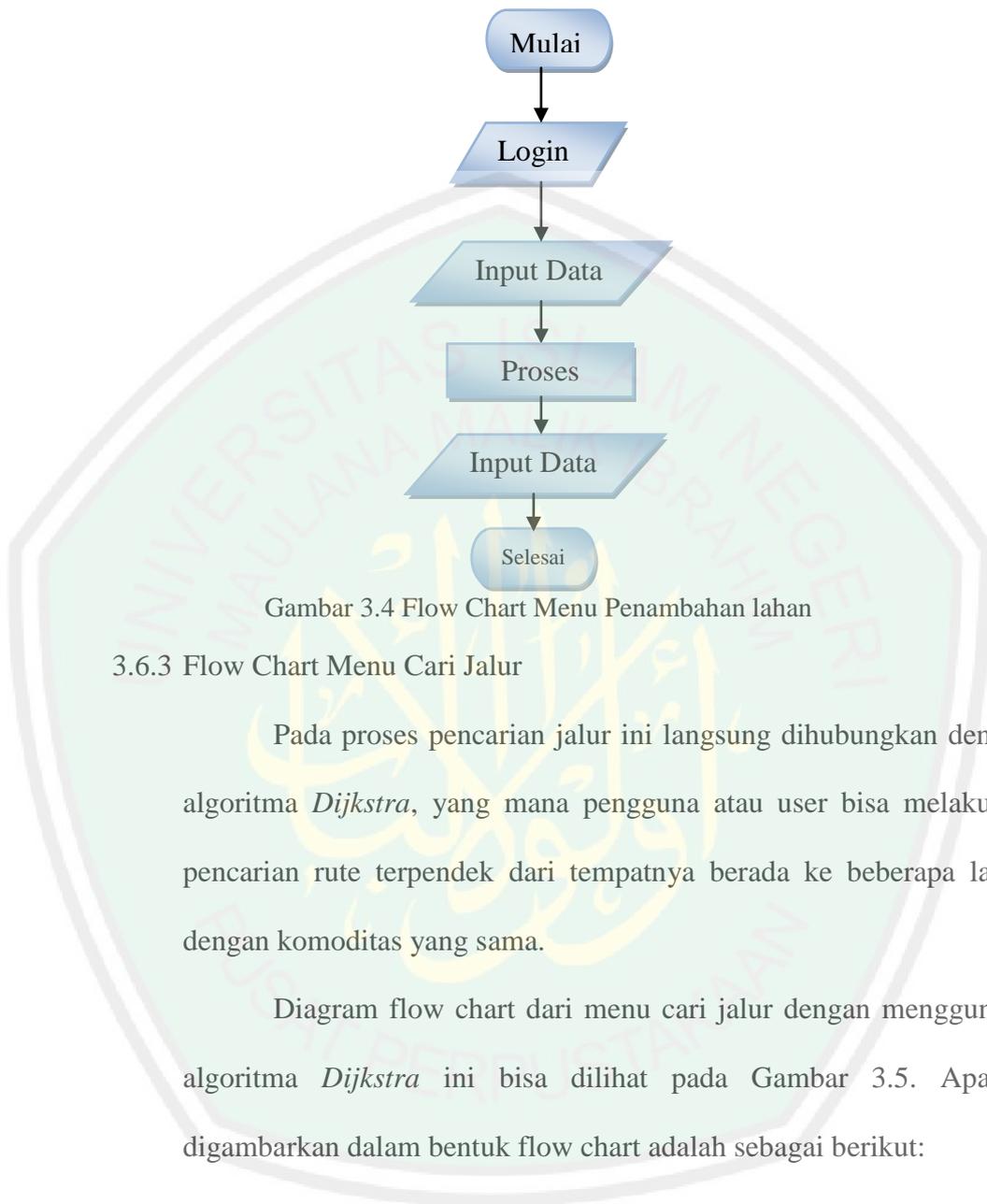


Gambar 3.3 Flow Chart Menu Tampilan Peta

3.6.2 Flow Chart Menu Penambahan lahan

Pada proses penambahan lahan hanya dapat di tambah oleh admin oleh sebab itu admin harus melalui proses login terlebih dahulu pada proses ini seorang admin dapat menambah point atau titik lahan pada peta beserta data lahan tersebut.

Diagram flow chart dari menu penamhan lahan ini bisa dilihat pada Gambar 3.4. Apabila digambarkan dalam bentuk flow chart adalah sebagai berikut:

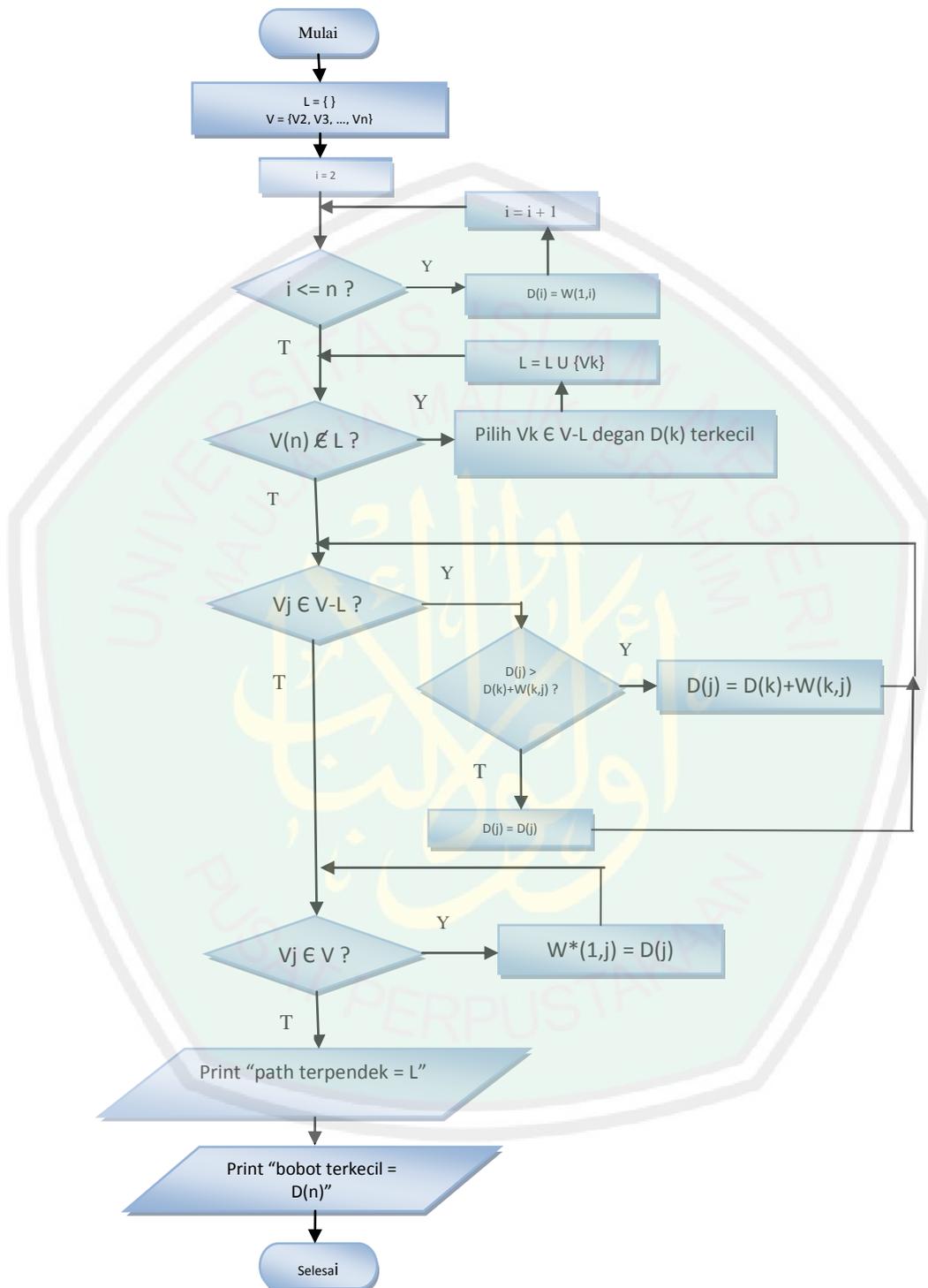


Gambar 3.4 Flow Chart Menu Penambahan lahan

3.6.3 Flow Chart Menu Cari Jalur

Pada proses pencarian jalur ini langsung dihubungkan dengan algoritma *Dijkstra*, yang mana pengguna atau user bisa melakukan pencarian rute terpendek dari tempatnya berada ke beberapa lahan dengan komoditas yang sama.

Diagram flow chart dari menu cari jalur dengan menggunakan algoritma *Dijkstra* ini bisa dilihat pada Gambar 3.5. Apabila digambarkan dalam bentuk flow chart adalah sebagai berikut:



Gambar 3.5 Flow Chart Menu Cari Lahan Menggunakan Algoritma *Dijkstra*

Berdasarkan terminologi teori graph, maka suatu jaringan akan terdiri dari suatu himpunan titik-titik yang disebut node. Node-node tersebut saling dihubungkan oleh suatu garis dan disebut edge. Untuk

setiap dua node dapat terjadi beberapa lintasan, dimana lintasan dengan bobot yang minimum disebut sebagai lintasan atau rute terpendek. Bobot di sini dapat berupa jarak, waktu tempuh, atau ongkos transportasi dari satu node ke node yang lainnya yang membentuk rute tertentu. Algoritma mencari rute terpendek ini dikembangkan oleh Dijkstra (Jong Jek Siang: 2002).

Misalkan:

$$V(G) = \{V_1, V_2, \dots, V_n\}$$

L = Himpunan titik-titik $\in V(G)$ yang sudah terpilih dalam alur path terpendek.

$D(j)$ = Jumlah bobot path terkecil dari V_1 ke V_j

$W(i,j)$ = Bobot garis dari titik V_i ke titik V_j

$W^*(1,j)$ = Jumlah bobot terkecil dari V_1 ke V_j

Maka, algoritma Dijkstra untuk mencari path terpendek adalah sebagai berikut :

1. $L = \{ \}$;

$$V = \{V_2, V_3, \dots, V_n\}$$

2. Untuk $i = 2, \dots, n$, lakukan $D(i) = W(1,i)$

3. Selama $V_n \in L$ lakukan :

a. Pilih titik $V_k \in V-L$ dengan $D(k)$ terkecil $L = L \cup \{V_k\}$

b. Untuk setiap $V_j \in V-L$ lakukan : Jika $D(j) > D(k) + W(k,j)$

maka ganti $D(j)$ dengan $D(k) + W(k,j)$

4. Untuk setiap $V_j \in V$, $W^*(1,j) = D(j)$

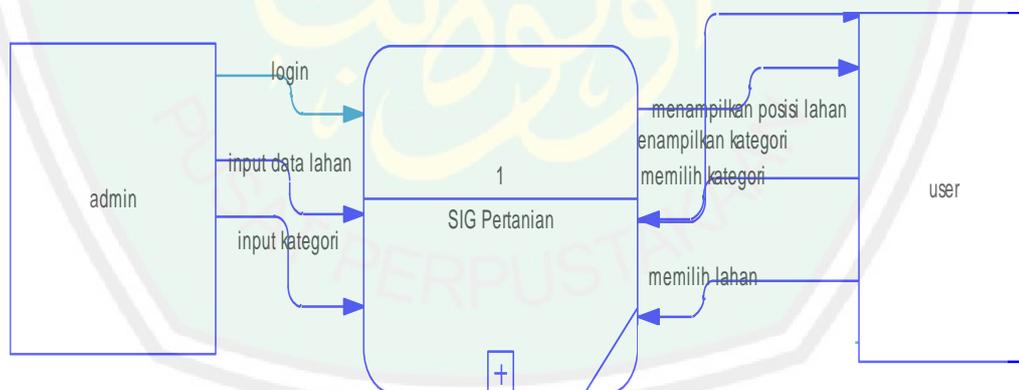
Menurut algoritma diatas, path terpendek dari titik V_1 ke V_n adalah melalui titik-titik dalam L secara berurutan dan jumlah bobot path terkecilnya adalah $D(n)$.

3.7 DFD (Data Flow Diagram)

Data Flow Diagram (DFD) adalah representasi grafik dari sebuah sistem. DFD menggambarkan komponen-komponen sebuah sistem, aliran-aliran data di mana komponen-komponen tersebut, asal, tujuan, dan penyimpanan dari data tersebut.

3.7.1 DFD (Data Flow Diagram) Level 0

Data flow diagram pada gambar dibawah menampilkan proses-proses utama yang terjadi pada sistem aplikasi.



Gambar 3.6 DFD level 0

Penjelasan DFD level 0 aplikasi pencarian lokasi lahan pertanian di kota malang adalah sebagai berikut:

1. Admin

Admin bertugas untuk memasukkan data-data yang dibutuhkan kedalam database sistem. Data-data dapat dikelompokkan menjadi tiga pokok bahasan yaitu data peta, data lokasi lahan pertanian, dan data kategori.

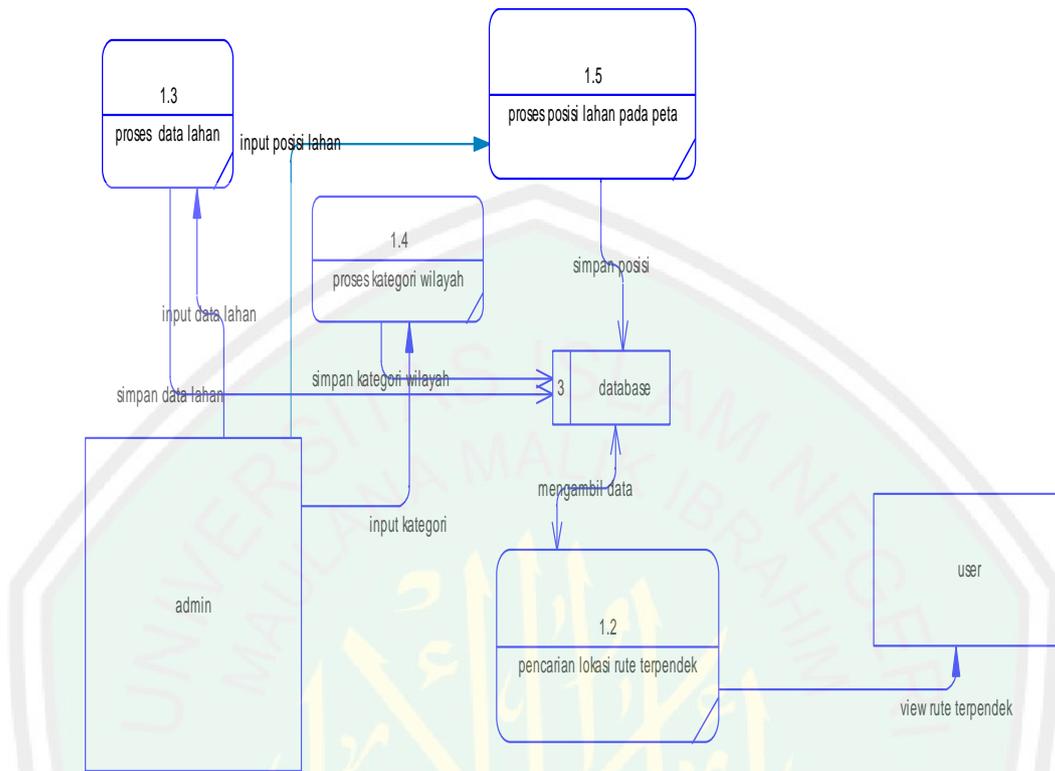
2. User

User dapat memilih kriteria lokasi lahan pertanian yang ingin dikunjungi. Kemudian memilih lokasi lahan pertanian dari daftar lokasi lahan. User harus mengaktifkan fitur GPS pada smartphone yang digunakan sehingga secara otomatis dapat memasukkan koordinat lokasi user saat ini berada.

3.7.2 DFD (Data Flow Diagram) Level 1

Data flow diagram level 1 proses entry data pada gambar

3.7 memuat proses penyimpanan data.



Gambar 3.7 DFD level 1

DFD level 1 diatas menjelaskan proses penyimpanan data dan proses pengolahan data untuk masing-masing proses utama yang dijelaskan dalam DFD level 0. Admin bertugas memasukkan data kategori dan data lokasi lahan kedalam database. Kemudian diproses pada pengolahan data untuk menghasilkan informasi yang ditampilkan pada user. Informasi yang diperoleh user yaitu informasi tentang kategori, daftar lokasi lahan pertanian serta informasi rute terpendek menuju lokasi lahan.

3.8 Struktur Basis Data

Basis data adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi. (Kadir, 1999, hal. 254)

Aplikasi pencarian lokasi lahan pertanian ini menggunakan MySQL sebagai manajemen databasenya. Di bawah ini akan diberikan struktur tabel-tabel yang digunakan dalam aplikasi ini:

1. Tabel Login

Tabel Login adalah tabel yang digunakan untuk login ke aplikasi.

Tabel 3.1 Login

No	Nama Field	Type	Size
1	Id	Int	3
2	Username	Varchar	50
3	Password	Varchar	10

2. Tabel Kategori wilayah

Tabel kategori wilayah adalah tabel yang digunakan untuk menyimpan data wilayah.

Tabel 3.2 Kategori wilayah

No	Nama Field	Type	Size
1	Id	Int	3
2	Nama kecamatan	Varchar	50

3. Tabel Tipe Lahan

Tabel tipe lahan adalah tabel yang berisi tipe lahan.

Tabel 3.3 Tipe Lahan

No	Nama Field	Type	Size
1	Id	Int	3
2	Type	Varchar	50

4. Tabel Lokasi Lahan Pertanian

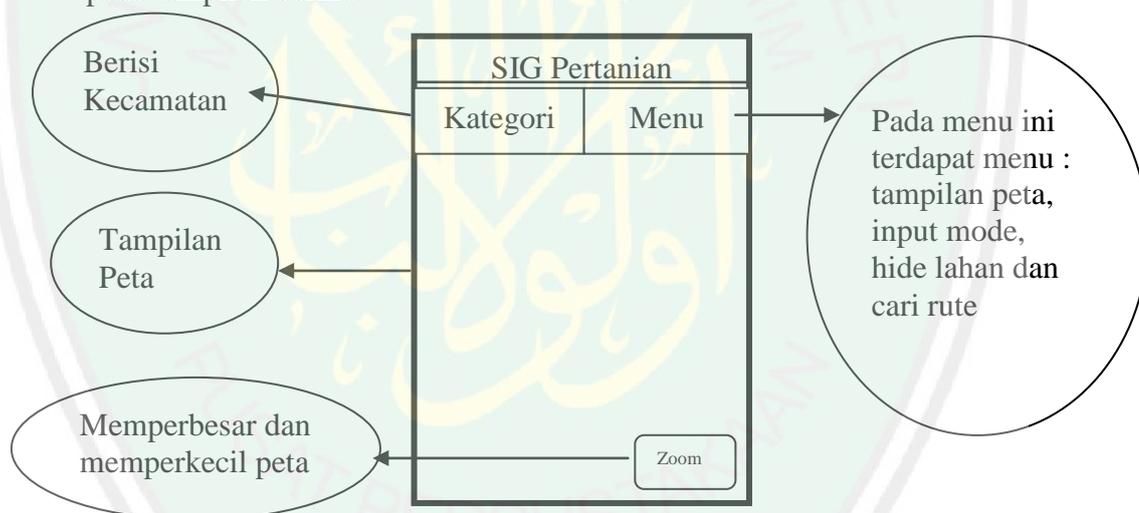
Tabel lokasi lahan pertanian adalah tabel yang digunakan untuk menyimpan data lokasi lahan yang ada di daerah kota malang.

Tabel 3.4 Lokasi Lahan Pertanian

Id	Nama Field	Type	Size
1	Id	Int	3
2	Nama pemilik	Varchar	300
3	Type	Varchar	1000
4	Alamat	Varchar	30
5	Longitude	Varchar	15
6	Latitude	Varchar	15
7	Deskripsi	Varchar	300

3.9 Desain *Interface* SIG Pertanian

Hal yang perlu diperhatikan dalam mendesain sebuah sistem adalah rancangan tersebut harus dapat memudahkan pengguna dalam menggunakan sistem aplikasi yang dibuat. Sehingga perlu diperhatikan dalam mengatur letak *button*, *textfield*, menu, ataupun komponen visual yang lain sehingga tidak membingungkan pengguna dalam pemakaian. Sehingga desain menu tampilan antar muka pada Sistem Informasi Geografis untuk Pencarian Lahan Pertanian Berbasis Android digambarkan pada tampilan berikut:



Gambar 3.8 Desain *Interface* SIG Pertanian

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Pada bab ini akan dibahas mengenai hasil uji coba terhadap sistem aplikasi pencarian lokasi lahan pertanian yang telah dibuat. Uji coba ini bertujuan untuk mengetahui apakah sistem aplikasi yang dibuat telah dapat berjalan sebagaimana mestinya sesuai dengan rancangan sistem pada BAB 3. Pada bab ini juga akan dibahas mengenai fitur dan *interface* yang terdapat di dalam sistem aplikasi beserta pembahasan mengenai algoritma dijkstra dalam menentukan rute terpendek.

4.1.1 Implementasi SIG Pertanian *Provider*

Desain SIG Pertanian pada sisi *provider* diimplementasikan dengan database MySQL dan *Google Maps* sebagai penyedia peta secara online sedangkan PHP sebagai pemroses untuk menerima dan mengirim data yang diminta oleh user. Dalam hal ini, *file PHP* tersebut merupakan penjemputan antara database MySQL dengan *request* dari aplikasi SIG Pertanian *client*. Sedangkan *Apache web server* berfungsi sebagai jembatan komunikasi dengan protokol HTTP.

Sistem pemroses dalam *Apache web server* dibedakan menjadi dua folder, yaitu folder pertama berisi file-file html dan php yang berfungsi untuk menampilkan website. Sedangkan folder kedua

berisi *file-file* php yang berfungsi untuk menangkap *request* dan memberikan *response* kepada aplikasi SIG Pertanian dengan format JSON.

4.1.2 Implementasi SIG Pertanian *Client*

Pada SIG Pertanian *client* terdapat dua sub sistem, sub sistem administrasi dan sub sistem pengguna. Sub sistem administrasi dilengkapi dengan fitur pengelolaan data. Sedangkan sub sistem pengguna, menampilkan data yang telah diinputkan oleh administrator. Pengguna bebas memilih lokasi lahan untuk dicari jalur terpendek dari lokasi pengguna berada menuju lokasi lahan yang dipilih. Pemrosesan pencarian jalur terpendek menggunakan algoritma *dijkstra* disediakan pada sisi *client*. Dimulai dengan pengambilan rute jalan dan selanjutnya diproses menggunakan algoritma *dijkstra* sehingga pemrosesan rute lebih cepat.

Uji *coba* perangkat *client* menggunakan *smartphone* Samsung Galaxy GIO GT-S5660 yang bersistem operasi android dengan sistem operasi Android versi 2.3.4. Sedangkan koneksi antara *client* dan *server* menggunakan internet GPRS.

4.1.3 Ruang Lingkup Perangkat Keras

Ruang lingkup perangkat keras yang diperlukan oleh sistem aplikasi SIG Pertanian ini antara lain:

1. AMD Dual-Core A4-3305M
2. RAM 2 GB

3. Hardisk 500GB
4. VGA AMD Radeon HD 6480G
5. Monitor 14"
6. *Keyboard*
7. *Mouse PS2*
8. *Smartphone Android*

4.1.4 Ruang Lingkup Perangkat Lunak

4.1.4.1 Ruang Lingkup Perangkat Lunak SIG Pertanian *Provider*

Ruang lingkup perangkat lunak yang diperlukan oleh sistem SIG Pertanian *provider* antara lain:

1. Google Maps
2. MySQL
3. FileZilla FTP Client

4.1.4.2 Ruang Lingkup Perangkat Lunak SIG Pertanian *Client*

Ruang lingkup perangkat lunak yang diperlukan oleh sistem SIG Pertanian *client* antara lain:

1. *Windows 7*
2. *Java Development Kit (JDK)*
3. *Java Runtime Environment (JRE)*
4. *Eclipse*
5. *Android SDK*
6. ADT

4.2 Implementasi *Interface* SIG Pertanian

Pada aplikasi *SIG Pertanian client* ini terdapat 4 *interface*, yaitu:

4.2.1 Halaman Splash

Saat pengguna membuka aplikasi maka yang tampil pertama kali adalah halaman splash. Halaman *Splash* berisi ucapan selamat datang kepada pengguna.



Gambar 4.1 Halaman *Splash*

4.2.2 Halaman Utama

Selanjutnya yang tampil adalah halaman utama. Pada halaman utama terdapat tampilan peta beserta point atau titik lahan pertanian. Pengguna nantinya bisa memilih lahan peta berdasarkan point atau titik tersebut Gambar 4.2.



Gambar 4.2 Halaman Utama

Source code untuk menampilkan halaman utama adalah sebagai berikut:

```

getSupportActionBar().setBackgroundDrawable(new
ColorDrawable(Color.parseColor("#DDDDDD")));
    setContentView(R.layout.activity_main);

    FragmentManager fm =
getSupportFragmentManager();
    SupportMapFragment smf = (SupportMapFragment)
fm.findFragmentById(R.id.map);
    map = smf.getMap();
    map.setMapType(GoogleMap.MAP_TYPE_NORMAL);

    map.getUiSettings().setZoomControlsEnabled(true);

    map.animateCamera(CameraUpdateFactory.newLatLngZoom(ma
lang,12));

    // Initializing
    listJalan = new ArrayList<Polyline>();
    listMarker = new ArrayList<Marker>();
    listLahan = new ArrayList<Marker>();
    inputPoint = new ArrayList<Marker>();
    markerPoints = new ArrayList<LatLng>();
    route = new ArrayList<LatLng>();

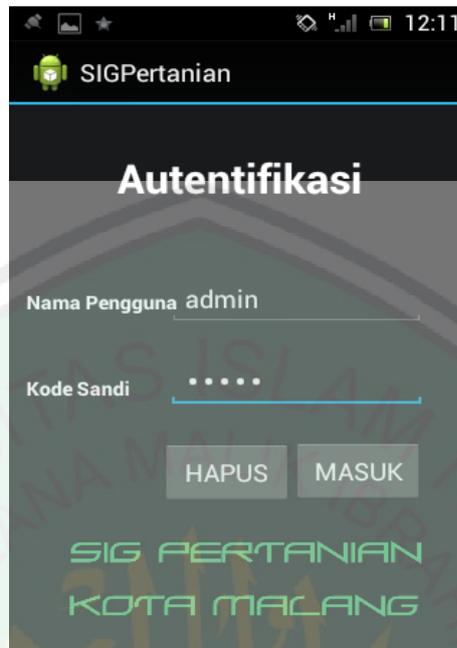
```

```
LoadTipeLahan ltp = new LoadTipeLahan();  
ltp.execute();  
  
ViewLahan vl = new ViewLahan();  
vl.execute();
```

4.2.3 Login

Login dilakukan oleh admin jika admin ingin menambah point atau titik lahan beserta data lahan tersebut.. user atau pengguna yang lain tidak perlu melakukan login jika ingin menggunakan SIG Pertanian untuk pencarian rute. Langkahnya adalah sebagai berikut:

1. Isi *username* dan *password* pada form *login*, kemudian tekan perintah *login*.
2. Jika *login* diterima maka akan masuk ke halaman utama, jika tidak maka ada peringatan bahwa autentifikasi gagal dan dipersilahkan untuk mengulangi memasukkan *username* dan *password* dengan benar. Dijelaskan pada Gambar 4.3 dibawah ini.



Gambar 4.3 Form Login

3. Source code untuk login adalah sebagai berikut :

```

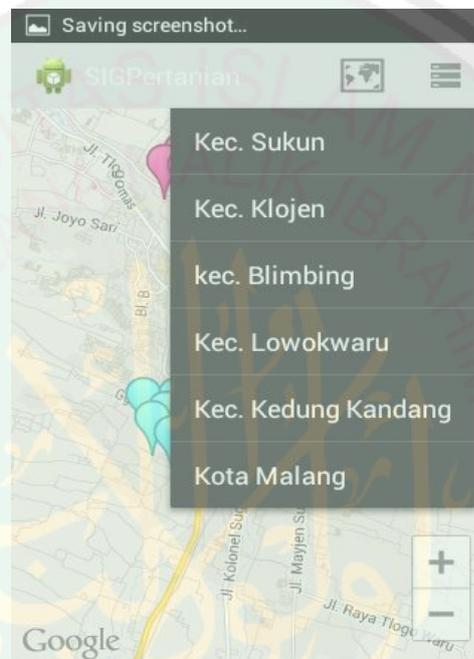
Button submit = (Button) findViewById(R.id.submit_btn);
submit.setOnClickListener (new OnClickListener()
{
    public void onClick (View v){
        List<NameValuePair> param = new
ArrayList<NameValuePair>(3);
        param.add(new BasicNameValuePair("uname",
name.getText().toString()));
        param.add(new BasicNameValuePair("paswd",
pass.getText().toString()));
        Ress = r.POST(link, param);
        try{
            JSONObject user = new
JSONObject(Ress);

            if(user.getString("argv").equals("1")){
                Intent i = null;
                if(SIG.getLV().equals("2")){
                    i = new
Intent(login.this,insert_peta.class);
                    startActivity(i);
                }
            }
        }else{

```

4.2.4 Halaman Kategori Wilayah

Pada halaman kategori wilayah terdapat pilihan kecamatan yang dapat dipilih pengguna sehingga tampilan peta lebih spesifik menuju kecamatan yang telah dipilih (Gambar 4.4)



Gambar 4.4 Kategori Wilayah

Source code untuk menampilkan kategori wilayah adalah sebagai berikut:

```
mLocation = menu.addSubMenu(MLocation, MLocation, 0,
null);
    mLocation.setIcon(R.drawable.ic_location_map);
    mLocation.addSubMenu(MLocation, MLocation_1,
MLocation_1, "Kec. Sukun");
    mLocation.addSubMenu(MLocation, MLocation_2,
MLocation_2, "Kec. Klojen");
    mLocation.addSubMenu(MLocation, MLocation_3,
MLocation_3, "kec. Blimbing");
    mLocation.addSubMenu(MLocation, MLocation_4,
MLocation_4, "Kec. Lowokwaru");
    mLocation.addSubMenu(MLocation, MLocation_5,
MLocation_5, "Kec. Kedung Kandang");

mLocation.getItem().setShowAsAction(MenuItem.SHOW_AS_A
```

CTION_ALWAYS) ;

4.2.5 Halaman Menu

Ketika pengguna menekan menu ditampilkan empat pilihan menu yaitu tampilan, input mode, hide lahan, cari rute (Gambar 4.5)



Gambar 4.5 Menu Tampilan

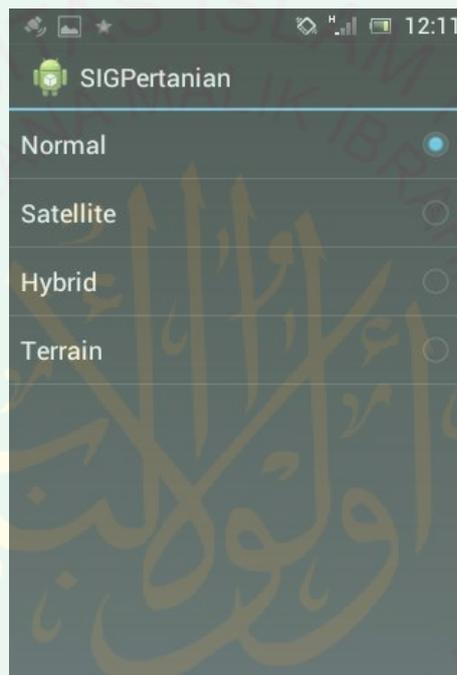
Source code untuk menampilkan menu adalah sebagai berikut:

```
mMenu = menu.addSubMenu(MMenu_biasa, MMenu_biasa, 0, null);

mMenu.setIcon(R.drawable.ic_device_access_storage);
mMenu.addSubMenu(MMenu_biasa, MMenu_1, MMenu_1,
"Tampilan");
mMenu.addSubMenu(MMenu_biasa, MMenu_2, MMenu_2,
"Input Mode");
mMenu.addSubMenu(MMenu_biasa, MMenu_3, MMenu_3,
"Hide Lahan");
mMenu.addSubMenu(MMenu_biasa, MMenu_4, MMenu_4,
"Cari Rute");
//mMenu.addSubMenu(MMenu_biasa, MMenu_4, MMenu_4,
"Logout");
```

4.2.6 Halaman Tampilan Peta

Ketika pengguna milih menu tampilan maka ditampilkan 4 pilihan tampilan peta yaitu normal, satelitt, hybrid, terrain dimana tampilan peta akan berubah sesuai tampilan peta yang di pilih (Gambar 4.6)



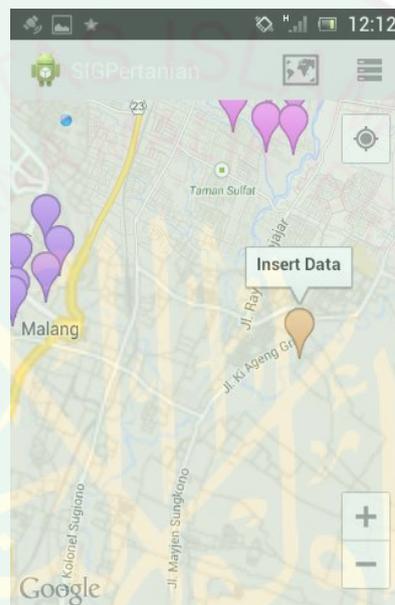
Gambar 4.6 Tampilan Petan

Source code untuk tampilan peta adalah sebagai berikut:

```
protected void changeMapMode(int mapMode){
    if(mapMode==0){
        map.setMapType(GoogleMap.MAP_TYPE_NORMAL);
    }else if(mapMode==1){
        map.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
    }else if(mapMode==2){
        map.setMapType(GoogleMap.MAP_TYPE_HYBRID);
    }else if(mapMode==3){
        map.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
    }
}
```

4.2.7 Halaman Penambah Lahan

Ketika pengguna menekan menu input mode maka pengguna yang khususnya admin bisa tambah lahan baru beserta datanya. (Gambar 4.7)



Gambar 4.7 Penambah Lahan

Source code untuk insert data matakuliah adalah sebagai berikut:

```

if (inputMode==1) {
    markerPoints.add(point);
    MarkerOptions options = new
MarkerOptions();

    // Setting the position of the marker
    options.position(point);
    options.title("Insert Data");
    //options.snippet(point.toString());

    options.icon(BitmapDescriptorFactory.defaultMarker(Bit
mapDescriptorFactory.HUE_ORANGE));

    marker = map.addMarker(options);
    marker.showInfoWindow();
    inputPoint.add(marker);

```

```

        map.setOnInfoWindowClickListener(new
        OnInfoWindowClickListener(){
        @Override
        public void onInfoWindowClick(Marker arg0) {
        List<String> ias = cv.getTipeLahan();
        Intent insert = new Intent(MainActivity.this,
        insertLahan.class);
        insert.putExtra("lat", arg0.getPosition().latitude);
        insert.putExtra("lng", arg0.getPosition().longitude);

        insert.putExtra("tipeLahan", ias.toArray(new
        String[ias.size()]));

        startActivityForResult(insert, 0);
        }

        });

```

4.2.8 Halaman Rute

Ketika pengguna menekan cari rute maka pengguna bisa menentukan posisi lahan yang akan dicari rute terpendek dari posisi pengguna sekarang. (Gambar 4.8)



Gambar 4.8 Rute Terpendek

Algoritma Dijkstra bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Misalkan titik menggambarkan gedung dan garis menggambarkan jalan, maka algoritma Dijkstra melakukan kalkulasi terhadap semua kemungki bobot terkecil dari setiap titik.

Pada Gambar 2.7 Contoh Keterhubungan *Dijkstra*. Yang dimana pertama-tama tentukan titik mana yang akan menjadi bobot jarak pada node lalu beri bobot jarak pada node pertama ke node terdekat satu per satu, Dijkstra akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap. Inilah urutan logika dari algoritma Dijkstra:

1. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga terhadap node lain (belum terisi).
2. Set semua node “Belum dilewati” dan set node awal sebagai “Node keberangkatan”.

Dari node keberangkatan, pertimbangkan node tetangga yang belum dilewati dan hitung jaraknya dari titik keberangkatan. Sebagai contoh, jika titik keberangkatan A ke B memiliki bobot jarak 6 dan dari B ke node C berjarak 2, maka jarak ke C melewati B menjadi $6+2=8$. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.

1. Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah dilewati sebagai “Node dilewati”. Node dilewati tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
2. Set “Node belum dilewati” dengan jarak terkecil (dari node keberangkatan) sebagai “Node Keberangkatan” selanjutnya dan lanjutkan dengan kembali ke step 3.

Implementasi Algoritma dijkstra dalam source code untuk menampilkan rute terdekat adalah sebagai berikut:

```
public class Algoritma {

    /** Receives a JSONObject and returns a list of lists
    containing latitude and longitude */
    public List<List<HashMap<String,String>>>
    parse(JSONObject jsonObject){

        List<List<HashMap<String, String>>> routes = new
        ArrayList<List<HashMap<String,String>>>() ;
        JSONArray jRoutes = null;
        JSONArray jLegs = null;
        JSONArray jSteps = null;

        try {
            jRoutes = jsonObject.getJSONArray("routes");

            /** Traversing all routes */
            for(int i=0;i<jRoutes.length();i++){
                jLegs =
                ((JSONObject)jRoutes.get(i)).getJSONArray("legs");
                List path = new
                ArrayList<HashMap<String, String>>();

                /** Traversing all legs */
                for(int j=0;j<jLegs.length();j++){
                    jSteps = (
                    (JSONObject)jLegs.get(j)).getJSONArray("steps");
```

```

        /** Traversing all steps */
        for(int
k=0;k<jSteps.length();k++){
            String polyline = "";
            polyline =
(String)((JSONObject)((JSONObject)jSteps.get(k)).get("polyli
ne")).get("points");
            List<LatLng> list =
decodePoly(polyline);
            //List list = (Vertex)
getShortestPathTo(list);
            //List ilst = (Vertex)
getShortestPathTo(list);
            //Vertex llst = null;
            //computePaths(llst);

            /** Traversing all points
*/
            for(int
l=0;l<list.size();l++){
                HashMap<String,
String> hm = new HashMap<String, String>();
                hm.put("lat",
Double.toString(((LatLng)list.get(l)).latitude) );
                hm.put("lng",
Double.toString(((LatLng)list.get(l)).longitude) );
                path.add(hm);
            }
            routes.add(path);
        }
    } catch (JSONException e) {
        e.printStackTrace();
    } catch (Exception e){
    }
    return routes;
}
}

```

```

private List<LatLng> decodePoly(String encoded) {

    List<LatLng> poly = new ArrayList<LatLng>();
    int index = 0, len = encoded.length();
    int lat = 0, lng = 0;

    while (index < len) {
        int b, shift = 0, result = 0;
        do {
            b = encoded.charAt(index++) - 63;

```

```

        result |= (b & 0x1f) << shift;
        shift += 5;
    } while (b >= 0x20);
    int dlat = ((result & 1) != 0 ? ~(result >>
1) : (result >> 1));
    lat += dlat;

    shift = 0;
    result = 0;
    do {
        b = encoded.charAt(index++) - 63;
        result |= (b & 0x1f) << shift;
        shift += 5;
    } while (b >= 0x20);
    int dlng = ((result & 1) != 0 ? ~(result >>
1) : (result >> 1));
    lng += dlng;

    LatLng p = new LatLng((((double) lat /
1E5)),
                        (((double) lng / 1E5)));
    poly.add(p);
}
return poly;
}

public static void computePaths(Vertex source) {
    source.minDistance = 0.;
    PriorityQueue<Vertex> vertexQueue = new
PriorityQueue<Vertex>();
    vertexQueue.add(source);

    while (!vertexQueue.isEmpty()) {
        Vertex u = vertexQueue.poll();

        // Visit each edge exiting u
        for (Edge e : u.adjacencies) {
            Vertex v = e.target;
            double weight = e.weight;
            double distanceThroughU =
u.minDistance + weight;
            if (distanceThroughU < v.minDistance)
            {
                vertexQueue.remove(v);
                v.minDistance =
distanceThroughU;
                v.previous = u;
                vertexQueue.add(v);
            }
        }
    }
}

```

```

    }

    public static List<Vertex> getShortestPathTo(Vertex
target) {
        List<Vertex> path = new ArrayList<Vertex>();
        for (Vertex vertex = target; vertex != null;
vertex = vertex.previous)
            path.add(vertex);
        Collections.reverse(path);
        return path;
    }
}

```

4.2.9 Tentang Program

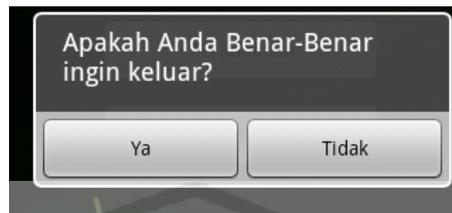
Berisi tentang identitas pembuat dan maksud dari pembuatan aplikasi SIG Pertanian. (Gambar 4.9)



Gambar 4.9 Tentang Program

4.2.10 Logout

Berisi tentang permintaan untuk informasi keluar atau tidak dari aplikasi SIG Pertanian. (Gambar 4.10)



Gambar 4.10 Logout

Source code untuk logout dari aplikasi adalah sebagai berikut:

```
private void keluar() {
    AlertDialog.Builder builder = new
AlertDialog.Builder(this);
    builder.setMessage("Apakah Anda Benar-Benar ingin
keluar?")
        .setCancelable(false)
        .setPositiveButton("Ya",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog,
                    int id) {
                    dsn_home.this.finish();
                }
            })
        .setNegativeButton("Tidak",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog,
                    int id) {
                    dialog.cancel();
                }
            })
        .show();
}
```

4.3 Impelemntasi Algoritma

Penerapan algoritma Dijkstra pada penelitian ini memiliki beberapa tahap yaitu pengambilan data dari Google Apis berupa data JSON, ekstraksi data JSON, pencarian data dalam algoritma, kemudian menggambar hasil pencarian pada tampilan.

4.3.1 Pengambilan Data

Untuk pengambilan data dari Google Apis, pada aplikasi client dibuat class tersendiri yang sifatnya asinkron. Class yang asinkron

memungkinkan aplikasi tetap berjalan walau prosesnya belum selesai. Class asinkron berjalan di belakang layar (background).

Berikut source code pada aplikasi yang digunakan untuk mendownload data dari Google Apis :

```
private String getDirectionsUrl(LatLng origin,LatLng dest){
    // Origin of route
    String str_origin =
"origin="+origin.latitude+","+origin.longitude;
    String str_dest =
"destination="+dest.latitude+","+dest.longitude;
    String sensor = "sensor=false";
    String mode = "mode=driving";
    String altr = "&alternatives=true";

    if(RMOD == RMOD_NONE){
        mode = "mode=driving";
        altr = "";
    }else if(RMOD == RMOD_DRIVE){
        mode = "mode=driving";
    }else if(RMOD == RMOD_CYCLE){
        mode = "mode=bicycling";
    }else if(RMOD == RMOD_WALK){
        mode = "mode=walking";
    }

    String parameters =
str_origin+"&"+str_dest+"&"+sensor+"&"+mode+altr;
    String output = "json";
    String url =
"https://maps.googleapis.com/maps/api/directions/"+output+"?"+parameters;
    System.out.println(url);

    return url;
}

/** A method to download json data from url */
public String downloadUrl(String strUrl) throws
IOException{
    String data = "";
    InputStream iStream = null;
    HttpURLConnection urlConnection = null;
    try{
        URL url = new URL(strUrl);
        urlConnection = (HttpURLConnection)
url.openConnection();
```

```

        urlConnection.connect();
        iStream = urlConnection.getInputStream();
        BufferedReader br = new BufferedReader(new
InputStreamReader(iStream));
        StringBuffer sb = new StringBuffer();
        String line = "";
        while( ( line = br.readLine()) != null){
            sb.append(line);
        }

        data = sb.toString();
        br.close();

    }catch(Exception e){
        Log.d("Exception while downloading url",
e.toString());
    }finally{
        iStream.close();
        urlConnection.disconnect();
    }
    return data;
}

// Fetches data from url passed
public class DownloadTask extends AsyncTask<String,
Void, String>{
    @Override
    protected String doInBackground(String... url) {
        String data = "";

        try{
            // Fetching the data from web service
            data = downloadUrl(url[0]);
        }catch(Exception e){
            Log.d("Background Task",e.toString());
        }
        return data;
    }

    @Override
    protected void onPostExecute(String result) {
        super.onPostExecute(result);
        ParserTask parserTask = new ParserTask();
        parserTask.execute(result);
    }
}

/** A class to parse the Google Places in JSON format */
private class ParserTask extends AsyncTask<String,
Integer, List<List<HashMap<String,String>>> >{
    @Override

```

```

        protected List<List<HashMap<String, String>>>
doInBackground(String... jsonData) {
    JSONObject jsonObject;
    List<List<HashMap<String, String>>> routes =
null;

    try{
        jsonObject = new JSONObject(jsonData[0]);
        Algoritma parser = new Algoritma();
        routes = parser.parse(jsonObject);
    }catch(Exception e){
        e.printStackTrace();
    }
    return routes;
}

@Override
protected void
onPostExecute(List<List<HashMap<String, String>>> result) {
    ArrayList<LatLng> points = null;
    PolylineOptions lineOptions = null;

    // Traversing through all the routes
    for(int i=0;i<result.size();i++){
        points = new ArrayList<LatLng>();
        lineOptions = new PolylineOptions();
        List<HashMap<String, String>> path =
result.get(i);
        for(int j=0;j<path.size();j++){
            HashMap<String,String> point =
path.get(j);

            double lat =
Double.parseDouble(point.get("lat"));
            double lng =
Double.parseDouble(point.get("lng"));
            LatLng position = new LatLng(lat, lng);

            points.add(position);
        }

        // Adding all the points in the route to
LineOptions
        lineOptions.addAll(points);
        lineOptions.width(2);

        if(RMOD == RMOD_NONE){
            lineOptions.color(Color.RED);
        }else if(RMOD == RMOD_DRIVE){
            lineOptions.color(Color.BLUE);
        }else if(RMOD == RMOD_CYCLE){

```

```

        lineOptions.color(Color.BLUE);
    }else if(RMOD == RMOD_WALK){
        lineOptions.color(Color.BLUE);
    }

    jalan = map.addPolyline(lineOptions);
    listJalan.add(jalan);

    if(RMOD == RMOD_NONE){
    }else if(RMOD == RMOD_DRIVE){
        RMOD = RMOD_CYCLE;
        String url = getDirectionsUrl(ORIGIN,
DESTINATION);
        DownloadTask downloadTask = new DownloadTask();
        downloadTask.execute(url);
    }else if(RMOD == RMOD_CYCLE){
        RMOD = RMOD_WALK;
        String url = getDirectionsUrl(ORIGIN,
DESTINATION);
        DownloadTask downloadTask = new DownloadTask();
        downloadTask.execute(url);
    }else if(RMOD == RMOD_WALK){
        RMOD = RMOD_NONE;
        String url = getDirectionsUrl(ORIGIN,
DESTINATION);
        DownloadTask downloadTask = new DownloadTask();
        downloadTask.execute(url);
    }
    }

    if(result.size()<1){
        Toast.makeText(getBaseContext(),
Points", Toast.LENGTH_SHORT).show();
        return;
    }
}
}
}

```

4.3.2 Ekstraksi JSON

Data yang didapat dari Google Apis berupa format JSON.

Berikut ini fungsi yang digunakan untuk mengekstrak data JSON.

```

/** Receives a JSONObject and returns a list of lists
containing latitude and longitude */
public List<List<HashMap<String,String>>>
parse(JSONObject jobject){

```

```

        List<List<HashMap<String, String>>> routes =
new ArrayList<List<HashMap<String, String>>>();
        JSONArray jRoutes = null;
        JSONArray jLegs = null;
        JSONArray jSteps = null;

        try {
            jRoutes = jsonObject.getJSONArray("routes");

            /** Traversing all routes */
            for(int i=0;i<jRoutes.length();i++){
                jLegs =
((JSONObject)jRoutes.get(i)).getJSONArray("legs");
                List path = new
ArrayList<HashMap<String, String>>();

                /** Traversing all legs */
                for(int j=0;j<jLegs.length();j++){
                    jSteps = (
(JSONObject)jLegs.get(j)).getJSONArray("steps");

                    /** Traversing all steps */
                    for(int
k=0;k<jSteps.length();k++){
                        String polyline = "";
                        polyline =
(String)((JSONObject)((JSONObject)jSteps.get(k)).get("poly
line")).get("points");
                        List<LatLng> list =
decodePoly(polyline);
                        //List list = (Vertex)
getShortestPathTo(list);
                        //List list = (Vertex)
getShortestPathTo(list);
                        //Vertex list = null;
                        //computePaths(list);

                        /** Traversing all points
*/
                            for(int
l=0;l<list.size();l++){
                                HashMap<String,
String> hm = new HashMap<String, String>();
                                hm.put("lat",
Double.toString(((LatLng)list.get(l)).latitude) );
                                hm.put("lng",
Double.toString(((LatLng)list.get(l)).longitude) );
                                path.add(hm);
                            }
                        }
                    routes.add(path);
                }
            }
        }
    }
}

```

```

    }
}
} catch (JSONException e) {
    e.printStackTrace();
} catch (Exception e){
}
return routes;
}

```

4.3.3 Penghitungan Algoritma Dan Drawing

Fungsi berikut untuk decode polygon dan perhitungan algoritma Dijkstra.

```

private List<LatLng> decodePoly(String encoded) {

    List<LatLng> poly = new ArrayList<LatLng>();
    int index = 0, len = encoded.length();
    int lat = 0, lng = 0;

    while (index < len) {
        int b, shift = 0, result = 0;
        do {
            b = encoded.charAt(index++) - 63;
            result |= (b & 0x1f) << shift;
            shift += 5;
        } while (b >= 0x20);
        int dlat = ((result & 1) != 0 ? ~(result
>> 1) : (result >> 1));
        lat += dlat;

        shift = 0;
        result = 0;
        do {
            b = encoded.charAt(index++) - 63;
            result |= (b & 0x1f) << shift;
            shift += 5;
        } while (b >= 0x20);
        int dlng = ((result & 1) != 0 ? ~(result
>> 1) : (result >> 1));
        lng += dlng;

        LatLng p = new LatLng((((double) lat /
1E5)),
                                (((double) lng / 1E5)));
    }
}

```

```

        poly.add(p);
    }
    return poly;
}

public static void computePaths(Vertex source) {
    source.minDistance = 0.;
    PriorityQueue<Vertex> vertexQueue = new
PriorityQueue<Vertex>();
    vertexQueue.add(source);

    while (!vertexQueue.isEmpty()) {
        Vertex u = vertexQueue.poll();

        // Visit each edge exiting u
        for (Edge e : u.adjacencies) {
            Vertex v = e.target;
            double weight = e.weight;
            double distanceThroughU =
u.minDistance + weight;
            if (distanceThroughU <
v.minDistance) {
                vertexQueue.remove(v);
                v.minDistance =
distanceThroughU;
                v.previous = u;
                vertexQueue.add(v);
            }
        }
    }
}

public static List<Vertex>
getShortestPathTo(Vertex target) {
    List<Vertex> path = new ArrayList<Vertex>();
    for (Vertex vertex = target; vertex != null;
vertex = vertex.previous)
        path.add(vertex);
    Collections.reverse(path);
    return path;
}

```

4.4 Uji Coba Sistem

Setelah program dapat bekerja dengan baik pada komputer dan *emulator*, maka selanjutnya program akan diuji coba pada *smartphone* yang sebenarnya. Hal ini bertujuan untuk mengetahui apakah program juga dapat bekerja dengan baik pada *smartphone* yang sebenarnya. Selain itu juga untuk mengetahui seberapa efektifkah proses- proses yang dapat dilakukan.

4.4.1 Proses uji coba

Table 4.1 Uji Coba Pada Beberapa *Smartphone* ber Sistem Operasi Android

No	Jenis <i>Smartphone</i>	Keterangan
1.	Samsung Galaxy Tab	Berjalan dengan baik, interface form login kurang sesuai dengan resolusi <i>smartphone</i>
2.	Samsung Galaxy Wonder	Berjalan dengan baik
3.	Samsung Galaxy Ace	Berjalan dengan baik
4.	Samsung Galaxy Gio	Berjalan dengan baik
5.	Samsung Galaxy Mini	Berjalan dengan baik
6.	Samsung Galaxy Young	Berjalan dengan baik, button form login terlihat besar
7.	Sony Ericsson Xperia Mini Pro	Berjalan dengan baik
8.	Sony Ericsson Xperia	Berjalan dengan baik

4.4.2 Analisa Hasil Uji Coba

Pengguna yang diikutsertakan dalam uji coba aplikasi SIG Pertanian ini adalah masyarakat kota malang. Jumlah pengguna yang turut serta dalam uji coba aplikasi adalah 25 orang. Hal ini dilakukan agar pada proses uji coba benar-benar diketahui bahwa aplikasi sudah sesuai dengan kebutuhan masyarakat. Setelah dilakukan

ujicoba secara acak maka akan dilakukan analisis hasil untuk mengetahui tingkat akurasi dari aplikasi yang dibangun.



Table 4.2 Hasil Rekapitulasi Kuesioner

No	Komponen Penilaian	Nilai										Total responder	Total nilai	Rata-rata	%	Keterangan				
		1	2	3	4	5	6	7	8	9	10									
1	Tampilan				3	15	6	36	5	35	7	56	4	36	25	178	7.12	64	Cukup	
2	Kemudahan						4	24	6	42	12	96	3	27	25	189	7.56	84	Mudah	
3	Fasilitas				1	5	7	42	8	56	7	56	2	18	25	177	7.08	68	Cukup mendukung	
4	Kecepatan					5	25	2	12	10	70	5	40	4	36	25	183	7.32	76	cepat
5	Keakuratan data				4	20	5	30	7	49	8	64	1	9	25	172	6.88	64	Cukup akurat	
6	Kepraktisan						5	30	9	63	6	48	5	45	25	186	8	84	praktis	
7	Kelayakan						3	18	6	42	12	96	4	36	25	192	7.68	88	Layak	

Keterangan nilai rata-rata:

1 – 3.9 : kurang (nilai terendah)

4 – 6.9 : cukup

7 – 10 : (nilai tertinggi)

Sebanyak 25 pengguna telah melakukan uji coba pada aplikasi *SIG Pertanian*, setelah itu pengguna sebagai responden mengisi nilai pada masing- masing pertanyaan kuesioner. Rentang nilai yaitu antara 1-10. Dengan ketentuan bahwa nilai 1 – 3.9 adalah nilai terendah, 4 – 6.9 bernilai cukup, 7 – 10 nilai tertinggi. Pada kolom masing- masing nilai terdapat 2 kolom, kolom pertama adalah jumlah responden yang memilih nilai tersebut, sedangkan pada kolom kedua adalah hasil kali nilai yang dipilih dengan jumlah responden. Rata- rata diperoleh dari jumlah keseluruhan pada kolom kedua dibagi dengan keseluruhan jumlah responden. Persentase diperoleh dari jumlah keseluruhan pemilih pada nilai tertinggi yaitu nilai 7,8 dan 9 (lihat Table 4.2).

Dari hasil uji coba dapat disimpulkan bahwa hasil penilaian oleh pengguna terhadap aplikasi *SIG Pertanian* menunjukkan penilaian yang baik. Hasil rata- rata nilai yang telah dipilih responden menyatakan bahwa aplikasi *SIG Pertanian*:

1. Memiliki tampilan yang cukup bagus
2. Mudah digunakan
3. Menyediakan fasilitas yang cukup mendukung
4. Cepat dalam menyajikan informasi
5. Cukup akurat dalam menyajikan data
6. Praktis sebagai aplikasi *SIG*
7. Layak untuk digunakan

Oleh karena itu, aplikasi SIG Pertanian ini layak diimplementasikan. SIG Pertanian juga dapat membantu mempermudah pemantauan lahan pertanian milik petani.

4.5 SIG Pencarian Lokasi lahan pertanian dalam Pandangan Islam

Dalam Q.S Al Mujadalah:11 disebutkan bahwa sesungguhnya Allah SWT meninggikan beberapa derajat orang-orang yang beriman dan orang-orang yang berilmu pengetahuan. Jadi, Islam merupakan agama yang memuliakan dan mewajibkan umatnya baik laki-laki maupun perempuan untuk senantiasa menuntut ilmu.

Hingga saat ini, *SIG (Sistem Informasi Geografis)* mengalami banyak perkembangan dan sangat mempermudah manusia untuk menuntut ilmu dengan belajar (Abdurrahman R Effendi dan Gina Puspita, 2007, hal. 13). Bukankah manusia dilahirkan tanpa mengetahui sesuatu apapun, sedangkan manusia dituntut untuk mengemban tugas sebagai khalifah didunia ini dengan mengamalkan ilmu pengetahuan.

وَاللَّهُ أَخْرَجَكُمْ مِنْ بُطُونِ أُمَّهَاتِكُمْ لَا تَعْلَمُونَ شَيْئًا وَجَعَلَ لَكُمُ السَّمْعَ وَالْأَبْصَرَ وَالْأَفْئِدَةَ
لَعَلَّكُمْ تَشْكُرُونَ ﴿٧٨﴾

Artinya:

“Dan Allah mengeluarkan kamu dari perut ibumu dalam Keadaan tidak mengetahui sesuatupun, dan Dia memberi kamu pendengaran, penglihatan dan hati, agar kamu bersyukur”.(QS. An Nahl / 16: 78)

وَإِذْ قَالَ رَبُّكَ لِلْمَلَائِكَةِ إِنِّي جَاعِلٌ فِي الْأَرْضِ خَلِيفَةً ۗ قَالُوا أَتَجْعَلُ فِيهَا مَنْ يُفْسِدُ فِيهَا وَيَسْفِكُ الدِّمَاءَ وَنَحْنُ نُسَبِّحُ بِحَمْدِكَ وَنُقَدِّسُ لَكَ ۗ قَالَ إِنِّي أَعْلَمُ مَا لَا تَعْلَمُونَ ﴿٣٠﴾

Artinya :

“Ingatlah ketika Tuhanmu berfirman kepada Para Malaikat: “Sesungguhnya aku hendak menjadikan seorang khalifah di muka bumi.” mereka berkata: “Mengapa Engkau hendak menjadikan (khalifah) di bumi itu orang yang akan membuat kerusakan padanya dan menumpahkan darah, Padahal Kami Senantiasa bertasbih dengan memuji Engkau dan mensucikan Engkau?” Tuhan berfirman: “Sesungguhnya aku mengetahui apa yang tidak kamu ketahui”. (QS. Al Baqarah / 2: 30).

Jika kita mengingat lagi hadist nabi Muhammad SAW yang diriwayatkan oleh Thabrani: *”Barangsiapa yang menginginkan kehidupan dunia, maka ia harus memiliki ilmu, dan barang siapa yang menginginkan kehidupan akhirat maka itupun harus dengan ilmu,...”* maka kehadiran *mobile learning* menarik perhatian orang untuk mendapatkan informasi yang berupa ilmu pengetahuan karena ilmu adalah penyelaras kehidupan akherat dan dunia.

Secara umum, Al-Qur’an menyatakan bahwa orang yang berilmu tidak sama dengan orang yang tidak berilmu. Allah SWT berfirman dalam surat Azumar ayat 9:

أَمْ مَنْ هُوَ قَنِيئٌ ءَانَاءَ اللَّيْلِ سَاجِدًا وَقَائِمًا يَحْذَرُ الْآخِرَةَ وَيَرْجُوا رَحْمَةَ رَبِّهِ ۗ قُلْ هَلْ يَسْتَوِي الَّذِينَ يَعْمُونَ وَالَّذِينَ لَا يَعْلَمُونَ ۗ إِنَّمَا يَتَذَكَّرُ أُولُو الْأَلْبَابِ ﴿٩﴾

Artinya:

“(Apakah kamu Hai orang musyrik yang lebih beruntung) ataukah orang yang beribadat di waktu-waktu malam dengan sujud dan berdiri, sedang ia takut kepada (azab) akhirat dan mengharapkan rahmat Tuhannya? Katakanlah: “Adakah sama orang-orang yang mengetahui dengan orang-orang yang tidak mengetahui?” Sesungguhnya orang yang berakallah yang dapat menerima pelajaran”.(QS. Az Zumar / 39: 9)

Dari berbagai ayat yang menganugerahkan kegunaan ilmu pengetahuan dapat disimpulkan (Syahrin Harahap, 1999, hal. 8) , bahwa fungsi dan kegunaan ilmu pengetahuan adalah sebagai berikut :

Menyadari jati diri dan memahami berbagai kebaikan yang terkandung dalam ajaran syariat. Allah SWT berfirman:

أَيَّامًا مَّعْدُودَاتٍ ۚ فَمَن كَانَ مِنكُم مَّرِيضًا أَوْ عَلَىٰ سَفَرٍ فَعِدَّةٌ مِّنْ أَيَّامٍ أُخَرَ ۗ وَعَلَىٰ الَّذِينَ يُطِيقُونَهُ فِدْيَةٌ طَعَامُ مِسْكِينٍ ۚ فَمَن تَطَوَّعَ خَيْرًا فَهُوَ خَيْرٌ لَّهُ ۗ وَأَن تَصُومُوا خَيْرٌ لَّكُمْ إِن كُنتُمْ تَعْلَمُونَ ﴿١٨٤﴾

Artinya:

“(Yaitu) dalam beberapa hari yang tertentu. Maka Barangsiapa diantara kamu ada yang sakit atau dalam perjalanan (lalu ia berbuka), Maka (wajiblah baginya berpuasa) sebanyak hari yang ditinggalkan itu pada hari-hari yang lain. dan wajib bagi orang-orang yang berat menjalankannya (jika mereka tidak berpuasa) membayar fidyah, (yaitu): memberi Makan seorang miskin. Barangsiapa yang dengan kerelaan hati mengerjakan kebajikan[114], Maka Itulah yang lebih baik baginya. dan berpuasa lebih baik bagimu jika kamu mengetahui”.(QS. Al Baqarah / 2: 184).

Mengetahui rahasia alam, orang yang dapat menangkap pelajaran dari penciptaan alam raya adalah orang-orang yang mendayagunakan akalinya. Fakta tersebut sekaligus menggambarkan adanya manfaat yang harus oleh orang-orang yang memerhatikan fenomena alam. Allah SWT berfirman:

إِنَّ فِي خَلْقِ السَّمَوَاتِ وَالْأَرْضِ وَاخْتِلَافِ اللَّيْلِ وَالنَّهَارِ وَالْفُلْكِ الَّتِي تَجْرِي فِي الْبَحْرِ بِمَا يَنْفَعُ النَّاسَ وَمَا أَنْزَلَ اللَّهُ مِنَ السَّمَاءِ مِنْ مَّاءٍ فَأَحْيَا بِهِ الْأَرْضَ بَعْدَ مَوْتِهَا وَبَثَّ فِيهَا مِنْ كُلِّ دَابَّةٍ وَتَصْرِيفِ الرِّيْحِ وَالسَّحَابِ الْمُسَخَّرِ بَيْنَ السَّمَاءِ وَالْأَرْضِ لَآيَاتٍ لِّقَوْمٍ يَعْقِلُونَ ﴿١٦٤﴾

Artinya:

“Sesungguhnya dalam penciptaan langit dan bumi, silih bergantinya malam dan siang, bahtera yang berlayar di laut membawa apa yang berguna bagi manusia, dan apa yang Allah turunkan dari langit berupa air, lalu dengan air itu Dia hidupkan bumi sesudah mati (kering)-nya dan Dia sebarkan di bumi itu segala jenis hewan, dan pengisaran angin dan awan yang dikendalikan antara langit dan bumi; sungguh (terdapat) tanda-tanda (keesaan dan kebesaran Allah) bagi kaum yang memikirkan”.(QS. Al Baqarah / 2: 164).

Terpenuhinya tuntutan hidup yang lebih baik, dengan ilmu pengetahuan, manusia bisa merekayasa dan menciptakan inovasi dengan memanfaatkan sumber daya yang telah Allah sediakan jauh lebih besar.

Allah berfirman:

وَمِنَ الْأَنْعَامِ حَمُولَةً وَفَرْشًا ۚ كُلُوا مِمَّا رَزَقَكُمُ اللَّهُ وَلَا تَتَّبِعُوا خُطُوتِ الشَّيْطَانِ ۚ إِنَّهُ لَكُمْ عَدُوٌّ مُّبِينٌ ﴿١٤٢﴾

Artinya:

“Dan di antara hewan ternak itu ada yang dijadikan untuk pengangkutan dan ada yang untuk disembelih. makanlah dari rezki yang telah diberikan Allah kepadamu, dan janganlah kamu mengikuti langkah-langkah syaitan. Sesungguhnya syaitan itu musuh yang nyata bagimu”.(QS. Al An’am / 6 : 142)

BAB V

PENUTUP

5.1 Kesimpulan

1. Dari hasil uji coba, responden menunjukkan penilaian yang baik terhadap aplikasi SIG Pertanian. Hal ini berdasarkan bahwa 68 % responden menyatakan tampilan aplikasi bagus, 96 % menyatakan aplikasi mudah digunakan, 64 % menyatakan fasilitas yang disediakan mendukung, 42 % menyatakan cukup cepat dalam menyajikan informasi, 62 % menyatakan cukup akurat dalam menyajikan data, 88 % menyatakan praktis sebagai aplikasi akademik, 96 % menyatakan aplikasi layak untuk digunakan. Sehingga dapat disimpulkan bahwa aplikasi SIG Pertanian dapat diimplementasikan guna membantu pencarian lokasi lahan Pertanian. Sedangkan dari hasil pencarian menggunakan Algoritma *Dijkstra* yang penerapannya pada SIG Pertanian berbasis *Android* dapat diukur melalui kemampuan pencarian dan juga keakuratan terdapat jarak tempuh pada peta dengan membandingkan pada jarak tempuh menggunakan alat ukur manual dapat di peroleh keakuratan Algoritma *Dijkstra* mencapai 98,70 %. Dari hasil ini diperoleh tingkat keakuratan yang tinggi sehingga dapat disimpulkan penerapan Algoritma *Dijkstra* pada SIG Pertanian berbasis *Android* layak untuk dikembangkan.

2. Aplikasi SIG Pertanian dirancang menjadi dua sisi, yaitu SIG Pertanian *provider* dan SIG Pertanian *client*.

5.2 Saran

Perlu diadakan penelitian lebih lanjut untuk memaksimalkan fungsi SIG Pertanian, misal menambahkan: penambahan objek lebih kompleks, audio tutorial. Sehingga, aplikasi ini mempunyai fitur yang lebih lengkap.



DAFTAR PUSTAKA

- Abdulrahim, Imaduddin.1997. *Al-Quran Merangsang Perkembangan Ilmu dan Teknologi*. Jakarta: Gema Insani Press.
- Abdurrahman R Effendi dan Gina Puspita,2007. *Membangun Sains dan Teknologi Menurut Kehendak Tuhan*. Jakarta: Giliran Timur.
- Al-Maragi, Ahmad Mustafa.1992.*Terjemah Tafsir Al-Maragi*. Semarang: Toha Putra.
- Anonim. 2010. *Profil Kabupaten atau Kota Malang*. Malang: Ditjen Cipta Karya
- Arifin, Muhammad Zaenal. 2012. *Sistem Informasi Geografis untuk Fasilitas Perguruan Tinggi Berbasis Android Di Kota Surabaya*. Surabaya. Jurusan Teknik Informatika. Politeknik Elektronika Negeri Surabaya. Institut Teknologi Sepuluh Nopember.
- Azam, zainal. 2012. *Islam Tuntut Umat Usahakan Pertanian Hingga Akhir Zaman*.http://pasbukitbendera.com/index.php?option=com_content&task=view&id=147&Itemid=45. Dalam www.google.com. Diakses pada tanggal 15 Agustus 2013. Pukul 18.00 WIB.
- Bagir, Zainal Abidin. “Sains dan Agama-Agama: Perbandingan Beberapa Tipologi Mutakhir”. Dalam Zainal Abidin, Liek, Arqom, dan M. Yusuf (eds.), Ilmu, Etika dan Agama: Menyingkap Tabir Alam dan Manusia, Yogyakarta: CRCS UGM, 2006.
- Greeners. 2012. *Lahan Pertanian di Kota Malang Tersisa 1.300 Hektare*. <http://www.greenersmagz.com> Diakses pada tanggal 01 September 2013.
- Harahap, Syahrin. *Islam Untuk Berbagai Aspek Kehidupan*. Yogyakarta: Tiara Wacana Yogya, 1999.
- Irsyad, pelita. 2002. *Pertanian dan Islam*. http://www.brunet.bn/gov/mufti/irsyad/pelita/2002/ic44_2002.htm. Dalam www.google.com. Pukul 18.00, WIB.
- Kadir, A. 1999. *Konsep dan Tuntunan Praktis Basis Data*. Yogyakarta: Andi.
- Mahrn, Jamaluddin. 2005. *Al-Qur'an Bertutur Tentang Makanan dan Obat-Obatan*. Yogyakarta: Mitra Pustaka.
- Munir, Rinaldi. 2010. *Matematika Diskrit*. Bandung: Informatika.

- Najati, Muhammad Utsman. 2005. *Psikologi dalam Al-Quran*. Terj. M. Zaka Al-Farisi dari Judul Asli *Al-Quran Wa Ilmun Nafsi*. Bandung: CV Pustaka Setia.
- Nata, Abuddin. 2002. *Tafsir Ayat-ayat Pendidikan*. Jakarta: Raja Grafindo Persada.
- Nata, Abuddin. 2009. *Metodologi Studi Islam*. Jakarta: Rajawali Press.
- Quthb, Sayyid. 2004. *Fi Zhilalil-Qur'an*. Jakarta: Gema Insani.
- Riyanto, 2010. *Sistem Informasi Geografis Berbasis Mobile + CD*. Yogyakarta. Andi Offset.
- Rosadisastra, Andi, Metode Tafsir Ayat- Ayat Sains dan Sosial, Jakarta : Amzah, 2007
- Safaat, N., 2011. *Android, Pemrograman Aplikasi Mobile Smartphone dan Tablet PC berbasis Android*. Bandung: Informatika.
- Safaat, N., 2012. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Revisi ed. Bandung: Informatika.
- Saputra, Haris. 2012. *Modul Pembelajaran Praktek Basis Data (MySQL)*. Baturaja. Akmi Baturaja.
- Siang, J. Jek. 2002. *Matematika Diskrit dan Aplikasinya pada Ilmu Komputer*, Yogyakarta. Andi Offset.
- Suhadak, Akhmad. 2010. *Sistem Informasi Geografis Lahan Pertanian Kabupaten Banyuwangi*. Malang. Jurusan Teknik Informatika. Fakultas Sains dan Teknologi. Universitas Islam Negeri Maulana Malik Ibrahim.
- Shihab, M. Quraish. 2002. *Tafsir Al-Misbah: Pesan, Kesan dan Keserasian Al-Qur'an*. Jakarta: Lentera Hati.
- Siswanto. 2012. *Sistem Informasi Geografis Objek Wisata Menggunakan Google Maps Api Studi Kasus Kabupaten Mojokerto*. Surabaya. Jurusan Teknik Informatika. Politeknik Elektronika Negeri Surabaya. Institut Teknologi Sepuluh Nopember.
- Shihab, Quraish. 2007. *Membumikan Al-Qur'an Fungsi dan Peran Wahyu dalam Kehidupan Masyarakat*. Bandung : Mizan.