

**APLIKASI SISTEM INFORMASI GEOGRAFIS  
PENCARIAN LOKASI LEMBAGA KURSUS  
BERBASIS ANDROID DI  
KOTA GORONTALO**  
(Studi Kasus Dinas Pendidikan Kota Gorontalo)

**Oleh**  
**RIA PRATIWI ULOLI**  
**T3114294**

**SKRIPSI**



**PROGRAM SARJANA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS ICHSAN GORONTALO  
GORONTALO  
2017**

## **HALAMAN PERSETUJUAN**

# **APLIKASI SISTEM INFORMASI GEOGRAFIS PENCARIAN LOKASI LEMBAGA KURSUS BERBASIS ANDROID DI KOTA GORONTALO**

(Studi Kasus Dinas Pendidikan Kota Gorontalo)

Oleh

**RIA PRATIWI ULOLI**

T3114294

## **SKRIPSI**

Untuk memenuhi salah satu syarat ujian  
guna memperoleh gelar Sarjana  
Program Studi Teknik Informatika, ini  
telah disetujui oleh Tim Pembimbing.

Gorontalo, April 2017

Pembimbing Utama

Pembimbing Pendamping

**Irvan Abraham Salihi, M.Kom**  
**NIDN : 0928028101**

**Irma Surya Kumala I, M.Kom**  
**NIDN : 0921128801**

## HALAMAN PENGESAHAN

# APLIKASI SISTEM INFORMASI GEOGRAFIS PENCARIAN LOKASI LEMBAGA KURSUS BERBASIS ANDROID DI KOTA GORONTALO

(Studi Kasus Dinas Pendidikan Kota Gorontalo)

Oleh

RIA PRATIWI ULOLI

T3114294

Diperiksa oleh Panitia Ujian Strata Satu (S1)  
Universitas Ichsan Gorontalo

1. Ketua Penguji  
**Amiruddin, M.Kom** .....
2. Anggota  
**Haditsah Annur, M.Kom** .....
3. Anggota  
**Muis Nanja, M.Kom** .....
4. Anggota  
**Irvan Abraham Salihi, M.Kom** .....
5. Anggota  
**Irma Surya Kumala I, M.Kom** .....

## **HALAMAN PERNYATAAN**

Dengan ini saya menyatakan bahwa :

1. Karya tulis saya (Skripsi) ini adalah asli dan belum pernah di ajukan untuk mendapatkan gelar akademik (Sarjana) baik di Universitas Ichsan Gorontalo maupun di Perguruan Tinggi Lainnya.
2. Karya tulis ini murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan dari Tim Pembimbing.
3. Dalam karya tulis ini tidak terdapat karya ataupun pendapat yang telah dipublikasikan orang lain, kecuali secara tertulis dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi ini.

Gorontalo, April 2017  
Yang Membuat Pernyataan

Ria Pratiwi Uloli  
T3114294

## **ABSTRACT**

*One of the most important aspect of a city is developing the educational aspect, especially in Gorontalo City. The quality of education sector in a good city will also affect the quality of the city. The quality of education is also supported by some factors, one of which is the presence of courses institution. Generally, people will look for the courses institution with the good facilities, teachers were high-grade teaching, affordable price and based on need. To facilitate the public to get information about the courses institution, so at this study created an application to find the locations of courses institutions using smartphone with Android operating system. Android-based smartphone have been selected because it is many people use it.*

*This application is designed to facilitate the public to know the location of the courses institution and to ease the public to get the complete information about the courses institution that they seek. This application is built by using Android Studio and Java programming language with the Android Operating System. Android is very flexible and can be used on a variety of hardware platforms and easy to use. This application, harness google maps service accessed via Internet and has some menus like searching the location by the name of courses institution, searching the location by category of courses institution, about and exit.*

*Keywords:* *Geographic Information Systems, Location Search of Course Institution, Android, Smartphone*

## **ABSTRAK**

Salah satu aspek terpenting dalam mengembangkan suatu kota adalah aspek pendidikan, khususnya di Kota Gorontalo. Kualitas pendidikan disuatu kota yang baik akan berpengaruh juga pada kualitas kota tersebut. Kualitas pendidikan juga ditunjang oleh berbagai faktor, salah satunya adalah kehadiran lembaga kursus. Umumnya, masyarakat akan mencari lembaga kursus yang fasilitasnya memadai, kualitas pengajarnya yang bagus, harga terjangkau dan sesuai dengan kebutuhan. Untuk memudahkan masyarakat mendapatkan informasi tentang lembaga kursus yang diinginkan, maka pada penulisan ini dibuat aplikasi pencarian lokasi lembaga kursus menggunakan *smartphone* berbasis Android. *Smartphone* berbasis Android dipilih karena saat ini semakin banyak penggunanya.

Aplikasi ini dibuat untuk memudahkan masyarakat mengetahui lokasi lembaga kursus serta memudahkan masarakat untuk mendapatkan informasi lengkap mengenai lembaga kursus yang dicari. Aplikasi dibangun dengan Android Studio serta bahasa pemograman Java dengan *operating system* (OS) Android. OS Android sangat fleksibel dan dapat digunakan pada berbagai *platform hardware* dan mudah penggunaannya. Aplikasi ini memanfaatkan layanan google map yang diakses melalui internet dan berisi beberapa menu yaitu pencarian lokasi berdasarkan nama lembaga kursus, pencarian lokasi berdasarkan kategori lembaga kursus, tentang dan keluar.

Kata Kunci: Sistem Informasi Geografis, Pencarian Lokasi Lembaga Kursus, Android, *Smartphone*

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas berkat dan rahmat-Nya penulis dapat menyelesaikan Skripsi ini dengan Judul “Aplikasi Sistem Informasi Geografis Pencarian Lokasi Lembaga Kursus berbasis Android di Kota Gorontalo”, sesuai dengan yang direncanakan. Skripsi ini dibuat untuk memenuhi salah satu syarat untuk memperoleh gelar Sarjana. Penulis menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, Skripsi ini tidak dapat penulis selesaikan. Oleh karena itu penulis menyampaikan terima kasih kepada :

1. Ibu Dr. Hj. Juriko Abdussamad, M.Si, selaku Ketua Yayasan Pengembangan Ilmu Pengetahuan dan Teknologi (YPIPT) Ichsan Gorontalo.
2. Bapak Dr. Abd. Gaffar Latjokke, M.Si, selaku Rektor Universitas Ichsan Gorontalo.
3. Ibu Zohrahayaty, M.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.
4. Ibu Asmaul Husna, M.Kom, selaku Wakil Dekan I Bidang Akademik,
5. Ibu Irma Surya Kumala Idris, M.Kom , selaku Wakil Dekan II Bidang Administrasi Umum dan Keuangan serta Pembimbing Pendamping yang telah membimbing penulis dalam mengerjakan Skripsi ini.
6. Bapak Yasin Aril Mustofa, M. Kom, selaku Wakil Dekan III Bidang Kemahasiswaan.
7. Bapak Irvan Abraham Salihi, M.Kom, selaku Ketua Jurusan Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo dan Pembimbing Utama yang telah membimbing penulis dalam mengerjakan Skripsi ini.
8. Bapak Ibu Dosen yang telah mendidik dan membimbing penulis dalam mengerjakan Skripsi ini.
9. Pihak Dinas Pendidikan Kota Gorontalo Bidang Pendidikan Non Formal dan Informal yang telah mengizinkan penulis untuk melakukan penelitian dan pengambilan data.

10. Brother dan Sister di MY LIBERTY IES Gorontalo yang telah mengizinkan penulis melaksanakan Kuliah Kerja Lapangan Pengabdian (KKLP) selama 2 bulan.
11. Ucapan terima kasih yang sebesar-besarnya kepada kedua orang tua Ibu Dra. Hj. Salmawati Usuli dan Bapak Abd. Rahman Uloli, S.Sos, atas segala doa, jerih payah, cinta, kasih sayang serta dukungan moril yang sangat besar kepada penulis.
12. Adik Muh. Zulkifli Uloli, Rugaya Ade Putri Uloli, Tante Maimun Usuli, M.Si serta seluruh keluarga besar yang telah memberikan banyak cinta, kasih sayang, dukungan dan do'a sehingga penulis dapat menyelesaikan Skripsi ini.
13. Sunarti Adam, Fifi Deswari F. Junus, sahabat-sahabat tercinta yang selalu men-*support* dan memotivasi sehingga penulis dapat menyelesaikan Skripsi ini.
14. Semua pihak yang ikut membantu dalam penyelesaian Skripsi ini yang tak sempat penulis sebutkan satu-persatu.

Semoga mendapatkan imbalan yang lebih besar dari Allah SWT, melebihi apa yang beliau-beliau berikan kepada penulis.

Gorontalo, April 2017

Penulis

## DAFTAR ISI

HALAMAN SAMPUL .....	i
HALAMAN JUDUL.....	ii
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN.....	iv
HALAMAN PERNYATAAN .....	v
<i>ABSTRACT</i> (Inggris).....	vi
ABSTRAK (Indonesia).....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR .....	xv
DAFTAR TABEL.....	xix
BAB I. PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Identifikasi Masalah .....	3
1.3. Rumusan Masalah .....	4
1.4. Batasan Masalah.....	4
1.5. Tujuan Penelitian.....	4
1.6. Manfaat Penelitian.....	4
BAB II. TINJAUAN PUSTAKA.....	6
2.1. Tinjauan Studi .....	6
2.2. Tinjauan Teori.....	8
2.2.1. Lembaga Kursus .....	8
2.2.2. Pengertian Sistem Informasi Geografis .....	12
2.2.3. Konsep Sistem Informasi Geografis .....	12
2.2.4. Komponen Utama Sistem Informasi Geografis.....	14
2.2.5. Data Sistem Informasi Geografis .....	16
2.2.6. Perangkat dan Aplikasi Sistem Informasi Geografis .....	17
2.2.7. Sistem Informasi Geografis Berbasis <i>Mobile</i> .....	18

2.2.8. Google Maps.....	21
2.2.8.1. Cara Kerja Google Maps .....	21
2.2.8.2. Macam-macam Tampilan Google Maps .....	22
2.2.9. Google Maps API .....	23
2.3. Java .....	24
2.3.1. <i>Java Runtime Environment (JRE)</i> .....	26
2.3.2. <i>Java Development Kit (JDK)</i> .....	26
2.3.3. <i>Android Studio</i> .....	26
2.3.4. <i>Framework</i> .....	26
2.3.5. <i>Spring</i> .....	27
2.3.6. <i>Intergrated Development Environment (IDE)</i> .....	28
2.3.7. <i>Netbeans</i> .....	29
2.4. Basis Data ( <i>Database</i> ) .....	29
2.5. <i>Web-server</i> .....	31
2.6. <i>Client-server</i> .....	31
2.6.1. <i>Client</i> .....	32
2.6.2. <i>Server</i> .....	32
2.6.3. Cara Kerja <i>Client-Sever</i> .....	33
2.7. Android .....	33
2.7.1. Versi-versi Android .....	36
2.8. Siklus Hidup Pengembangan Sistem .....	37
2.8.1. Analisis Sistem .....	39
2.8.2. Desain Sistem .....	40
2.8.3. Implementasi Sistem.....	48
2.8.4. Pemeliharaan Sistem.....	49
2.9. Teknik Pengujian Sistem.....	52
2.9.1. <i>White Box Testing</i> .....	52
2.9.2. <i>Black Box Testing</i> .....	56
2.10. Perangkat Lunak Pendukung.....	57
2.11. Kerangka Pemikiran.....	59
BAB III. OBJEK DAN METODE PENELITIAN .....	60

3.1. Objek Penelitian .....	60
3.2. Metode Penelitian.....	60
3.2.1. Tahap Pengumpulan Data.....	60
3.2.2. Tahap Analisis Sistem .....	61
3.2.3. Tahap Desain Sistem .....	62
3.2.4. Tahap Konstruksi Sistem.....	62
3.2.5. Tahap Pengujian Sistem .....	62
3.2.6. Tahap Implementasi Sistem.....	63
3.2.7. Tahap Pemeliharaan Sistem.....	63
BAB IV. ANALISA DAN DESAIN SISTEM .....	64
4.1 Analisis Sistem .....	64
4.1.1 Sistem Yang Diusulkan .....	64
4.2 Desain Sistem .....	65
4.2.1 Desain Sistem Secara Umum .....	65
4.2.1.1 Diagram Konteks .....	65
4.2.1.2 Diagram Berjenjang.....	65
4.2.1.3 Diagram Arus Data (DAD).....	66
4.2.1.3.1 DAD Level 0 .....	66
4.2.1.3.1 DAD Level 0 .....	67
4.2.1.4 Kamus Data .....	67
4.2.1.5 Desain <i>Input</i> Secara Umum .....	72
4.2.1.6 Desain <i>Database</i> Secara Umum .....	73
4.2.1.7 Desain <i>Output</i> Secara Umum .....	73
4.2.2 Desain Sistem Secara Terinci .....	74
4.2.2.1 Desain <i>Input</i> Terinci .....	74
4.2.2.2 Desain <i>Database</i> Terinci .....	80
4.2.2.3 Desain <i>Ouput</i> Terinci.....	83
4.2.3 Desain Relasi Antar Tabel.....	88
4.2.4 Desain Menu Utama .....	89
BAB V. HASIL PENELITIAN DAN PEMBAHASAN .....	90
5.1 Hasil Penelitian.....	90

5.1.1 Gambaran Umum Lokasi Penelitian.....	90
5.1.1.1 Dinas Pendidikan Nasional Kota Gorontalo.....	90
5.1.1.2 Struktur Organisasi .....	94
5.1.1.3 Job Deskripsi .....	95
5.1.2 Hasil Pengujian Sistem.....	104
5.1.2.1 Pengujian <i>White Box</i> .....	104
5.1.2.2 Pengujian <i>Black Box</i> .....	108
5.2 Pembahasan .....	119
5.2.1 Deskripsi Kebutuhan Hardware/Software .....	119
5.2.2 Langkah-langkah Menjalankan Sistem .....	120
5.2.2.1 Tampilan Halaman Login .....	120
5.2.2.2 Tampilan Home .....	121
5.2.2.3 Tampilan Daftar Lembaga Kursus .....	122
5.2.2.4 Tampilan Form Tambah Data Lembaga Kursus .....	123
5.2.2.5 Tampilan Daftar Operator .....	124
5.2.2.6 Tampilan Form Tambah Operator .....	124
5.2.2.7 Tampilan Form Tambah Kategori .....	126
5.2.2.8 Tampilan Daftar Kategori.....	126
5.2.2.9 Tampilan Form Tambah Pengajar .....	127
5.2.2.10 Tampilan Daftar Pengajar.....	127
5.2.2.11 Tampilan Form Tambah Fasilitas.....	128
5.2.2.12 Tampilan Daftar Fasilitas .....	128
5.2.2.13 Tampilan Form Upload Album (Foto) .....	129
5.2.2.14 Tampilan Album.....	129
5.2.2.15 Tampilan Form Upload Brosur.....	130
5.2.2.16 Tampilan Brosur .....	131
5.2.2.17 Tampilan Aplikasi SIGOPLEK .....	131
5.2.2.18 Tampilan Profil dan Lokasi Lembaga Kursus.....	134
BAB VI. KESIMPULAN DAN SARAN .....	140
6.1 Kesimpulan.....	140
6.2 Saran .....	140

**DAFTAR PUSTAKA**

**LAMPIRAN**

Listing Program

Rekomendasi Hasil Penelitian Dari Tempat Penelitian

Riwayat Hidup

## DAFTAR GAMBAR

Gambar 2.1 Arsitektur <i>Mobile SIG</i> .....	20
Gambar 2.2 Tampilan Map Normal .....	22
Gambar 2.3 Tampilan Map Hybrid.....	22
Gambar 2.4 Tampilan Map Terrain .....	22
Gambar 2.5 Tampilan Map Satelit.....	23
Gambar 2.6 Logo Android .....	35
Gambar 2.7 Logo Android Versi 7.0 (Nougat) .....	37
Gambar 2.8 Siklus Hidup Pengembangan Sistem.....	48
Gambar 2.9 Notasi Kesatuan Luar di DAD .....	47
Gambar 2.10 Notasi Arus Data di DAD .....	47
Gambar 2.11 Notasi Proses di DAD .....	47
Gambar 2.12 Notasi Simpanan Data di DAD .....	48
Gambar 2.13 Bagan Alir .....	53
Gambar 2.14 Grafik Alir .....	54
Gambar 4.1 Bagan Alir Sistem Yang Diusulkan .....	64
Gambar 4.2 Diagram Konteks.....	65
Gambar 4.3 Diagram Berjenjang .....	65
Gambar 4.4 DAD Level 0 .....	66
Gambar 4.5 DAD Level 1 .....	67
Gambar 4.6 Desain Input Data Lembaga Kursus.....	74
Gambar 4.7 Desain Input Data Operator.....	75
Gambar 4.8 Desain Input Data Pengajar.....	75
Gambar 4.9 Desain Input Data Kategori.....	76
Gambar 4.10 Desain Input Data Fasilitas .....	76
Gambar 4.11 Desain Input Tambah Album .....	77
Gambar 4.12 Desain Input Upload Brosur.....	77
Gambar 4.13 Desain Input Edit Profil Lembaga Kursus .....	78
Gambar 4.14 Desain Pencarian Lokasi Berdasarkan Nama.....	79
Gambar 4.15 Desain Pencarian Lokasi Berdasarkan Kategori .....	79

Gambar 4.16 Desain Input Pencarian Rute Lembaga Kursus .....	80
Gambar 4.17 Desain Output Daftar Lembaga Kursus .....	83
Gambar 4.18 Desain Output Daftar Operator .....	84
Gambar 4.19 Desain Output Daftar Kategori .....	84
Gambar 4.20 Desain Output Daftar Fasilitas .....	85
Gambar 4.21 Desain Output Daftar Pengajar .....	85
Gambar 4.22 Desain Output Daftar Album .....	85
Gambar 4.23 Desain Output Foto Brosur .....	86
Gambar 4.24 Desain Output Tampilan Awal Aplikasi <i>Mobile</i> .....	86
Gambar 4.25 Desain Output Menu .....	87
Gambar 4.26 Desain Output Profil Lembaga Kursus .....	87
Gambar 4.27 Desain Output Rute Lembaga Kursus .....	88
Gambar 4.28 Relasi Antar Tabel.....	88
Gambar 4.29 Desain Menu Utama.....	89
Gambar 5.1 Struktur Organisasi Dinas Pendidikan Kota Gorontalo .....	94
Gambar 5.2 Flowgraph Proses Pencarian Lokasi .....	106
Gambar 5.3 Tampilan Form Login .....	108
Gambar 5.4 Tampilan Home.....	108
Gambar 5.5 Tampilan Daftar Lembaga Kursus .....	109
Gambar 5.6 Form Input Data Lembaga Kursus .....	109
Gambar 5.7 Tampilan Pesan Tersimpan Data Lembaga Kursus .....	110
Gambar 5.8 Tampilan Daftar Operator .....	110
Gambar 5.9 Form Input Data Operator .....	111
Gambar 5.10 Tampilan Daftar Kategori .....	111
Gambar 5.11 Form Tambah Kategori .....	111
Gambar 5.12 Tampilan Pesan Tersimpan Data Kategori.....	112
Gambar 5.13 Tampilan Daftar Pengajar .....	112
Gambar 5.14 Form Tambah Pengajar .....	113
Gambar 5.15 Tampilan Pesan Tersimpan Data Pengajar.....	113
Gambar 5.16 Tampilan Daftar Fasilitas .....	114
Gambar 5.17 Form Tambah Fasilitas.....	114

Gambar 5.18 Tampilan Pesan Tersimpan Data Fasilitas .....	114
Gambar 5.19 Tampilan Daftar Album Foto.....	115
Gambar 5.20 Form Upload Foto .....	115
Gambar 5.21 Tampilan Pesan Foto Tersimpan.....	116
Gambar 5.22 Tampilan Foto Brosur .....	116
Gambar 5.23 Form Upload Brosur .....	117
Gambar 5.24 Tampilan Pesan Brosur Tersimpan .....	117
Gambar 5.25 Pencarian Lokasi Lembaga Kursus .....	118
Gambar 5.26 Form Login.....	120
Gambar 5.27 Tampilan Home Admin (Bid. PNFI).....	121
Gambar 5.28 Tampilan Home Operator (Lembaga Kursus).....	121
Gambar 5.29 Tampilan Daftar Lembaga Kursus .....	122
Gambar 5.30 Tampilan Form Tambah Lembaga Kursus.....	123
Gambar 5.31 Tampilan Daftar Operator .....	124
Gambar 5.32 Tampilan Form Tambah Operator.....	124
Gambar 5.33 Tampilan Form Tambah Operator.....	125
Gambar 5.34 Tampilan Form Tambah Kategori.....	126
Gambar 5.35 Tampilan Form Tambah Kategori.....	126
Gambar 5.36 Tampilan Daftar Kategori .....	126
Gambar 5.37 Tampilan Form Tambah Pengajar.....	127
Gambar 5.38 Tampilan Daftar Pengajar .....	127
Gambar 5.39 Tampilan Form Tambah Fasilitas .....	128
Gambar 5.40 Tampilan Daftar Fasilitas .....	128
Gambar 5.41 Tampilan Form Upload Foto.....	129
Gambar 5.42 Tampilan Album .....	129
Gambar 5.43 Tampilan Form Upload Brosur .....	130
Gambar 5.44 Tampilan Brosur.....	131
Gambar 5.45 File SIGOPLEK.apk .....	132
Gambar 5.46 Install SIGOPLEK.....	132
Gambar 5.47 Proses Install SIGOPLEK .....	133
Gambar 5.48 Proses Install SIGOPLEK selesai .....	133

Gambar 5.49 Tampilan SIGOPLEK .....	134
Gambar 5.50 Tampilan Menu SIGOPLEK .....	135
Gambar 5.51 Pencarian Lokasi Lembaga Kursus Berdasarkan Nama .....	135
Gambar 5.52 Profil Lembaga Kursus.....	136
Gambar 5.53 Ikon Peta.....	136
Gambar 5.54 Tampilan Rute Menuju Lokasi Lembaga Kursus .....	137
Gambar 5.55 Pencarian Lokasi Lembaga Kursus Berdasarkan Kategori .....	137
Gambar 5.56 Menu Rute .....	138
Gambar 5.57 Tampilan About.....	139
Gambar 5.58 Menu Keluar.....	139

## DAFTAR TABEL

Tabel 1.1 Jumlah Lembaga Kursus di Kota Gorontalo.....	2
Tabel 2.1 Daftar Lembaga Kursus di Kota Gorontalo .....	10
Tabel 2.2 Macam-macam tampilan Google Maps .....	24
Tabel 2.3 Versi-versi Android .....	36
Tabel 2.4 Simbol Bagan Alir Sistem.....	44
Tabel 2.5 Perangkat Lunak Pendukung .....	57
Tabel 4.1 Kamus Data Lembaga Kursus.....	68
Tabel 4.2 Kamus Data Operator.....	69
Tabel 4.3 Kamus Data Pengajar.....	69
Tabel 4.4 Kamus Data Kategori.....	70
Tabel 4.5 Kamus Data Fasilitas .....	70
Tabel 4.6 Kamus Data Album.....	71
Tabel 4.7 Kamus Data Brosur .....	71
Tabel 4.8 Desain Input Secara Umum .....	72
Tabel 4.9 Desain Database Secara Umum .....	73
Tabel 4.10 Desain Output Secara Umum.....	73
Tabel 4.11 Tabel Data Lembaga Kursus.....	80
Tabel 4.12 Tabel Data Operator.....	81
Tabel 4.13 Tabel Data Pengajar .....	81
Tabel 4.14 Tabel Kategori.....	82
Tabel 4.15 Tabel Fasilitas .....	82
Tabel 4.16 Tabel Album .....	82
Tabel 4.17 Tabel Brosur.....	83
Tabel 5.1 Tabel Pengujian <i>Black Box</i> Form Login .....	108
Tabel 5.2 Tabel Pengujian <i>Black Box</i> Menu Lembaga.....	110
Tabel 5.3 Tabel Pengujian <i>Black Box</i> Menu Operator.....	111
Tabel 5.4 Tabel Pengujian <i>Black Box</i> Menu Kategori .....	112
Tabel 5.5 Tabel Pengujian <i>Black Box</i> Menu Pengajar.....	113

Tabel 5.6 Tabel Pengujian <i>Black Box</i> Menu Fasilitas.....	115
Tabel 5.7 Tabel Pengujian <i>Black Box</i> Menu Album.....	116
Tabel 5.8 Tabel Pengujian <i>Black Box</i> Menu Brosur .....	117
Tabel 5.9 Tabel Pengujian <i>Black Box</i> Pencarian Lokasi Lembaga Kursus .....	118

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Lembaga kursus adalah lembaga non formal yang memberikan pelajaran, pengetahuan, pelatihan atau keterampilan tertentu yang diberikan dalam waktu singkat. Di lembaga kursus, terdapat para pengajar atau instruktur yang handal serta ahli dibidangnya untuk dapat membantu proses pengembangan keterampilan. Lembaga-lembaga kursus ini menyediakan berbagai jasa pelatihan keterampilan, seperti pengembangan diri, pelajaran bahasa asing, bimbingan belajar, mengemudi, menjahit dan masih banyak lagi.

Masyarakat tentu ingin memilih lembaga kursus yang berkompeten dibidangnya dengan metode pembelajaran yang berkualitas serta biaya yang terjangkau. Berdasarkan pertimbangan tersebut, masyarakat akan memutuskan lembaga kursus mana yang akan mereka pilih sesuai dengan kebutuhan. Saat ini, sudah banyak lembaga kursus yang tersedia, terutama yang tersebar di wilayah Kota Gorontalo. Namun yang menjadi kendala adalah informasi. Baik itu informasi tentang lokasi, metode pembelajaran serta biaya kursus.

Biasanya informasi seperti ini hanya bisa didapatkan melalui brosur / pamphlet yang dibagikan atau dengan mengunjungi langsung lokasi lembaga kursus tersebut. Masyarakat sebagai calon peserta kursus, tentunya ingin mendapatkan informasi lengkap dengan cepat. Sedangkan akses informasi tercepat saat ini adalah melalui *internet*.

Seiring berkembangnya teknologi, pengguna *internet* terus meningkat dan berasal dari berbagai kalangan usia. Hal ini menunjukkan bahwa *internet* memberikan kemudahan dalam hal akses informasi. *Smartphone* dengan sistem operasi Android pun saat ini makin diminati dan penggunanya terus meningkat, ini dikarenakan *smartphone* menawarkan kemudahan bagi para penggunanya (*user friendly*) dalam menunjang aktifitas sehari-hari. Sehingga dapat disimpulkan bahwa *internet* dan *smartphone* bukanlah hal yang baru lagi bagi masyarakat luas.

Berdasarkan permasalahan tersebut, Sistem Informasi Geografis (SIG) dinilai cocok untuk menyelesaiannya. Aplikasi SIG ini akan didukung dengan *Google Maps API* dan dikemas dalam bentuk aplikasi yang berbasis Android. Aplikasi ini nantinya dapat diakses pada *smartphone* dengan sistem operasi Android minimal versi 4.0.3 (Ice Cream Sandwich) sampai dengan versi 7.0 (Nougat).

**Tabel 1.1** Jumlah Lembaga Kursus di Kota Gorontalo

Kecamatan	Kategori Kursus	Mengemudi	Komputer	Bahasa Inggris	Bimbel	Public Speaking	Menjahit
Utara	-	-	-	-	-	-	1
Selatan	2	5	2	1	1	1	2
Timur	-	-	1	-	-	-	2
Barat	-	1	1	-	-	-	1
Tengah	-	1	1	1	1	1	-
Dungingi	-	2	-	-	-	-	-
Hulonthalangi	-	-	-	-	-	-	2
Total	2	9	5	2	2	2	8

**Sumber:** Data Unit PNFI (Pendidikan Non Formal Informal) Dinas Pendidikan Kota Gorontalo

Sistem yang akan dibangun ini akan mencari berdasarkan kategori atau nama lembaga kursus kemudian menampilkan informasi lengkap mengenai lembaga kursus tersebut (profil, foto lokasi, titik lokasi, metode pembelajaran, fasilitas yang disediakan dan biaya kursus) sehingga memudahkan masyarakat untuk mendapatkan informasi lengkap dan cepat.

Dengan adanya aplikasi ini, diharapkan dapat memudahkan masyarakat dalam mengakses informasi tentang lembaga kursus yang ada di wilayah Kota Gorontalo serta memudahkan pihak Dinas Pendidikan Nasional Kota Gorontalo dalam hal pendataan lembaga-lembaga kursus yang ada di wilayah Kota Gorontalo.

Adapun judul dari penelitian yang penulis usulkan adalah **“Aplikasi Sistem Informasi Geografis Pencarian Lokasi Lembaga Kursus Berbasis Android Di Kota Gorontalo”**.

## **1.2 Identifikasi Masalah**

Berdasarkan permasalahan di atas dapat diketahui bahwa:

1. Masyarakat kesulitan mendapatkan informasi mengenai lokasi lembaga kursus secara lengkap dan cepat.
2. Dibutuhkan aplikasi yang memberikan informasi tentang rute serta informasi lengkap mengenai lembaga kursus.

### **1.3 Rumusan Masalah**

Rumusan masalah dalam penelitian ini adalah:

1. Bagaimana merekayasa Sistem Informasi Geografis untuk pencarian lokasi lembaga kursus yang ada di wilayah Kota Gorontalo menggunakan Google Maps API?
2. Bagaimana hasil penerapan Google Maps API berbasis sistem operasi Android ke dalam Sistem Informasi Geografis untuk pencarian lokasi lembaga kursus di wilayah Kota Gorontalo?

### **1.4 Batasan Masalah**

Sistem yang akan dibangun nanti, dibatasi untuk kategori kursus bahasa Inggris, bimbingan belajar, mengemudi, komputer, menjahit dan *public speaking*.

### **1.5 Tujuan Penelitian**

1. Untuk memudahkan pencarian lokasi lembaga kursus di Kota Gorontalo serta menampilkan informasi lengkap tentang lembaga kursus tersebut.
2. Memanfaatkan fasilitas Google Maps API untuk membangun aplikasi SIG Pencarian Lokasi Lembaga Kursus.

### **1.6 Manfaat Penelitian**

Manfaat yang dapat diraih dalam penelitian ini antara lain:

- a. Pengembangan Ilmu

Penelitian ini diharapkan dapat mengembangkan ilmu pengetahuan dibidang teknologi komputer pada umumnya dan dalam pembuatan aplikasi Sistem Informasi Geografis untuk memetakan suatu lokasi berbasis Android.

b. Praktisi

Sebagai bahan masukan (*input source*) bagi semua elemen-elemen ataupun unsur-unsur yang terlibat dalam perancangan aplikasi Sistem Informasi Geografis untuk memetakan lokasi tempat kursus berbasis Android.

c. Peneliti

Sebagai bahan masukkan kepada peneliti berikutnya yang akan meneliti masalah tersebut.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Tinjauan Studi**

Penelitian ini menggunakan beberapa referensi yang berhubungan dengan obyek penelitian terutama dari penelitian sebelumnya :

1. Penelitian yang dilakukan oleh Fadhoelor Rohman, Agung Budi Cahyono (2013) *Mobile GIS Fasilitas Umum Untuk Pengguna Jalan Berbasis Android.* Aplikasi SIG ini didesain dan dikembangkan dengan bahasa pemrograman Java. Aplikasi ini dapat diakses pada *smartphone* dengan sistem operasi Android minimal versi 2.3 (Gingerbread) dan dapat menampilkan fasilitas umum yaitu sebanyak 35 SPBU, 11 Kantor Polisi dan 9 Puskesmas / Rumah Sakit.
2. Penelitian yang dilakukan oleh Juwairiah, Ial Irwan Arahman, Budi Santosa (2013) *Aplikasi Mobile GIS Layanan Lokasi Penting Kota Surakarta Berbasis Android.* Aplikasi ini dibangun dengan metodologi penelitian berupa metode GRAPPLE. Fitur-fitur pada aplikasi ini yaitu dapat mengetahui daftar lokasi berdasarkan kategori, koordinat lokasi yang dituju, koordinat lokasi *user*, informasi (gambar, nama tempat, alamat, nomor telepon, *web* lokasi, deskripsi, jarak lokasi berdasarkan posisi *user* dengan lokasi yang dituju) dan petunjuk arah lokasi. Kategori lokasi penting pada sistem meliputi Swalayan, Hotel, Rumah Sakit, Masjid, Gereja, Kuliner, Stadion, SPBU, Pasar, Perguruan Tinggi, Toko Batik,

Stasiun Kereta Api, Terminal Bus, Bandara dan lokasi lain. Aplikasi ini dibangun menggunakan Google Map API untuk menampilkan peta, SDK Android, Eclipse sebagai editor, Java sebagai bahasa pemrograman untuk membuat aplikasi *client*, PHP sebagai bahasa pemrograman untuk membuat aplikasi *server*, MySQL sebagai *database* dan ponsel Android sebagai alat pendukung.

3. Penelitian yang dilakukan oleh Rastuti, Leon Andretti Abdillah, Eka Puji Agustini (2015) *Sistem Informasi Geografis Potensi Wilayah Kabupaten Banyuasin Berbasis Web*. Sistem ini dibangun dengan menggunakan model *waterfall* dan berisi tentang informasi Potensi Wilayah yang ada di Kabupaten Banyuasin yang bertujuan untuk memudahkan para pengguna dalam mengakses informasi tentang letak potensi wilayah dibidang pertanian, perkebunan dan perindustrian yang ada di Kabupaten Banyuasin.
4. Penelitian yang dilakukan oleh Bambang Yuwono, Agus Sasmito Aribowo, Febri Arif Setyawan (2015) *Sistem Informasi Geografis Berbasis Android Untuk Pariwisata Di Daerah Magelang*. Aplikasi ini berbasis *mobile* dengan mengintegrasikan *Global Positioning System* (GPS). Aplikasi ini terhubung langsung dengan *Google Maps*. Tujuan sistem ini dibangun untuk memberikan informasi lokasi pariwisata di daerah Magelang dengan mudah dan sederhana karena dapat dioperasikan dimanapun *user* berada menggunakan piranti *mobile device* berbasis Android. Pengguna sistem ini dibagi menjadi dua yaitu *admin* dan *user*

umum. Pengguna *admin* dapat melakukan olah data yang meliputi edit, hapus dan tambah data terkait dengan wisata, kuliner, penginapan, serta lokasinya. Aplikasi ini dapat berjalan pada *smartphone* Android 2.2 (Froyo) dan versi yang berada di atasnya.

5. Penelitian yang dilakukan oleh Meides Palabiran, Dedi Cahyadi, Zainal Arifin (2015) *Sistem Informasi Geografis Kuliner, Seni Dan Budaya Kota Balikpapan Berbasis Android*. Sistem ini berintegrasi dengan Google Maps API, SQLite sebagai *database* dan dibangun menggunakan *Eclipse Juno* sebagai IDE. Aplikasi ini berisi informasi tentang Kuliner, Seni dan Budaya yang ada di Kota Balikpapan, tujuannya adalah menunjukkan lokasi Kuliner, Seni dan Budaya yang ada disekitar pengguna berada.

## **2.2 Tinjauan Teori**

### **2.2.1 Lembaga Kursus**

Lembaga kursus merupakan Satuan Pendidikan Pendidikan Luar Sekolah (Nonformal) yang diselenggarakan bagi warga masyarakat yang memerlukan bekal untuk mengembangkan diri, bekerja mencari nafkah, dan atau melanjutkan ke tingkat atau jenjang pendidikan yang lebih tinggi.

Setiap pendirian Lembaga Kursus, harus memiliki izin dari Pemerintah seperti yang diatur dalam UU Nomor 20 Tahun 2003 Ayat (1) Setiap satuan pendidikan formal dan nonformal yang didirikan wajib memperoleh izin Pemerintah atau Pemerintah Daerah, Ayat (2) Syarat-syarat untuk memperoleh izin meliputi isi pendidikan, sarana dan prasarana pendidikan, pembiayaan

pendidikan, sistem evaluasi dan sertifikasi serta manajemen dan proses pendidikan. Pengaturan perizinan lembaga kursus dilakukan dengan tujuan:

- a. Memudahkan Pemerintah atau Pemerintah Daerah dalam mengadakan pembinaan yang mencakup perencanaan, pelaksanaan, penilaian, dan evaluasi, serta pengawasan secara tertib, teratur dan terarah terhadap setiap jenis kursus dan pelatihan;
- b. Memelihara dan meningkatkan mutu pendidikan yang serasi dengan tuntutan perkembangan ilmu pengetahuan dan teknologi, kebutuhan masyarakat, dan dunia usaha/industri;
- c. Mengarahkan, menyerasikan, dan mengembangkan program pendidikan nonformal guna menunjang suksesnya program pembangunan;
- d. Melindungi lembaga kursus dan pelatihan dari tindakan di luar peraturan perundang-undangan yang berlaku;
- e. Melindungi warga masyarakat dari penyalahgunaan penyelenggararaan kursus dan pelatihan yang mengakibatkan kerugian;
- f. Memberikan tanggung jawab hukum kepada lembaga kursus dan pelatihan.

Standar minimal (kelayakan) membuka Lembaga Kursus adalah:

- a. Isi pendidikan, meliputi: struktur kurikulum yang berbasis kompetensi dan berorientasi pada keunggulan lokal dan bahan ajar berupa buku/modul bahan ajar;

- b. Pendidik dan Tenaga Kependidikan, meliputi: jumlah, kualifikasi, dan kompetensi masing-masing pendidik dan tenaga kependidikan yang sesuai dengan bidangnya;
- c. Sarana dan prasarana, meliputi ketersediaan ruang kantor, ruang belajar teori, ruang praktik, sarana belajar mengajar, dan media pembelajaran, dengan ukuran, jenis dan jumlah yang sesuai;
- d. Pembiayaan, meliputi biaya operasional dan biaya personal untuk mendukung terselenggaranya program pendidikan;
- e. Manajemen meliputi struktur organisasi lembaga dan deskripsi tugas yang jelas dan terarah guna memudahkan jalannya kegiatan dalam pencapaian tujuan; dan
- f. Proses pendidikan, meliputi: Silabus dan Rencana Pelaksanaan Pembelajaran (RPP).

Lembaga kursus yang tersedia saat ini meliputi Kursus Komputer, Menjahit, Bimbingan Belajar, Bahasa Asing, Mengemudi, *Public Speaking*, dll.

Berikut ini alamat lembaga kursus di Kota Gorontalo:

**Tabel 2.1** Daftar Lembaga Kursus di Kota Gorontalo

N o	Nama	Alamat	Kategori Kursus
1	Nusa Prima Computer Center	Jl. Prof. Dr. HB Jassin Kel. Paguyaman Kec. Kota Tengah	Komputer
2	MY LIBERTY International English School Gorontalo	Jl. HOS Cokroaminoto Kel. Limba U1 Kec. Kota Selatan	Bahasa Inggris
3	LKP Pelangi Driving School	Jl. Jendral Sudirman No.88 Kec. Kota Selatan	Mengemudi

4	Kids 2 Success English Course Center	Jl. Budi Utomo Kel. Limba U1 Kec. Kota Selatan	Bahasa Inggris
5	Mallcom Cendekia Gorontalo	Jl. Prof. Dr. HB Jassin Kel. Limba U2 Kec. Kota Selatan	Komputer & Bahasa Inggris
6	Briliant Brain	Jl. Dewi Sartika No.45 Kel. Wumialo Kec. Tengah	Bimbel & Public Speaking
7	Q Math	Jl. Sultan Botutihe No. 155 Kota Timur	Bimbel
8	Yayasan Taman Cendekia	Jl. Nani Wartabone Kel. Limba U1 Kec. Kota Selatan	Bimbel
9	LKP Handayani	Jl. Prof. Dr. HB Jassin Kel. Limba U2 Kec. Kota Selatan	Mengemudi
10	Kharisma Gorontalo	Jl. Jamaludin Malik Kel. Limba U1 Kec. Kota Selatan	Komputer
11	Intensive English Course Gorontalo	Jl. Irian Kel. Liliwo Kec. Kota Tengah	Bahasa Inggris
12	LKP Bina Taruna	Jl. Jaksa Agung Suprapto Kel. Limba U2 Kec. Kota Selatan	Komputer
13	LKP Melati	Jl. Ki Hajar Dewantoro Kel. Limba U2 Kec. Kota Selatan	Menjahit
14	LKP Miftahul Jannah	Jl. Prof. DR.HB. Yassin No.61	Menjahit
15	LKP P3KP Al Adha	Jl. Nani Wartabone No.233	Komputer
16	LPK Tinelo	Jl. Jalaludin Tantu	Menjahit
17	LPK Sunarti	Jl. Komite 12 Kel. Ipilo	Menjahit
18	LPK Sonny Cinere	Jl. Panca Wardana	Menjahit
19	LPK Andin	Jl. Lokakarya Kel. Wongkaditi Barat Kec. Kota Utara	Menjahit
20	LPK Fajrinah	Jl. Usman Isa Kel. Dembe I Kec. Kota Barat	Menjahit
21	LPK Smart	Jl. Raja Eyato Kec. Kota Barat	Bahasa Inggris & Komputer

22	LPK Nofer	Jl. Jeruk Kel. Huangobotu Kec. Dungingi	Komputer
23	LKP Karya Baru	Jl. Beringin No.652 Kel.Tomulabutao Kec. Dungingi	Komputer
24	LKP Duta Nusantara	Jl. Beringin No.652 Kel.Tomulabutao Kec. Dungingi	Komputer
25	LPK Maharani	Jl. Gg. Rinjani Kel. Siendeng Kec. Kota Selatan	Menjahit

**Sumber:** Data Unit PNFI (Pendidikan Non Formal Informal) Dinas Pendidikan Kota Gorontalo

### 2.2.2 Pengertian Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) adalah sistem informasi yang mampu menampilkan data dan informasi geospasial. Dalam penerapannya SIG sangat berhubungan dengan proses pemetaan, karena dengan ciri khas SIG yang mampu menampilkan citra geografis dan data spasial suatu lokasi. Manfaat SIG ini juga diutarakan Pugas dkk (2011:2) ” SIG mampu mengakomodasi penyimpanan, pemrosesan, dan penayangan data spasial digital bahkan integrasi data yang beragam, mulai dari citra satelit, foto udara, peta analog bahkan data statistik.”

### 2.2.3 Konsep Sistem Informasi Geografis

SIG (Sistem Informasi Geografis) atau dikenal pula dengan GIS (*Geographical Information System*) merupakan suatu istilah dalam bidang pemetaan yang memiliki ruang lingkup mengenai bagaimana suatu sistem dapat menghubungkan objek geografis dengan informasinya (Eddy, 2009).

Rahmad Husein mendefinisikan SIG (Sistem Informasi Geografis) berdasarkan tiap-tiap kata yaitu :

a. Sistem

Sistem merupakan kumpulan elemen-elemen yang saling berintegrasi dan berketergantungan dalam lingkungan yang dinamis untuk mencapai tujuan tertentu. Pada SIG (Sistem Informasi Geografis) sistem merupakan kumpulan dari informasi, data geospasial dan juga sistem komputer atau perangkat elektronik lainnya.

b. Informasi

Pada representasi permukaan geografis SIG (Sistem Informasi Geografis), terdapat beberapa objek-objek dimana tiap-tiap objek memiliki informasi masing – masing yang unik. Hubungan langsung antara objek dan informasi yang bersifat interaktif membuat peta menjadi *intelligent*.

c. Geografis

SIG (Sistem Informasi Geografis) dibangun berdasarkan geografi atau spasial, di dalamnya terdapat objek-objek yang dapat berupa fisik, budaya, atau ekonomi alamiah. Keterangan spasial dari objek-objek ini ditampilkan pada suatu peta untuk memberikan gambaran yang representatif dari keadaan sebenarnya dimuka bumi.

Dalam artikelnya mengenai SIG (Sistem Informasi Geografis), Rahmad Husein juga menjelaskan beberapa karakteristik SIG (Sistem Informasi Geografis), yaitu :

- a. Merupakan suatu sistem hasil pengembangan perangkat keras dan perangkat lunak untuk tujuan pemetaan, sehingga fakta wilayah dapat disajikan dalam suatu sistem berbasis komputer.

- b. Melibatkan ahli geografi, informatika dan komputer, serta aplikasi terkait.
- c. Masalah dalam pengembangan meliputi : cakupan, kualitas dan standar data, struktur, model dan visualisasi data, koordinasi kelembagaan dan etika, pendidikan, *expert system* dan *decision support system* serta penerapannya.
- d. Perbedaannya dengan sistem informasi lainnya : data dikaitkan dengan letak geografis, dan terdiri dari data textual maupun grafik.
- e. Bukan hanya sekedar merupakan pengubahan peta konvensional (tradisional) ke bentuk peta digital untuk kemudian disajikan (dicetak / diperbanyak) kembali.
- f. Mampu mengumpulkan, menyimpan, mentransformasikan, menampilkan, memanipulasi, memadukan dan menganalisis data spasial dari fenomena geografis wilayah.
- g. Mampu menyimpan data dasar yang dibutuhkan untuk penyelesaian suatu masalah. Contoh : penyelesaian masalah perubahan iklim memerlukan informasi dasar seperti curah hujan, suhu, angin, kondisi awan. Data dasar biasanya dikumpulkan secara berkala dalam jangka yang cukup panjang.

#### **2.2.4 Komponen Utama Sistem Informasi Geografis**

Dalam merancang Sistem Informasi Geografis dibutuhkan 3 komponen utama yaitu sistem komputer, data geospasial serta pengguna. Ketiganya saling berhubungan. Sistem Komputer terdiri dari *hardware* dan *software*. Komponen *software* terdiri dari *program*, *database*, dan *Graphical User Interface* (GUI). Dimana perlu diketahui bahwa bagian GUI merupakan tampilan dari program

yang berinteraksi langsung dengan pengguna. Dalam berkomunikasi dan mendapatkan informasi, GUI menjembatani program kompleks dan kumpulan informasi dalam *database* yang ingin diakses dengan kemampuan seorang pengguna yang awam. Sedangkan *hardware* merupakan perangkat elektronik atau juga dapat disebut dengan *platform* dimana program dan *database* berjalan. *Hardware* dapat berupa komputer atau perangkat-perangkat elektronik bersifat *mobile* seperti alat GPS (*Global Positioning Sistem*), PDA ataupun *Smartphone*.

Data geospasial mengandung rujukan geografi secara langsung seperti *latitude* (Garis Lintang), *longitude* (Garis Bujur), atau sebuah rujukan implisit seperti sebuah alamat, kode pos, dan lain-lain. Pada aplikasi yang kompleks, rujukan geografi mempunyai sebuah proses yang otomatis yang disebut *geocoding*; digunakan untuk menciptakan rujukan geografi eksplisit dari implisit atau gambaran seperti sebuah alamat.

Kumpulan dari data geospasial dihubungkan pada suatu sistem komputer. Sistem ini dapat mengenal informasi yang terkandung pada data geospasial dan mengidentifikasi informasi yang dibutuhkan oleh pengguna. Lalu kebutuhan pengguna dapat disesuaikan dengan data yang tersedia. Maka dapat dimunculkan data geospasial yang berhubungan.

Apabila kita melihat secara perspektif global maka ketiga data tersebut berinteraksi pada suatu sistem yang memiliki aturan atau prosedur tertentu dalam berhubungan satu sama lain.

### 2.2.5 Data Sistem Informasi Geografis

Data Sistem Informasi Geografis pada umumnya dibagi menjadi 4 kelompok, yaitu peta umum (mengenai jalan, jalan raya, batas wilayah, sungai danau, nama-nama tempat); data dan peta lingkungan (mengenai cuaca, lingkungan topografi, sumber daya alam); serta peta rujukan umum (rujukan peta-peta yang bersifat umum seperti peta Dunia dan Negara).

Pada tiap-tiap kelompok data diatas, terdapat sumber yang beragam tempat data didapatkan. Menurut Dhani Gumelar dalam artikelnya mengenai data spasial, data pada peta dapat dihasilkan dari berbagai macam sumber, diantaranya adalah :

1. Citra satelit; satelit dapat merekam kondisi atau gambaran dari permukaan bumi dengan menggunakan sensor / kamera.
2. Peta analog; merupakan bentuk tradisional dari data spasial, dimana data ditampilkan dalam bentuk kertas atau film. Seiring dengan perkembangan teknologi, peta analog dapat disimpan dalam format digital dengan menggunakan alat *scanner*.
3. Foto udara (*Aerial Photographs*); serupa dengan citra satelit, namun pengambilan gambar dilakukan dari pesawat udara.

Data yang digunakan oleh SIG (Sistem Informasi Geografis) sebagai berikut :

- a. Data tabular; berfungsi sebagai atribut bagi data spasial seperti data sensus penduduk, data sosial dan data ekonomi.
- b. Data statistik; metode pengumpulan data periodik pada tempat pengamatan geografis. Misal data curah hujan.

- c. Data tracking; cara pengumpulan data dalam periode tertentu untuk tujuan pemantauan atau pengamatan perubahan, contoh : kebakaran hutan, gunung meletus, debit air sungai.

SIG (Sistem Informasi Geografis) bekerja dengan 2 model, yaitu model Raster dan model Vektor.

- a. Model raster

Model raster adalah bentuk peta yang mengandung kumpulan-kumpulan dari potongan peta berupa *grid* yang dapat merepresentasikan gambar atau bentuk permukaan. Data raster terdiri dari nilai-nilai dalam bentuk digital yang merepresentasikan suatu gambar. Oleh karena itu di butuhkan penandaan atau tag agar kumpulan gambar tersebut dapat di posisikan dengan tepat.

- b. Model vektor

Pada model ini, suatu objek geografis direpresentasikan secara eksplisit dengan dicantumkannya koordinat objek. Terdapat 3 bentuk objek geografis yaitu titik (*point*), garis (*line*) dan area (*polygon*).

### **2.2.6 Perangkat dan Aplikasi Sistem Informasi Geografis**

SIG (Sistem Informasi Geografis) dapat diterapkan pada berbagai macam peralatan atau perangkat. Contohnya adalah menggunakan perangkat *mobile*, seperti *Global Positioning System* (GPS), yang merupakan suatu teknologi yang menggabungkan sistem informasi geografis dengan sistem navigasi yang menggunakan komunikasi satelit. Contoh aplikasi lainnya yang menggunakan perangkat *mobile* adalah dengan *Personal Digital Assistant* (PDA) dan *smartphone*. Pada perangkat ini, implementasi SIG (Sistem Informasi Geografis)

dapat berupa program aplikasi GPS (*Global Positioning Sistem*) atau program aplikasi *mobile* yang akan dijelaskan lebih lanjut.

Untuk aplikasi SIG (Sistem Informasi Geografis) yang berbasis komputer program-program yang ada cukup beragam, yaitu *ESRI*, *mAPIInfo*, *Autodesk*, dan *Mapserver*. Sedangkan untuk aplikasi *database* terdapat beberapa aplikasi yang dapat digunakan sebagai *database* seperti *Oracle Spatial*, *PostgreSQL*, *Informix*, *DB2*, *Ingres*, dan *MySQL*. Komponen SIG *mobile*:

- *Positioning System* : Untuk pelacakan objek bergerak.
- *Mobile GIS Receiver* : Untuk menentukan posisi GPS.
- *Software* : Untuk memproses data atau perintah / instruksi hingga mendapat hasil serta menjembatani antara pengguna dan *hardware*.
- *Data Synchronization and Wireless communication* : Untuk menghubungkan data.
- *Geospatial data* : Berisi informasi lokasi (garis lintang dan garis bujur).
- *GIS content server* : Untuk memberitahu *receiver* satelit mana yang sebaiknya dihubungi (yang terdekat dengan perangkat), tanpa harus mendeteksi seluruh satelit.

### **2.2.7 Sistem Informasi Geografis Berbasis *Mobile***

Sistem Informasi Geografis telah berkembang dari segi keragaman aplikasi dan juga media. Pengembangan aplikasi SIG (Sistem Informasi Geografis) kedepannya mengarah kepada aplikasi berbasis *mobile*. Hal ini disebabkan karena pengembangan aplikasi dilingkungan jaringan telah menunjukkan potensi yang besar dalam kaitannya dengan informasi geografis.

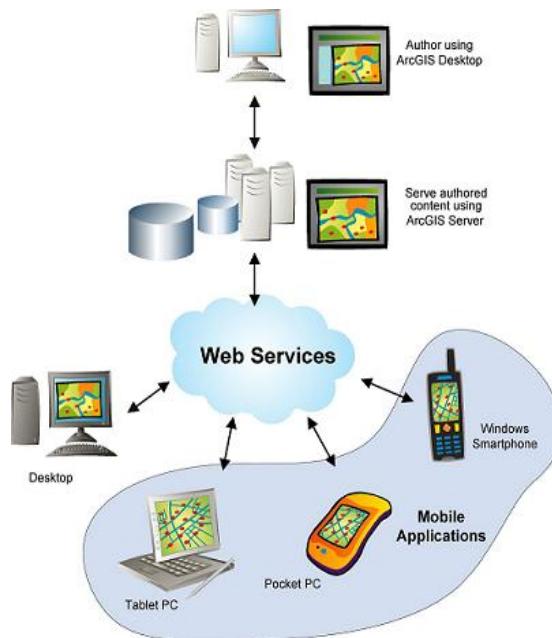
Sebagai contoh adalah adanya peta *online* interaktif sebuah kota, yang memudahkan pengguna dalam mencari informasi geografis terkini yang terdapat pada kota tersebut, tanpa mengenal batas lokasi geografis pengguna.

Pada aplikasi SIG (Sistem Informasi Geografis) berbasis *Mobile*, terdapat beberapa komponen yang saling berinteraksi. Komponen-komponen tersebut bisa saja terdapat pada beberapa lokasi pada jaringan. Oleh karena itu pada SIG berbasis *Mobile*, diperlukan adanya *server*. SIG *Mobile* merupakan sebuah integrasi cara kerja perangkat lunak/keras untuk pengaksesan data dan layanan geospasial melalui perangkat bergerak via jaringan kabel atau nirkabel. Secara umum, SIG *Mobile* diimplementasikan pada dua area aplikasi utama yaitu Layanan Berbasis Lokasi (*Location Based Service*) dan SIG untuk kegiatan lapangan (*Field Based GIS*). Berikut ini hal-hal yang berkenaan dengan aplikasi SIG *mobile* :

1. Diimplementasikan pada perangkat bergerak dengan keterbatasan ruang penyimpanan, memori, dan resolusi.
2. Dapat diimplementasikan secara mandiri (*stand alone*) dengan menyimpan data dalam perangkat bergerak (untuk aplikasi sederhana), atau disesuaikan dengan arsitektur *server* nya (untuk aplikasi SIG *web*).
3. Kemampuan aplikasi SIG *mobile*, seperti:
  1. Menampilkan atau melakukan navigasi.
  2. Mengidentifikasi.
  3. Pencarian atau *query*.
  4. Memodifikasi nilai atribut.

5. Pemberian tanda atau *redline*.
  6. Memodifikasi geometri.
  7. Mengintegrasikan dengan data kantor.
4. Terdapat dua jenis data, yaitu koleksi data (*data collection*) dan navigasi (*navigation*).

Arsitektur dari *Mobile SIG* (Sistem Informasi Geografis) dapat dilihat pada gambar berikut :



**Gambar 2.1** Arsitektur *Mobile SIG*

Gambar diatas adalah cara kerja aplikasi yang saat ini sering digunakan, yaitu dengan menggunakan sistem *Web Services*, dimana semua perangkat (*device*) yang berperan sebagai *client* mendapatkan informasi melalui *Web Services* yang dibagikan oleh *Server*. Teknologi *Web Services* merupakan salah satu cara agar setiap fungsi bisnis dapat berkomunikasi. *Web Services* tidak

menyediakan GUI (*Graphical User Interface*), *Web Services* hanya menyediakan *business logic*, proses dan data dalam antarmuka pemrograman antar jaringan.

### **2.2.8 Google Maps**

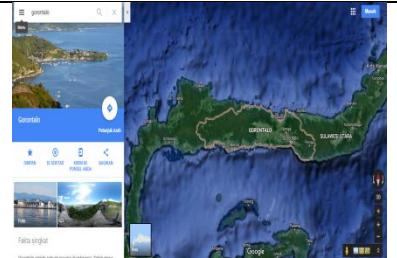
*Google Maps* adalah layanan aplikasi peta *online* yang disediakan oleh *Google* secara gratis. Layanan peta *Google Maps* secara resmi dapat diakses melalui situs <http://maps.google.com>. Pada situs tersebut dapat dilihat informasi geografis pada hampir semua permukaan di Bumi kecuali daerah kutub utara dan selatan. Layanan ini dibuat sangat interaktif, karena di dalamnya peta dapat digeser sesuai keinginan pengguna, mengubah level *zoom*, serta mengubah tampilan jenis peta. *Google Maps* mempunyai banyak fasilitas yang dapat dipergunakan misalnya pencarian lokasi dengan memasukkan kata kunci, kata kunci yang dimaksud seperti nama tempat, kota, atau jalan, fasilitas lainnya yaitu perhitungan rute perjalanan dari satu tempat ke tempat lainnya.

#### **2.2.8.1 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 potongan-potongan gambar yang diminta. Keseluruhan citra yang ada diintegrasikan ke dalam *database* pada *Google Server*, yang nantinya akan dapat dipanggil sesuai kebutuhan permintaan. Tiap-tiap potongan tile, mewakili gambar tertentu dalam *longitude*, *latitude* dan *zoom level* tertentu.

### 2.2.8.2 Macam-macam tampilan Google Maps

**Tabel 2.2** Macam-macam Tampilan Google Maps

Tipe Google Map	Kode	Keterangan	Gambar
Normal	<code>setMapType(GoogleMap.MAP_TYPE_NORMAL);</code>	Menampilkan jalan beserta label namanya	
Hybrid	<code>setMapType(GoogleMap.MAP_TYPE_HYBRID);</code>	Tipe satelit yang ditambahkan tampilan label jalannya	
Terrain	<code>setMapType(GoogleMap.MAP_TYPE_TERRAIN);</code>	Menampilkan data topografi daerah	

**Gambar 2.2** Tampilan Map Normal

**Gambar 2.3** Tampilan Map Hybrid

**Gambar 2.4** Tampilan Map Terrain

Satelit	<code>setMapType(GoogleMap.MAP_TYPE_SATELLITE);</code>	Menampilkan foto saja tanpa label jalan dan lainnya	
---------	--	---	---

**Gambar 2.5** Tampilan Map Satelit

**Sumber:** *Panduan Mudah Pengembangan Google Map Android*, Yusuf Mufti, 2015

### 2.2.9 Google Maps API

Google telah menyediakan layanan bagi para *developer* aplikasi Android untuk menggunakan *API Google Map* pada aplikasi mereka. API sendiri adalah singkatan dari *Application Programming Interface*.

*Google Maps API* adalah layanan gratis yang diberikan oleh *Google* dan sangat popular. *Google Maps API* juga bisa dikatakan sebagai suatu *library* yang berbentuk *JavaScript*. *Google Maps API* memanfaatkan teknologi *digital imaging*, seperti foto satelit untuk dapat melihat *landscape* planet bumi apabila dilihat dari luar angkasa.

Perlu diketahui bahwa perkembangan penggunaan *Google Map* di Android yang ada saat ini dimula dengan adanya *Google Map V1*, yang penggunaannya telah dihentikan pada akhir tahun 2012. Mulai tahun 2013, aplikasi Android yang ingin menampilkan *Google Map* harus menggunakan layanan *Google Map V2*. Mulai dari penggunaan *SHA1* yang menggantikan *MD5* untuk mendapatkan *Google API Key* hingga penggunaan *Fragment* yang menggantikan *MapView*.

Dalam pembuatan program *Google Maps API* menggunakan urutan sebagai berikut:

1. Memasukkan *Maps API JavaScript* ke dalam *HTML* kita.
2. Membuat element div dengan nama *map\_canvas* untuk menampilkan peta.
3. Membuat beberapa objek literal untuk menyimpan properti-properti pada peta.
4. Menuliskan fungsi *JavaScript* untuk membuat objek peta.
5. Menginisiasi peta dalam *tag body HTML* dengan *event onload*.

### 2.3 *Java*

Berbicara mengenai *Java*, kita sebenarnya membicarakan tentang dua hal yang saling berkaitan. Yang pertama adalah *Java* sebagai bahasa pemrograman dan *Java* sebagai *platform* pengembangan aplikasi. Bahasa pemrograman *Java* pada awalnya dibuat oleh James Gosling pada tahun 1995 sebagai bagian dari *Sun Microsystem Java Platform*. Sintaks *Java* banyak diturunkan dari C dan C++ tetapi lebih sederhana, ketat dan mempunyai akses ke OS yang lebih terbatas. Hal ini karena *Java* ditujukan sebagai bahasa pemrograman yang cukup sederhana untuk dipelajari dan mudah dibaca.

Aplikasi *Java* ditulis sebagai file berekstensi *.java* yang dicompile menjadi file *.class*. File *.class* ini adalah *bytecode* yang bisa dijalankan disemua *Java Virtual Machine*, tidak peduli apapun OS-nya ataupun arsitektur *processor* nya. *Java* adalah bahasa yang ditujukan untuk semua kebutuhan, *concurrent*, berbasis *class*, *object oriented* serta didesain agar tidak tergantung terhadap lingkungan dimana aplikasi dijalankan (OS dan *processor*).

*Java* ditujukan agar bisa “ditulis sekali, bisa jalan di manapun”. Sekarang ini *Java* adalah bahasa pemrograman paling populer dan paling banyak digunakan untuk membuat aplikasi baik aplikasi di *embedded system, mobile, desktop* hingga *web application*.

*Java* mempunyai lima prinsip penting yang dijadikan sebagai tujuannya, kelima prinsip ini adalah :

1. *Java* harus “sederhana, *object oriented* dan mudah dimengerti”
2. *Java* harus “kuat dan aman”
3. *Java* harus “netral terhadap arsitektur *system* (OS,processor) dan bisa jalan di manapun”
4. *Java* harus bisa dijalankan dengan “kinerja yang tinggi”
5. *Java* harus “*interpreted, threaded* dan dinamis”

Dengan kelima prinsip di atas, aplikasi *Java* mempunyai popularitas yang sangat tinggi terutama di dunia *enterprise application*. Dimana semua prinsip di atas sangat cocok untuk jenis aplikasi ini. Industri yang mempunyai *budget* tinggi untuk IT seperti perbankan dan telekomunikasi menggunakan *Java* secara ekstensif. Banyak aplikasi dengan skala raksasa dibangun menggunakan *platform Java*.

*Java Platform* terdiri dari tiga buah profile : *Java ME (Java Micro Edition)* adalah *java* yang bisa berjalan di dalam *embedded system* seperti *Java Card* dan *Handphone*. *Java SE (Java Standard Edition)* adalah *java* yang bisa berjalan di dalam PC maupun *server* sebagai aplikasi *standalone* maupun aplikasi *desktop*. *Java EE (Java Enterprise Edition)* adalah profile *java* yang ditujukan untuk

membuat aplikasi *Enterprise* seperti *Web Application (Servlet)* dan *Enterprise Java Bean (EJB)*. Instalasi *platform Java* terdiri dari dua paket aplikasi. Paket yang pertama adalah JRE (*Java Runtime Environment*), Paket kedua adalah JDK (*Java Development Kit*).

### **2.3.1 *Java Runtime Environment (JRE)***

JRE (*Java Runtime Environment*), paket ini terdiri dari semua aplikasi yang dibutuhkan agar sebuah aplikasi *Java* bisa berjalan, seperti *library* dan *JVM (Java Virtual Machine)*.

### **2.3.2 *Java Development Kit (JDK)***

JDK (*Java Development Kit*), paket ini terdiri dari JRE (*Java Runtime Environment*) dan ditambah dengan perkakas untuk membuat aplikasi *Java* seperti *java compiler (javac)*, *java documentation (javadoc)* dan *java archive (jar)*.

### **2.3.3 *Android Studio***

*Android Studio* adalah IDE (*Integrated Development Environment*) resmi Android. Yang dibuat Android untuk mempercepat pengembangan dan membantu dalam membangun aplikasi berkualitas tinggi untuk setiap perangkat Android serta menawarkan *tool* yang disesuaikan untuk pengembang Android, termasuk editing kode, *debugging*, pengujian, dan *profil tool*.

### **2.3.4 *Framework***

*Framework* atau kerangka kerja adalah sebuah *software* untuk memudahkan para *programmer* membuat aplikasi atau *web* yang isinya adalah berbagai fungsi, *plugin*, dan konsep sehingga membentuk suatu sistem tertentu.

Dengan menggunakan *framework*, sebuah aplikasi akan tersusun dan terstruktur dengan rapi. Namun, menggunakan *framework* bukan berarti kita bebas dari pengkodean. Sebagai pengguna/*programmer* menggunakan variabel dan fungsi-fungsi yang ada di *framework* tersebut. Karena itulah, kerja menjadi efektif karena tidak harus membuat fungsi-fungsi lagi.

### 2.3.5 *Spring*

*Spring* adalah *Dependency Injection* (DI) atau *Inversion of Control* (IoC). Tugas utama *Spring* adalah mengatur objek-objek yang dibutuhkan dalam aplikasi. Misalnya *SessionFactory* dan *Session* adalah objek-objek yang diperlukan untuk bekerja dengan *Hibernate*, *Spring* dapat membantu mengatur objek-objek ini sehingga kode infransturktur (*boiler code*) dapat dikurangi secara signifikan karena ditangani oleh *Spring*.

Kode infrasturktur (*boiler plate code*) contohnya adalah membuka dan menutup *Sesssion*, menangani *transaction* seperti *transaction begin*, *commit* dan *rollback*. *Spring* akan mengambil alih tanggung jawab untuk mengatur kode infrastruktur ini dari tangan *developer*, hal ini sangat strategis karena *developer* seringkali lupa untuk menulis kode infrastruktur yang dapat mengakibatkan *memory leak* atau merusak konsistensi data.

Peranan utama *Spring* dalam aplikasi *desktop* cukup besar, salah satu tugas utama *Spring* adalah mengatur *Hibernate* agar *developer* tidak perlu secara manual harus membuat kode untuk membuka menutup *session* (*Transparent Session Management*) dan menangani *transaction* (*declarative transaction*). *Transaction* yang ditangani secara manual dengan kode disebut gaya

pemrograman *Programmatic Transaction*, sedangkan gaya pemrograman yang menyerahkan penanganan *transaction* ke *framework* disebut *Declarative Transaction*.

*Spring* mempunyai *feature Aspect Oriented Programming* (AOP) yang menjadi tulang punggung dibalik implementasi *declarative transaction* dan *transparent Session management*. AOP memungkinkan *Spring* menyisipkan kode untuk memulai *transaction* kemudian membuka *session* pada saat sebuah *method* dalam *service* akan dieksekusi, kemudian disisipkan pula kode untuk me-*rollback* dan menutup *session* ketika *method* dalam *service* muncul *exception*, dan setelah *method* selesai dilaksanakan *Spring* menyisipkan kode untuk meng-*commit transaction* plus menutup *session*. Semua proses penyisipan kode ini dilakukan secara *runtime* dan transparan terhadap *developer*. Sehingga *developer* cukup berkonsentrasi untuk mengimplementasikan *bussiness logic* tanpa harus khawatir lupa meng-*commit transaction* atau menutup *Session*.

### **2.3.6 Integrated Development Environment (IDE)**

IDE adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Dengan menggunakan Java IDE (*Integrated Development Environment*) tertentu, semua kebutuhan pemrograman akan dijadikan dalam satu tempat, mulai dari *text editor*, *compiler/interpreter*, *system help* dan terkadang juga terdapat fitur lain yang sangat bermanfaat dalam penulisan kode (seperti: *code auto-complete* dan *syntax highlight*).

### 2.3.7 Netbeans

*Netbeans* merupakan sebuah aplikasi *Integrated Development Environment* (IDE) yang berbasiskan *Java* dari *Sun Microsystems* yang berjalan di atas *swing*. *Swing* merupakan sebuah teknologi *Java* untuk pengembangan aplikasi *desktop* yang dapat berjalan pada berbagai macam *platform* seperti *windows*, *linux*, *Mac OS X* dan *Solaris*. Sebuah IDE merupakan lingkup pemrograman yang diintegrasikan ke dalam suatu aplikasi perangkat lunak yang menyediakan *Graphic User Interface* (GUI), suatu kode editor atau *text*, suatu *compiler* dan suatu *debugger*.

Netbeans juga digunakan oleh sang *programmer* untuk menulis, meng-*compile*, mencari kesalahan dan menyebarkan program *netbeans* yang ditulis dalam bahasa pemrograman *java*. Selain itu dapat juga mendukung bahasa pemrograman lainnya dan program ini pun bebas untuk digunakan dan untuk membuat professional *desktop*, *enterprise*, *web*, dan *mobile applications* dengan *Java*, *C/C++*, dan bahkan *dynamic languages* seperti *PHP*, *JavaScript*, *Groovy*, dan *Ruby*.

## 2.4 Basis data (*Database*)

Basis data atau yang dalam bahasa pemrograman lebih dikenal dengan istilah *database* didefinisikan sebagai himpunan kelompok data yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

Konsep basis data dianalogikan seperti kita ingin mencari sebuah buku yang ada diantara 12 buku. Untuk dapat menemukannya, kita harus memeriksa

satu persatu buku tersebut. Namun jika buku tersebut ada diantara 500 buku, tentu saja akan lebih repot ketika melakukan hal yang sama dengan hal sebelumnya. Solusinya adalah dengan memisahkan buku-buku tersebut kedalam sebuah rak yang disusun berdasarkan kategori, sehingga akan jauh lebih cepat dan mudah bagi kita untuk menemukan buku yang dicari.

Buku-buku itu diibaratkan sebagai data, baris rak itu diibaratkan sebagai tabel dan rak buku diibaratkan sebagai basis data. Data didalam file basis data disimpan dalam struktur tertentu yang membuatnya lebih cepat diakses dibandingkan jika data tersebut disimpan dalam *file* lain. Tujuan basis data adalah sebagai berikut:

1. Kecepatan dan kemudahan (*Speed*)
2. Efisiensi ruang penyimpanan (*Space*)
3. Keakuratan (*Accuracy*)
4. Ketersediaan (*Availability*)
5. Kelengkapan (*Completeness*)
6. Keamanan (*Security*)
7. Pemakaian bersama (*Sharability*)

Salah satu aplikasi *database* yang sering digunakan adalah *MySQL*. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management system*) atau DBMS yang *multithread, multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License*

(GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

## 2.5 *Web-server*

*Server* atau *Web server* adalah sebuah *software* yang memberikan layanan berbasis data dan berfungsi menerima permintaan dari *HTTP* atau *HTTPS* pada klien yang dikenal dan biasanya kita kenal dengan nama *web browser* (*Mozilla Firefox*, *Google Chrome*) dan untuk mengirimkan kembali yang hasilnya dalam bentuk beberapa halaman *web* dan pada umumnya akan berbentuk dokumen *HTML*.

Fungsi utama *Server* atau *Web server* adalah untuk melakukan atau akan mentransfer berkas permintaan pengguna melalui protokol komunikasi yang telah ditentukan sedemikian rupa. halaman *web* yang diminta terdiri dari berkas teks, video, gambar, file dan banyak lagi. pemanfaatan *web server* berfungsi untuk mentransfer seluruh aspek pemberkasan dalam sebuah halaman *web* termasuk yang di dalam berupa teks, video, gambar dan banyak lagi.

## 2.6 *Client-server*

*Client-server* adalah suatu bentuk arsitektur. Arsitektur *client server* mengurangi beban pemrosesan dari komputer sentral ke komputer *client*. Ini berarti semakin banyak *user* bertambah pada aplikasi *client/server*, kinerja *server* *file* tidak akan menurun dengan cepat. Dengan *client server*, *user* dari berbagai lokasi dapat mengakses data yang sama dengan sedikit beban pada sebuah mesin tunggal. Arsitektur *client-server* mempunyai kelebihan sebagai berikut:

- Lebih aman.
- Semua data dapat dibackup pada satu lokasi sentral.
- Kecepatan akses lebih tinggi karena penyediaan fasilitas jaringan dan pengelolaannya dilakukan secara khusus oleh satu komputer (*server*) yang tidak dibebani dengan tugas lain sebagai *workstation*.

Selain kelebihan, arsitektur client-server juga memiliki kekurangan sebagai berikut:

- Membutuhkan *administrator* yang handal.
- Pelaksanaannya mahal.
- Jika server mati, maka komputer client akan mati juga.

### **2.6.1 Client**

*Client* adalah perangkat yang akan menampilkan dan menjalankan sebuah program aplikasi (*software*). Fungsi *client*, adalah sebagai berikut:

- a. Mengatur *user interface*
- b. Menerima dan memeriksa sintaks input dari pemakai
- c. Memproses aplikasi
- d. *Generate* permintaan basis data dan memindahkannya ke *server*
- e. Memberikan respon balik kepada pemakai
- f. Menyediakan akses basis data secara bersamaan
- g. Menyediakan kontrol *recovery*

### **2.6.2 Server**

*Server* adalah perangkat yang menyediakan dan bertindak sebagai pengelola aplikasi, data, dan keamanannya. Fungsi *server* adalah sebagai berikut:

- a. Menerima dan memproses basis data yang diminta dari *client*
- b. Memeriksa autorisasi
- c. Menjamin tidak terjadi pelanggaran terhadap *integrity constraint*
- d. Melakukan *query/pemrosesan update* dan memindahkan respon ke *client*
- e. Memelihara *data dictionary*

### **2.6.3 Cara Kerja *Client-Server***

Dalam arsitektur *client server*, pemrosesan sebuah aplikasi terjadi pada *client* dan *server*. *Client server* adalah tipikal sebuah aplikasi *two-tier* dengan banyak *client* dan sebuah *server* yang dihubungkan melalui sebuah jaringan.

Aplikasi ditempatkan pada komputer *client* dan mesin *database* dijalankan pada *server* jarak jauh. Aplikasi *client* mengeluarkan permintaan ke *database* yang mengirimkan kembali data ke *client*-nya. Model *Two-tier* terdiri dari tiga komponen yang disusun menjadi dua lapisan: *client* meminta *service* dan *server* yang menyediakan *service*. Dan tiga komponen tersebut adalah:

1. *User Interface*: antamuka program aplikasi yang berhadapan dan digunakan langsung oleh *user*.
2. Manajemen proses.
3. *Database* model ini memisahkan peranan *user interface* dan *database* dengan jelas sehingga terbentuk dua lapisan.

## **2.7 Android**

Android merupakan sistem operasi yang berbasis *Linux* dan dirancang untuk perangkat seluler layar sentuh seperti *smartphone* serta komputer *tablet*.

Android pada awalnya dikembangkan oleh perusahaan bernama *Android, Inc.*, dengan dukungan finansial yang berasal dari *Google*, yang kemudian *Google* pun membelinya pada tahun 2005. Sistem Operasi Android tersebut secara resmi dirilis pada tahun 2007, bersamaan dengan didirikannya sebuah perusahaan *Open Handset Alliance*, konsorsium dari beberapa perusahaan-perusahaan perangkat keras, perangkat lunak, serta telekomunikasi yang memiliki tujuan untuk memajukan standar terbuka dari perangkat seluler. Ponsel yang berbasis sistem operasi Android pertama dijual pada bulan Oktober 2008.

Sejarah Android pada mulanya berasal dari perusahaan bernama *Android, Inc.* didirikan tempatnya di Palo Alto, California, pada Oktober tahun 2003 oleh Andy Rubin (pendiri Danger), Rich Miner seorang pendiri *Wildfire Communications, Inc.*, Nick Sears seorang mantan VP T-Mobile, dan Chris White seorang kepala desain dan pengembangan antarmuka WebTV untuk mengembangkan sebuah "perangkat seluler pintar yang lebih sadar tentang lokasi dan preferensi penggunanya".

Tujuan awal dari perkembangan tersebut pada mulanya diperuntukkan bagi kamera digital, namun disadari bahwa pasar dari kamera digital tidak besar potensinya, dan pengembangan Android lalu dialihkan pada pasar telefon pintar atau *smartphone* untuk menyaingi *Symbian* serta *Windows Mobile* (*iPhone Apple* pada saat itu belum dirilis). Meskipun para pengembang Android tersebut merupakan pakar-pakar teknologi yang berpengalaman, *Android Inc.* dijalankan secara diam-diam dan hanya diungkapkan bahwa para pengembang tersebut sedang berusaha menciptakan sebuah perangkat lunak yang dapat diperuntukkan

untuk telepon seluler. Masih pada tahun yang sama, Andy Rubin kehabisan uang. Steve Perlman adalah seorang teman dekat Andy Rubin dan meminjaminya \$10.000 tunai serta menolak tawaran saham di perusahaan.

*Google* mengakuisisi perusahaan *Android Inc.* pada tanggal 17 Agustus 2005 dan menjadikannya sebagai anak perusahaan yang dimiliki oleh *Google*. Pendiri *Android Inc.* yaitu Rubin, Miner, serta White tetap bekerja pada perusahaan tersebut setelah diakuisisi oleh *Google*. Di *Google*, tim yang dipimpin oleh Andy Rubin mulai untuk mengembangkan sebuah *platform* perangkat seluler dengan menggunakan *kernel Linux*.

Sejak tahun 2008, Android mulai secara bertahap melakukan sejumlah pembaruan atau *update* untuk meningkatkan kinerja dari sistem operasi tersebut dengan menambahkan fitur baru, memperbaiki *bug* pada versi Android yang sebelumnya. Setiap versi yang dirilis dinamakan secara alfabetis dengan berdasarkan nama sebuah makanan manis. Sejauh ini Android seperti cupcake, donut, dan sebagainya.



**Gambar 2.6** Logo Android

### 2.7.1 Versi-versi Android

Sebelum diberi nama-nama makanan manis sebagai nama untuk versi OS, Android sempat memiliki 2 versi awal dengan nama Android *Alpha* dan *Beta*. Nama untuk versi pertama sebenarnya adalah Android *Astro*, namun karena alasan hak cipta (*trademark*), nama ini tidak jadi digunakan. Di versi awal ini belum ada perangkat dengan sistem operasi Android yang dijual secara komersil. Versi yang dirilis berikutnya mulai dinamakan secara alfabetis berdasarkan nama sebuah makanan manis seperti yang terlampir dalam tabel berikut ini:

**Tabel 2.3** Versi – versi Android

Nama	Versi	Tanggal Rilis
Cupcake	1.5	30 April 2009
Donut	1.6	15 September 2009
Éclair	2.0 – 2.1	26 Oktober 2009
Froyo	2.2	20 Mei 2010
Gingerbread	2.3	6 Desember 2010
Honeycomb	3.0 – 3.2	10 Mei 2011
Ice Cream Sandwich	4.0	16 Desember 2011
Jelly Bean	4.1 – 4.3	9 Juli 2012
KitKat	4.4	31 Oktober 2013
Lollipop	5.0 – 5.1	15 Oktober 2014
Marshmallow	6.0	September 2015
Nougat	7.0	Juni 2016

Android versi *Nougat* resmi diperkenalkan pada akhir Juni 2016. Banyak netizen yang berspekulasi bahwa kemungkinan besar, pemberian nama untuk Android versi “N” ini adalah *Nutella*. Namun *Google* menepis kabar tersebut setelah resmi memperkenalkannya bersamaan dengan dipamerkannya patung *icon* Android yang berdiri diatas potongan *Nougat* (yang sepintas lebih mirip dengan tempe itu).

Sebelumnya, *Google* telah mengundang para penggunanya untuk memberikan ide penamaan pada versi ini. Beberapa nama termasuk *Nutella* dan *Nastar* pun muncul, hingga akhirnya *Google* lebih memilih nama *Nougat*.



**Gambar 2.7 Logo Android Versi 7.0 (Nougat)**

## 2.8 Siklus Hidup Pengembangan Sistem

Dalam membangun sebuah sistem digunakan metode siklus hidup pengembangan sistem (*System Development Life Cycle* atau SDLC). SDLC terdiri dari sejumlah tahapan yang dilaksanakan secara berurutan yaitu :

1. *Analisis*
2. *Design*
3. *Implementasi*

*System Development Life Cycle* atau siklus hidup pengembangan sistem (SDLC), merupakan metode alternatif. Metode SDLC mempunyai beberapa kelebihan dan kekurangan.

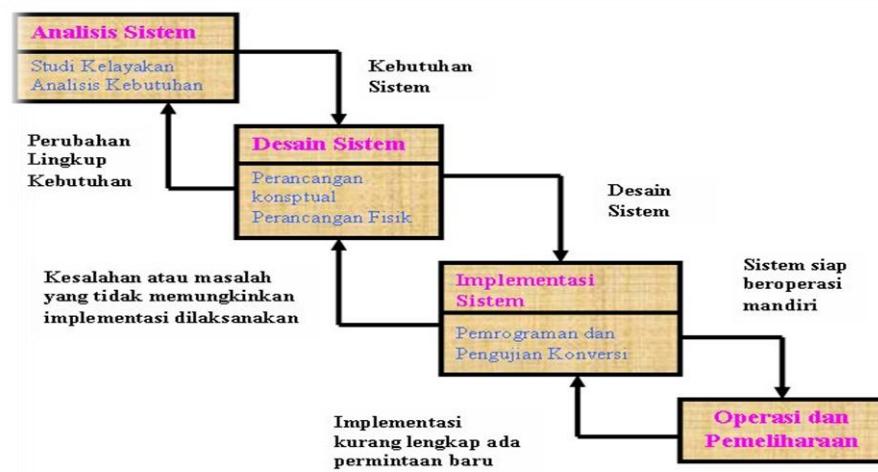
Kelebihan – kelebihan dari metode ini adalah :

1. Menyediakan tahapan yang dapat digunakan sebagai pedoman mengembangkan sistem.
2. Memberikan hasil sistem yang lebih baik karena sistem dianalisis dan dirancang secara keseluruhan sebelum diimplementasikan.

Disamping kelebihan-kelebihan tersebut, SDLC juga mempunyai kekurangan, diantaranya adalah :

1. Hasil dari SLDC tergantung dari hasil tahap analisis, sehingga jika terdapat kesalahan analisis, akan terbawa terus.
2. Dibutuhkan waktu yang lama untuk mengembangkannya karena sistem harus dikembangkan sampai selesai semua terlebih dahulu.

Siklus hidup pengembangan sistem dengan langkah-langkah utama adalah sebagai berikut:



**Gambar 2.8** Siklus Hidup Pengembangan Sistem

### **2.8.1. Analisis Sistem**

Menurut Jogiyanto (2005:129) analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

Tahap analisis merupakan tahap yang kritis dan sangat penting, karena kesalahan didalam tahap ini akan menyebabkan juga kesalahan ditahap selanjutnya.

Menurut Abdul Kadir (2003 : 38) analisis sistem mencakup studi kelayakan dan analisis kebutuhan yaitu :

a. Studi kelayakan

Studi kelayakan merupakan proses yang mempelajari atau menganalisa permasalahan yang telah ditentukan sesuai dengan tujuan akhir yang akan dicapai. Analisis kelayakan digunakan untuk menentukan kemungkinan keberhasilan solusi yang diusulkan. Tahapan ini berguna untuk memastikan bahwa solusi yang diusulkan tersebut benar-benar dapat tercapai dengan sumber daya dan dengan memperhatikan kendala yang terdapat pada permasalahan serta dampak terhadap lingkungan sekeliling. Lima macam kelayakan dalam merancang sistem informasi yaitu kelayakan teknik, kelayakan ekonomi, kelayakan operasi, kelayakan hukum dan kelayakan jadwal.

b. Analisis kebutuhan

Analisis kebutuhan merupakan proses untuk menghasilkan spesifikasi kebutuhan. Spesifikasi kebutuhan adalah spesifikasi yang rinci tentang pengolahan data yaitu jumlah data yang harus diproses, waktu pengolahan saat data siap diproses sampai informasi yang dihasilkan. Spesifikasi ini digunakan untuk membuat kesepakatan dalam pengembangan sistem.

Di dalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analis sistem sebagai berikut:

- a. *Identify*, yaitu mengidentifikasi masalah.
- b. *Understand*, yaitu memahami kerja dari sistem yang ada.
- c. *Analyze*, yaitu menganalisis sistem.
- d. *Report*, yaitu membuat laporan hasil analisis..

### **2.8.2. Desain Sistem**

Setelah tahap analisis sistem selesai dilakukan, maka analis sistem telah mendapatkan gambaran dengan jelas apa yang harus dikerjakan. Tiba waktunya sekarang bagi analis sistem untuk memikirkan bagaimana membentuk sistem tersebut. Tahap ini disebut dengan desain sistem (*systems design*). (Jogiyanto, 2005 : 195)

Tahap desain sistem mempunyai dua maksud atau tujuan utama, yaitu sebagai berikut :

1. Untuk memenuhi kebutuhan kepada pemakai sistem.

2. Untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada pemrogram komputer dan ahli-ahli teknik lainnya yang terlibat.

Desain sistem dapat dibagi dalam dua bagian, yaitu desain sistem secara umum (*general systems design*) dan desain sistem terinci (*detailed systems design*).

**a. Desain Sistem Secara Umum (*General Systems Design*)**

Tujuan dari desain sistem secara umum adalah untuk memberikan gambaran secara umum kepada user tentang sistem yang baru, yang mana merupakan persiapan dari desain sistem secara rinci. Desain secara umum dilakukan oleh analisis sistem untuk mengidentifikasi komponen-komponen sistem informasi yang akan di desain secara rinci oleh pemrograman komputer dan ahli teknik lainnya.

**b. Desain Sistem Terinci (*detailed systems design*)**

Desain sistem terinci terbagi atas:

**1. Desain *Output* Terinci**

Desain output terinci dimaksudkan untuk mengetahui bagaimana dan seperti apa bentuk *output-output* dari sistem yang baru. Desain *output* terinci terbagi atas dua, yaitu desain *output* berbentuk laporan di media kertas dan desain *output* dalam bentuk dialog di layar terminal.

a. Desain *Output* dalam bentuk laporan

Desain ini dimaksudkan untuk menghasilkan output dalam bentuk laporan di media kertas. Bentuk laporan yang paling banyak digunakan adalah dalam bentuk tabel dan berbentuk grafik atau bagan.

b. Desain *Output* dalam bentuk layar terminal

Desain ini merupakan rancang bangun dari percakapan antara pemakai sistem (*user*) dengan komputer. Percakapan ini terdiri dari proses memasukkan data ke sistem, menampilkan *output* informasi kepada *user*, atau keduanya.

## 2. Desain *Input* Terinci

Masukan merupakan awal dimulainya proses informasi. Bahan mentah dari informasi adalah data yang terjadi dari transaksi-transaksi yang dilakukan oleh organisasi. Data hasil dari transaksi merupakan masukan untuk sistem informasi. Hasil dari sistem informasi tidak lepas dari data yang dimasukan.

## 3. Desain *Database* Terinci

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan dan disimpan diluar komputer dan digunakan perangkat lunak tertentu untuk memanipulasinya. *Database* merupakan salah satu komponen yang penting di sistem informasi, karena berfungsi sebagai penyedia informasi bagi para pemakainya. Penerapan *database* dalam sistem informasi disebut *Database System*.

Sistem basis data (*database system*) adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan dengan yang

lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam di dalam suatu organisasi. Dengan sistem basis data ini tiap-tiap orang atau bagian dapat memandang *database* dari beberapa sudut pandang yang berbeda. Bagian kredit dapat memandangnya sebagai data penjualan, bagian personalia dapat memandangnya sebagai data karyawan, bagian gudang data dapat memandangnya sebagai data persediaan, semuanya terintegrasi dalam sebuah data yang umum.

#### **4. Desain Teknologi**

Dalam menjalankan model, menyimpan dan mengakses data, menghasilkan dan tahap ini kita menentukan teknologi yang akan digunakan dalam menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi yang dimaksud meliputi :

- a. Perangkat keras (*hardware*), yang terdiri dari alat masukan, alat pemroses, alat *output* dan simpanan luar.
- b. Perangkat lunak (*software*), yang terdiri dari perangkat lunak sistem operasi (*operating system*), perangkat lunak bahasa (*language software*) dan perangkat lunak aplikasi (*application software*).
- c. Sumber daya manusia (*brainware*), misalnya operator komputer, pemrograman, spesialis telekomunikasi, sistem analis dan lain sebagainya.

Desain teknologi sangat diperlukan pada tahap implementasi dan pengujian untuk membuktikan bahwa sistem dapat berjalan secara semestinya.

## 5. Desain Model

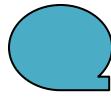
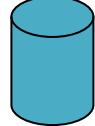
Analisis sistem dapat mendesain model dari sistem informasi yang diusulkan dalam bentuk *physical system* dan *logical model*. Bagan alir sistem (*systems flowchart*) merupakan alat yang tepat digunakan untuk menggambarkan *physical system*. Simbol-simbol bagan alir sistem ini menunjukkan secara tepat arti fisiknya, seperti simbol terminal, *hard disk*, laporan-laporan. *Logical Model* dari sistem informasi lebih menjelaskan kepada *user* bagaimana nantinya fungsi-fungsi di sistem informasi secara logika akan bekerja.

*Logical Model* dapat digambarkan dengan menggunakan diagram arus data (*data flow diagram*). Arus dari data di DAD dapat dijelaskan dengan menggunakan kamus data (*data dictionary*). Sketsa dari *physical system* dapat menunjukkan kepada user bagaimana nantinya sistem secara fisik akan diterapkan.

Bagan alir sistem merupakan bagan yang menunjukkan arus pekerjaan seara keseluruhan dari sistem. Bagan alir sistem digambar dengan simbol-simbol berikut :

**Tabel 2.4** Simbol Bagan Alir Sistem

NAMA SIMBOL	SIMBOL	KETERANGAN
Simbol Terminal		Menunjukkan permulaan atau akhir dari suatu program
Simbol Dokumen		Menunjukkan dokumen <i>input</i> dan <i>output</i> baik itu proses manual, mekanik, atau komputer

Simbol Kegiatan Manual		Menunjukan pekerjaan manual
Symbol Simpanan Offline		Menunjukan file non-komputer yang diarsip urut angka ( <i>numerical</i> ), huruf ( <i>alphabetical</i> ), atau tanggal ( <i>chronological</i> ).
Simbol Kartu Plong		Menunjukan input dan output yang menggunakan kartu plong ( <i>punched card</i> ).
Simbol Proses		Menunjukan kegiatan proses dari operasi program komputer
Simbol Operasi Luar		Menunjukkan operasi yang dilakukan diluar proses operasi komputer
Simbol Pengurutan Offline		Menunjukkan proses urut data di luar proses komputer
Simbol Pita Maknetik		Menunjukkan <i>input</i> dan <i>output</i> menggunakan pita <i>magnetic</i>
Simbol Hard disk		Menunjukkan input dan output menggunakan pita harddisk
Simbol Diskette		Menunjukkan input dan output menggunakan pita diskette
Simbol Drum Maknetik		Menunjukkan input dan output menggunakan drum magnetik
Simbol Pita Kertas Berlubang		Menunjukkan input dan output menggunakan pita kertas berlubang

Simbol Keyboard		Menunjukkan input yang menggunakan on-line keyboard
Simbol Display		Menunjukkan output yang ditampilkan di monitor
Simbol Pita Kontrol		Menunjukkan penggunaan pita kontrol (control tape) dalam batch control total untuk pencocokan di proses batch processing
Simbol Hubungan Komunikasi		Menunjukkan proses transmisi data melalui channel komunikasi
Simbol Garis Alir		Menunjukkan arus dari proses
Simbol Penjelasan		Menunjukkan penjelasan dari suatu proses
Simbol Penghubung		Menunjukkan penghubung ke halaman yang masih sama atau ke halaman yang lain

**Sumber :** Jogiyanto, 2005 : 802

Untuk mempermudah gambaran suatu sistem yang ada atau sistem yang baru yang akan dikembangkan secara logika tanpa memperhatikan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik di mana data tersebut akan di simpan, maka di gunakan Diagram Arus Data atau *Data Flow Diagram* (DFD). Dalam menggambarkan sistem perlu dilakukan pembentukan simbol, berikut ini simbol-simbol yang sering di gunakan dalam DAD :

1. *External entity* (Kesatuan luar) atau *boundary* (batas sistem)

Setiap sistem pasti mempunyai batas sistem (*boundary*) yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima *input*-an dan menghasilkan *output* kepada lingkungan luarnya. Kesatuan luar (*external entity*) merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lain yang berada di lingkungan luarnya yang akan memberikan *input* serta menerima *output* dari sistem. (Jogiyanto, HM, 2005 : 701)



**Gambar 2.9** Notasi kesatuan luar di DAD

2. *Data Flow* (arus data)

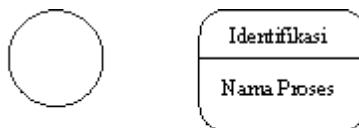
Arus data ini menunjukkan arus atau aliran data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem. (Jogiyanto, HM. 2005 : 701)



**Gambar 2.10** Notasi Arus Data di DAD

3. *Process* (proses)

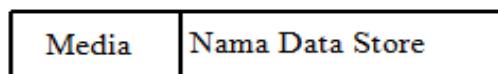
Suatu proses adalah kegiatan atau kerja yang dilakukan orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. (Jogiyanto, HM. 2005 : 705)



**Gambar 2.11** Notasi Proses di DAD

4. *Data Store* (simpanan data)

Simpanan data pada DFD dapat disimbolkan dengan sepasang garis horizontal parallel yang tertutup disalah satu ujungnya. (Jogiyanto, HM. 2005 : 707)



**Gambar 2.12** Notasi Simpanan Data di DAD

### 2.8.3. Implementasi Sistem

Menurut Abdul Kadir dalam bukunya “Perancangan Sistem Informasi” Andi, Yogyakarta, 2003 mengungkapkan : “Implementasi adalah kegiatan yang dilakukan untuk menguji data dan menerapkan sistem yang diperoleh dari kegiatan seleksi”.

Implementasi merupakan salah satu unsur pertahapan dari keseluruhan pembangunan sistem komputerisasi, dan unsur yang harus dipertimbangkan dalam pembangunan sistem komputerisasi yaitu masalah perangkat lunak (*software*), karena perangkat lunak yang digunakan haruslah sesuai dengan masalah yang akan diselesaikan, disamping masalah perangkat keras (*hardware*) itu sendiri.

Menurut Whitten, *et al*, 2004 Implementasi Sistem mempunyai 4 tahap:

1. Membuat dan menguji basis data & jaringan

Penerapan sistem yang baru atau perbaikan sistem dibuat pada basis data dan jaringan yang telah ada. Jika penerapan sistem yang baru memerlukan basis data dan jaringan yang baru atau dimodifikasi, maka sistem yang baru ini biasanya harus diimplementasikan sebelum pemasangan program komputer.

## 2. Membuat dan menguji program

Merupakan tahap pertama untuk siklus pengembangan sistem yang spesifik bagi *programmer*. Bertujuan untuk mengembangkan rencana yang lebih rinci dalam pengembangan dan pengujian program komputer yang baru.

## 3. Memasang dan menguji sistem baru.

Tahap ini dilakukan untuk menyakinkan bahwa kebutuhan integrasi sistem baru terpenuhi.

## 4. Mengirim sistem baru kedalam sistem operasi.

Tujuan tahap ini adalah untuk mengubah secara perlahan – lahan sistem lama menjadi sistem baru sehingga perlu dilakukan pemasangan basis data yang akan digunakan pada sistem baru.

### **2.8.4. Pemeliharaan Sistem**

Pemeliharaan Sistem adalah suatu kombinasi dari berbagai tindakan yang dilakukan untuk menjaga suatu sistem dalam, atau memperbaikinya sampai suatu kondisi yang bisa diterima. Pada bulan April 1970 didefinisikan sebuah istilah untuk Teknologi Pemeliharaan yang mencakup pengertian yang lebih luas dari pada pengertian Pemeliharaan diatas. Istilah ini adalah *Teroteknologi*.

Sistem perlu dipelihara karena beberapa hal, yaitu :

1. Sistem memiliki kesalahan yang dulunya belum terdeteksi, sehingga kesalahan-kesalahan sistem perlu diperbaiki.
2. Sistem mengalami perubahan-perubahan karena permintaan baru dari pemakai sistem.

3. Sistem mengalami perubahan karena perubahan lingkungan luar (perubahan bisnis).
4. Sistem terinfeksi.
5. Sistem berkas *corrupt*.
6. Perangkat keras melemah.
7. Sistem perlu ditingkatkan.

Tujuan dari pemeliharaan sistem:

1. Untuk memperpanjang usia kegunaan aset dari sistem tersebut. Hal ini terutama penting di Negara berkembang karena kurangnya sumber daya modal untuk penggantian. Di Negara-negara maju kadang-kadang lebih menguntungkan untuk ‘mengganti’ daripada ‘memelihara’.
2. Untuk menjamin ketersediaan optimum peralatan.
3. Untuk menjamin kesiapan operasional dari seluruh peralatan yang diperlukan dalam keadaan darurat setiap waktu.
4. Untuk menjamin keselamatan orang yang menggunakan sarana tersebut.

Jenis Pemeliharaan Sistem :

- a. Pemeliharaan Korektif

Pemeliharaan Korektif adalah bagian pemeliharaan sistem yang tidak begitu tinggi nilainya dan lebih membebani karena pemeliharaan ini mengoreksi kesalahan-kesahan yang ditemukan pada saat sistem berjalan.

- b. Pemeliharaan Adaptif

Pemeliharaan Adaptif dilakukan untuk menyesuaikan perubahan dalam lingkungan data atau pemrosesan dan memenuhi persyaratan pemakai baru.

Lingkungan tempat sistem beroperasi adalah dinamik, dengan demikian, sistem harus terus merespon perubahan persyaratan pemakai. Misalnya, Undang-Undang Perpajakan yang baru mungkin memerlukan suatu perubahan dalam kalkulasi pembayaran bersih. Umumnya pemeliharaan adatif ini baik dan tidak dapat dihindari.

c. Pemeliharaan Perfektif

Pemeliharaan Perfektif mempertinggi cara kerja atau maintainabilitas (kemampuan untuk dipelihara). Tindakan ini juga memungkinkan sistem untuk memenuhi persyaratan pemakai yang sebelumnya tidak dikenal. Ketika membuat perubahan substansial modul apapun, petugas pemeliharaan juga menggunakan kesempatan untuk *meng-upgrade* kode, mengganti cabang-cabang yang kadaluwarsa, memperbaiki kecerobohan, dan mengembangkan dokumentasi. Sebagai contoh, kegiatan pemeliharaan ini dapat berbentuk perekayasaan ulang atau restrukturisasi perangkat lunak, penulisan ulang dokumentasi, pengubahan format dan isi laporan, penentuan logika pemrosesan yang lebih efisien, dan pengembangan efisiensi pengoperasian perangkat.

d. Pemeliharaan Preventif

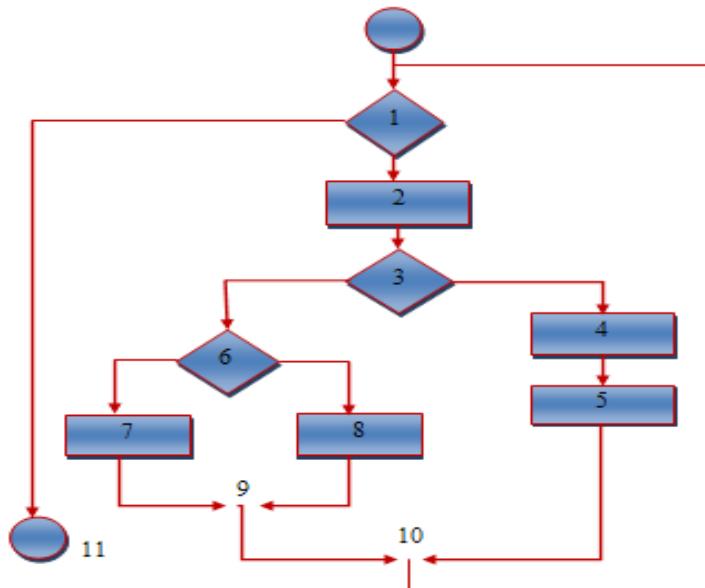
Pemeliharaan Preventif terdiri atas inspeksi periodik dan pemeriksaan sistem untuk mengungkap dan mengantisipasi permasalahan. Karena personil pemeliharaan sistem bekerja dalam sistem ini, mereka seringkali menemukan cacat-cacat (bukan kesalahan yang sebenarnya) yang menandakan permasalahan potensial. Sementara tidak memerlukan tindakan segera, cacat ini bila tidak

dikoreksi ditingkat awal, jelas sekali akan mempengaruhi baik fungsi sistem maupun kemampuan untuk memeliharanya dalam waktu dekat.

## **2.9 Teknik Pengujian Sistem**

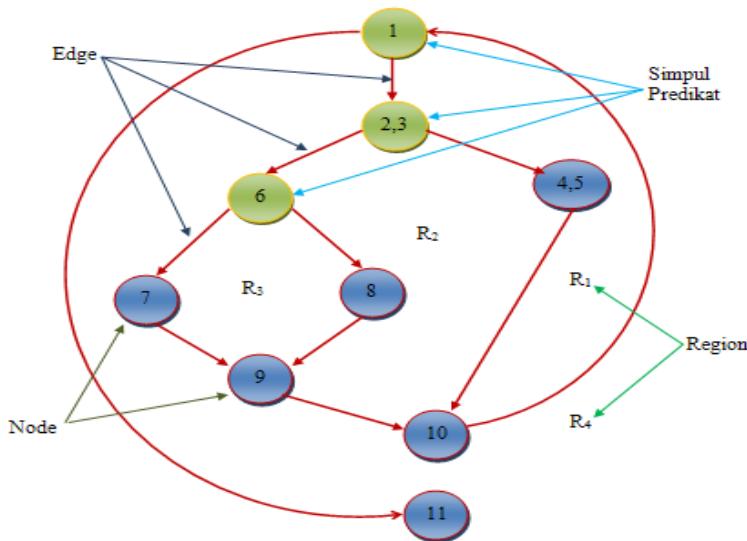
### **2.9.1. *White Box Testing***

Pengujian *White Box* adalah metode pengujian yang menggunakan struktur kontrol desain prosedur untuk memperoleh *test case*. Dengan menggunakan metode *white box*, perekayasa sistem dapat melakukan *test case* yang memberikan jaminan bahwa semua jalur independen pada suatu modul telah digunakan paling tidak satu kali, menggunakan semua keputusan logis pada sisi *true and false*, mengeksekusi semua *loop* pada batasan mereka dan pada batas operasional mereka, dan menggunakan struktur data internal untuk menjamin validitasnya. Pengujian *Basis Path* adalah teknik pengujian *White Box* yang diusulkan pertama kali oleh Tom McCabe. Metode *Basis Path* ini memungkinkan desainer *test case* mengukur kompleksitas logis dari desain prosedural dan menggunakan sebagai pedoman untuk menentukan basis set dari jalur eksekusi. (Roger S. Pressman, 2002 : 536).



**Gambar 2.13** Bagan Alir

Bagan alir digunakan untuk menggambarkan struktur kontrol program dan untuk menggambarkan grafik alir, harus memperhatikan representasi desain prosedural pada bagan alir. Pada gambar dibawah ini, grafik alir memetakan bagan alir tersebut kedalam grafik alir yang sesuai (dengan mengasumsikan bahwa tidak ada kondisi senyawa yang diisikan didalam diamond keputusan dari bagan alir tersebut). Masing-masing lingkaran, yang disebut *simpul* grafik alir, mempresentasikan satu atau lebih statemen prosedural. Urutan kotak proses dan permata keputusan dapat memetakan simpul tunggal. Anak panah disebut *edges* atau *links*, merepresentasikan aliran kontrol dan analog dengan anak panah bagan alir. *Edge* harus berhenti pada suatu simpul, meskipun bila simpul tersebut tidak merepresentasikan statemen prosedural. (Roger S. Pressman, 2002 : 536).



**Gambar 2.14** Grafik Alir

*Kompleksitas siklomatis* adalah metriks perangkat lunak yang memberikan pengukuran kuantitatif terhadap kompleksitas logis suatu program. Bila metriks ini digunakan dalam konteks metode pengujian *basis path*, maka nilai yang terhitung untuk *kompleksitas siklomatis* menentukan jumlah jalur independen. Jalur independen adalah jalur yang melalui program yang mengintroduksi sedikitnya satu rangkaian statemen proses baru atau suatu kondisi baru. Bila dinyatakan dengan terminologi grafik alir, jalur independen harus bergerak sepanjang paling tidak satu *edge* yang tidak dilewatkhan sebelum jalur tersebut ditentukan. Sebagai contoh, serangkaian jalur independen untuk grafik alir yang di tunjukkan pada gambar 2.20 adalah :

Jalur 1 : 1 – 11

Jalur 2 : 1 – 2 – 3 – 10 – 1 – 11

Jalur 3 : 1 – 2 – 3 – 6 – 8 – 9 – 10 – 1 – 11

Jalur 4 : 1 – 2 – 3 – 6 – 7 – 9 – 10 – 1 – 11

Jalur 1, 2, 3, dan 4 yang ditentukan diatas terdiri dari sebuah *basis set* untuk grafik alir pada gambar 2.20 Bagaimana kita tahu banyaknya jalur yang dicari? Komputasi *kompleksitas siklomatis* memberikan jawaban. Fondasi *Kompleksitas siklomatis* adalah teori grafik, dan member kita metriks perangkat lunak yang sangat berguna. Kompleksitas dihitung dalam salah satu dari tiga cara berikut :

1. Jumlah region grafik alir sesuai dengan *kompleksitas siklomatis*.
2. *Kompleksitas siklomatis*,  $V(G)$ , untuk grafik alir  $G$  ditentukan sebagai  $V(G) = E - N + 2$  di mana  $E$  adalah jumlah *edge* grafik alir dan  $N$  adalah jumlah simpul grafik alir.
3. *Kompleksitas siklomatis*,  $V(G)$ , untuk grafik alir  $G$  juga ditentukan sebagai  $V(G) = P + 1$ , di mana  $P$  adalah jumlah predikat yang diisikan dalam grafik alir  $G$ .

Pada gambar 2.20 grafik alir, *kompleksitas siklomatis* dapat dihitung dengan menggunakan masing-masing dari algoritma yang ditulis di atas :

1. Grafik alir mempunyai 4 region.
2.  $V(G) = 11 \text{ edge} - 9 \text{ simpul} + 2 = 4$ .
3.  $V(G) = 3 \text{ simpul yang diperkirakan} + 1 = 4$ .

Dengan demikian, kompleksitas siklomatis dari grafik alir pada gambar 2.20 adalah 4. Yang lebih penting, nilai untuk  $V(G)$  member kita batas atas untuk jumlah jalur independen yang membentuk *basis set*, dan implikasinya batas atas.

### 2.9.2. *Black Box Testing*

Menurut Ladjamudin (2006:379) Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Konsep pengujian *black box* (kotak hitam) digunakan untuk mempresentasikan sistem yang cara kerja didalamnya tidak tersedia untuk diinspeksi karena item-item yang diuji dianggap gelap karena *logic* nya tidak diketahui, yang diketahui hanya apa yang masuk dan apa yang keluar dari kotak hitam.

Uji coba *black box* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses *database eksternal*
4. Kesalahan performa
5. Kesalahan inisialisasi dan terminasi

Tidak seperti metode *white box* yang dilaksanakan diawal proses, ujicoba *black box* diaplikasikan dibeberapa tahapan berikutnya. Karena ujicoba *black box* dengan sengaja mengabaikan struktur kontrol, sehingga perhatiannya difokuskan pada informasi *domain*. Ujicoba didesain untuk dapat menjawab pertanyaan-pertanyaan berikut :

1. Bagaimana validitas fungsionalnya diuji ?
2. Jenis *input* seperti apa yang akan menghasilkan kasus uji yang baik ?
3. Apakah sistem secara khusus sensitif terhadap nilai input tertentu ?
4. Bagaimana batasan-batasan kelas data diisolasi ?

5. Berapa rasio data dan jumlah data yang dapat ditoleransi oleh sistem ?
6. Apa akibat yang akan timbul dari kombinasi spesifik data pada operasi sistem?

Teknik pengujian yang termasuk dalam kategori pengujian *black box* diantaranya *Graph-bases-testing*, *equivalence partitioning*, *comparison testing*, *artogonal array testing*.

## 2.10 Perangkat Lunak Pendukung

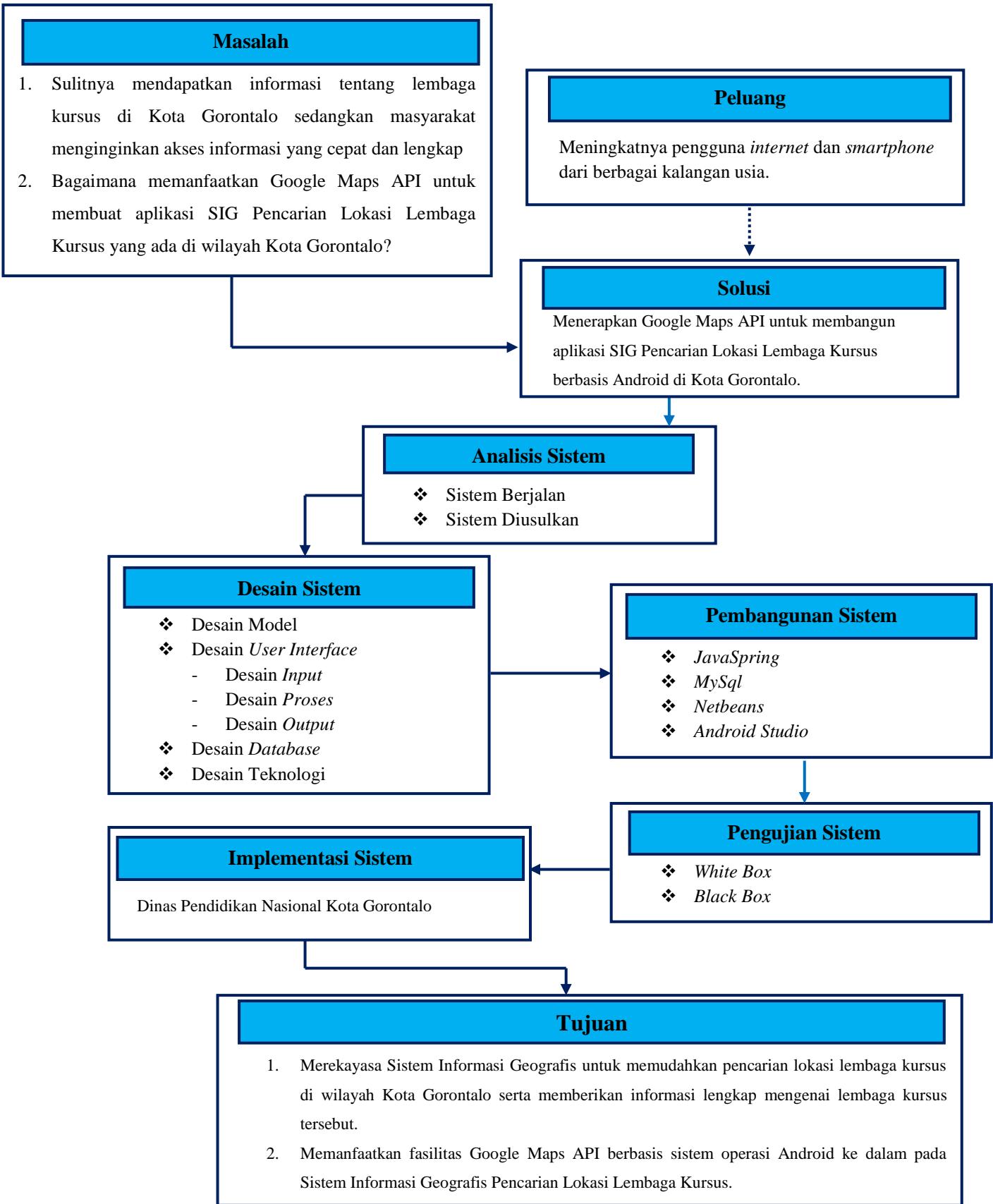
Penelitian ini menggunakan beberapa perangkat lunak pendukung, sebagai berikut :

**Tabel 2.5** Perangkat Lunak Pendukung

No.	Tools	Fungsi
1	<i>JavaSpring</i>	Tugas utama <i>Spring</i> adalah mengatur objek-objek yang dibutuhkan dalam aplikasi. Misalnya <i>SessionFactory</i> dan <i>Session</i> adalah objek-objek yang diperlukan untuk bekerja dengan <i>Hibernate</i> , <i>Spring</i> dapat membantu mengatur objek-objek ini sehingga kode infranstruktur ( <i>boiler code</i> ) dapat dikurangi secara signifikan karena ditangani oleh <i>Spring</i> .
2	<i>MySQL</i>	MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL ( <i>database management system</i> ) atau DBMS yang <i>multithread</i> , <i>multi-user</i> , dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU <i>General Public License</i> (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

3	<i>Netbeans</i>	<p><i>Netbeans</i> merupakan sebuah aplikasi <i>Integrated Development Environment</i> (IDE) yang berbasiskan <i>Java</i> dari <i>Sun Microsystems</i> yang berjalan di atas <i>swing</i>. Sebuah IDE merupakan lingkup pemrograman yang diintegrasikan ke dalam suatu aplikasi perangkat lunak yang menyediakan Graphic User Interface (GUI), suatu kode editor atau text, suatu compiler dan suatu <i>debugger</i>.</p>
4	<i>Android Studio</i>	<p><i>Android Studio</i> adalah IDE (<i>Integrated Development Environment</i>) resmi <i>Android</i>. Yang dibuat <i>Android</i> untuk mempercepat pengembangan dan membantu dalam membangun aplikasi berkualitas tinggi untuk setiap perangkat <i>Android</i> serta menawarkan <i>tool</i> yang disesuaikan untuk pengembang <i>Android</i>, termasuk editing kode, <i>debugging</i>, pengujian, dan <i>profil tool</i>.</p>

## 2.11 Kerangka Pemikiran



## **BAB III**

### **OBJEK DAN METODE PENELITIAN**

#### **3.1. Objek Penelitian**

Objek dari penelitian ini adalah “**Pencarian Lokasi Lembaga Kursus**” penelitian ini dilakukan di Dinas Pendidikan Nasional Kota Gorontalo.

#### **3.2. Metode Penelitian**

Metode yang digunakan dalam penelitian ini adalah penelitian deskriptif, yaitu penelitian yang berusaha untuk memecahkan masalah yang ada sekarang secara sistematis berdasarkan data-data yang ada. Tujuan dari penelitian deskriptif adalah untuk membuat deskripsi, menggambarkan atau melukiskan secara sistematis, faktual dan akurat mengenai fakta-fakta, sifat-sifat serta hubungan antar fenomena yang diselidiki. Tahapan penelitian dapat diuraikan sebagai berikut :

##### **3.2.1. Tahap Pengumpulan Data**

Untuk mengumpulkan data digunakan 2 (dua) jenis data yaitu data primer dan data sekunder. Data primer yaitu data yang berasal dari lapangan dan data sekunder berasal dari penelitian kepustakaan.

###### **1. Penelitian Data Primer (Lapangan)**

Data primer merupakan data langsung dari objek penelitian yang diperoleh dari tempat atau lokasi penelitian, baik berupa catatan, foto, artikel dan buku. (Jogiyanto, 2005).

Dalam penelitian ini, Penulis mengumpulkan data lembaga-lembaga kursus yang ada di wilayah Kota Gorontalo dari pihak Dinas Pendidikan Nasional Kota Gorontalo yang menaungi lembaga-lembaga kursus di wilayah Kota Gorontalo.

## **2. Penelitian Data Sekunder**

Metode Kepustakaan diperlukan untuk mendapatkan data sekunder dengan tujuan melengkapi data primer. Data sekunder didapatkan dari pengkajian kepustakaan yang berisi dasar-dasar teori. Metode kepustakaan digunakan oleh analis sistem dengan cara mengambil contoh dokumen-dokumen yang berhubungan dengan materi penelitian.

Penulis mengumpulkan buku maupun literatur yang berhubungan dengan Mobile GIS berbasis Android serta mengakses informasi dari *internet* untuk menunjang hasil penelitian data primer.

### **3.2.2. Tahap Analisis Sistem**

Pada tahap ini, selain merupakan tahap perencanaan yang merupakan tahapan awal dalam pengembangan sistem dengan maksud melakukan studi-studi terhadap kebutuhan-kebutuhan sistem atau pengguna, tahap ini juga menguraikan sistem yang sedang berjalan dan sistem yang diusulkan di Dinas Pendidikan Nasional Kota Gorontalo dengan maksud untuk megidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi, dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Adapun analisis sistem yang berjalan dan diusulkan dapat digambarkan menggunakan bagan alir (*flowchart*) sistem/dokumen.

### **3.2.3. Tahap Desain Sistem**

Tahap ini merancang sistem yang diusulkan berdasarkan hasil analisis yang telah dilakukan sebelumnya. Tahap ini merupakan strategi untuk memecahkan masalah dan mengembangkan solusi terbaik bagi permasalahan sistem. Jika pada tahap analisis menekankan pada masalah bisnis, maka sebaliknya tahap desain fokus pada sisi teknis dan implementasi perangkat lunak dari sistem yang diusulkan. Tahap desain merupakan tugas dan aktivitas yang difokuskan pada spesifikasi detail dari solusi berbasis komputer. Alat (*tools*) yang digunakan dalam desain sistem ini, dalam hal ini untuk desain model adalah DAD (Diagram Arus Data) dan Kamus Data. Untuk desain *input* dan *output* menggunakan *Edraw Max5*. Sedangkan untuk desain basis data menggunakan ERD (*Entity Relationship Diagram*).

### **3.2.4. Tahap Konstruksi Sistem**

Tahap konstruksi adalah tahap menerjemahkan hasil pada tahap desain sistem ke dalam kode-kode program komputer. Pada tahap ini akan digunakan beberapa perangkat lunak, antara lain :

- a. *JavaSpring*
- b. *MySQL*
- c. *Android*
- d. *Netbeans*

### **3.2.5. Tahap Pengujian Sistem**

Pengujian perangkat lunak, mengukur efisiensi dan efektifitas alur logika pemrograman yang dirancang dengan menggunakan pengujian *White Box Testing*

dan *Black Box Testing*. *White Box Testing* menguji perangkat lunak yang telah selesai dirancang kemudian di uji dengan cara : bagan alir (*flowchart*) yang dirancang sebelumnya dipetakan kedalam bentuk bagan alir kontrol (*flowgraph*) yang tersusun dari beberapa *node* dan *edge*. *Flowgraph* memudahkan penentuan jumlah *region*, *cyclomatic complexity* (CC), dan apabila *independent path* sama besar, maka sistem dinyatakan benar. Tetapi jika sebaliknya, maka sistem masih memiliki kesalahan.

Sedangkan *Black Box Testing* memfokuskan pada keperluan fungsional dari perangkat lunak. *Black Box Testing* merupakan alternatif dari *White Box Testing*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya. *Black Box Testing* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

- a. Kesalahan *interface*
- b. Kesalahan dalam struktur data atau akses basis data eksternal
- c. Kesalahan performa
- d. Kesalahan inisialisasi dan terminasi

### **3.2.6. Tahap Implementasi Sistem**

Sistem yang dibangun akan diimplementasikan di Dinas Pendidikan Nasional Kota Gorontalo.

### **3.2.7. Tahap Pemeliharaan Sistem**

Sistem yang telah diimplementasikan kemudian akan dievaluasi kelayakannya dan akan dilakukan pemeliharaan (*maintenance*) secara berkala baik terjadi terjadi kerusakan terhadap sistem maupun tidak.

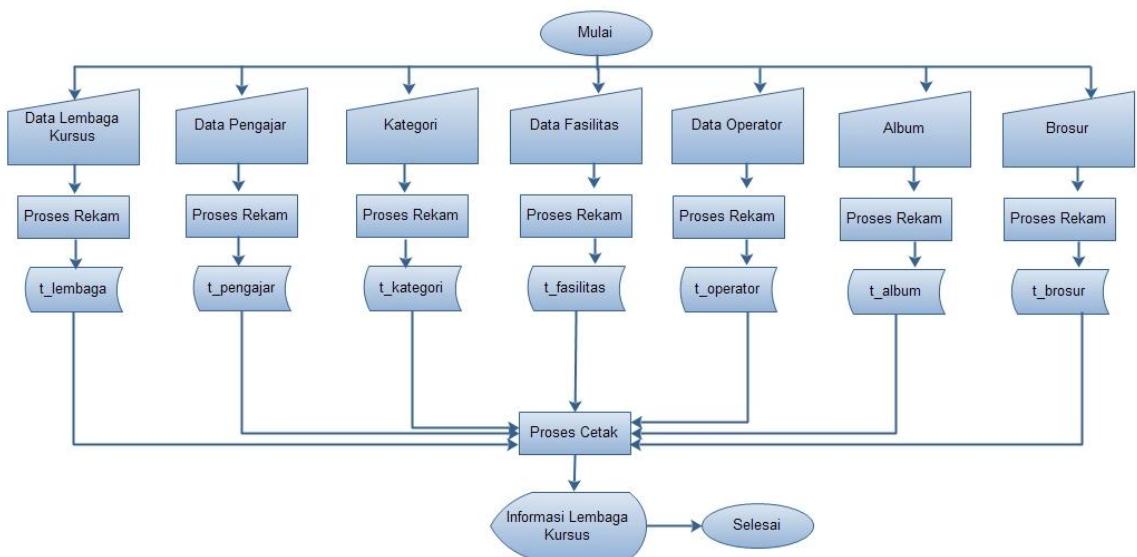
## BAB IV

# ANALISA DAN DESAIN SISTEM

### 4.1. Analisa Sistem

Analisa Sistem (*System Analysis*) adalah penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikan. Analisis merupakan tahap awal dalam pengembangan perangkat lunak sistem, dimana ahli teknik sistem menganalisis hal-hal yang diperlukan dalam melaksanakan proyek pembuatan atau pengembangan perangkat lunak.

#### 4.1.1. Sistem Yang Diusulkan

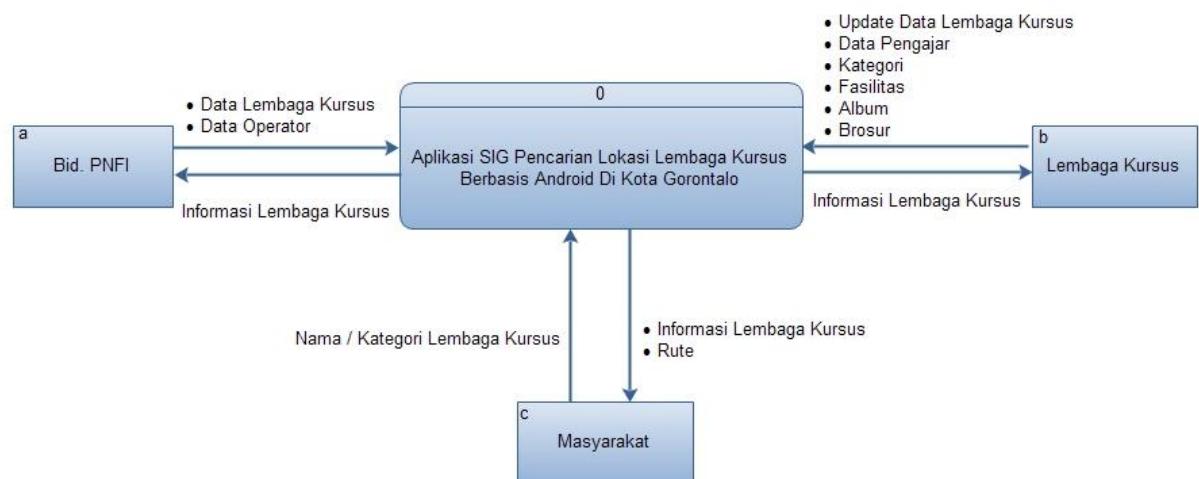


Gambar 4.1 Bagan Alir Sistem Yang Diusulkan

## 4.2. Desain Sistem

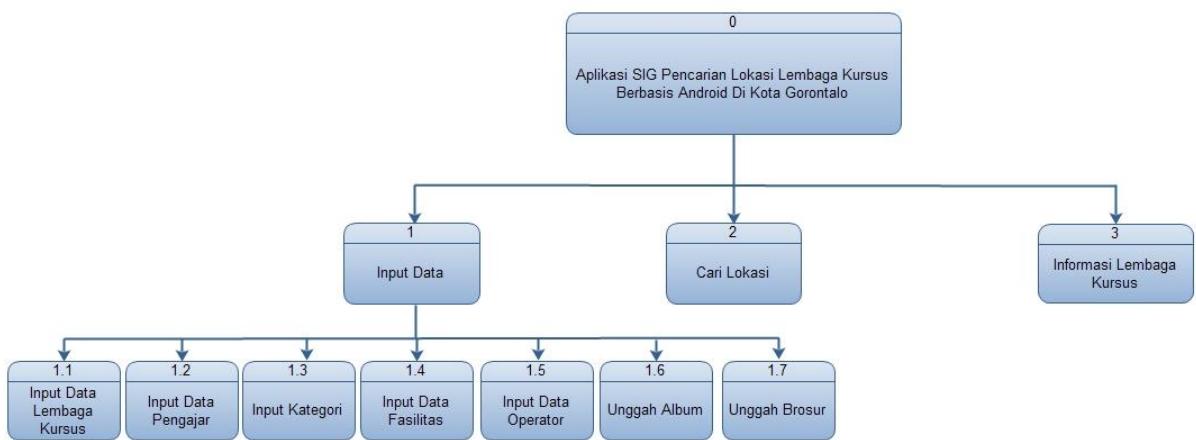
### 4.2.1. Desain Sistem Secara Umum

#### 4.2.1.1. Diagram Konteks



Gambar 4.2 Diagram Konteks

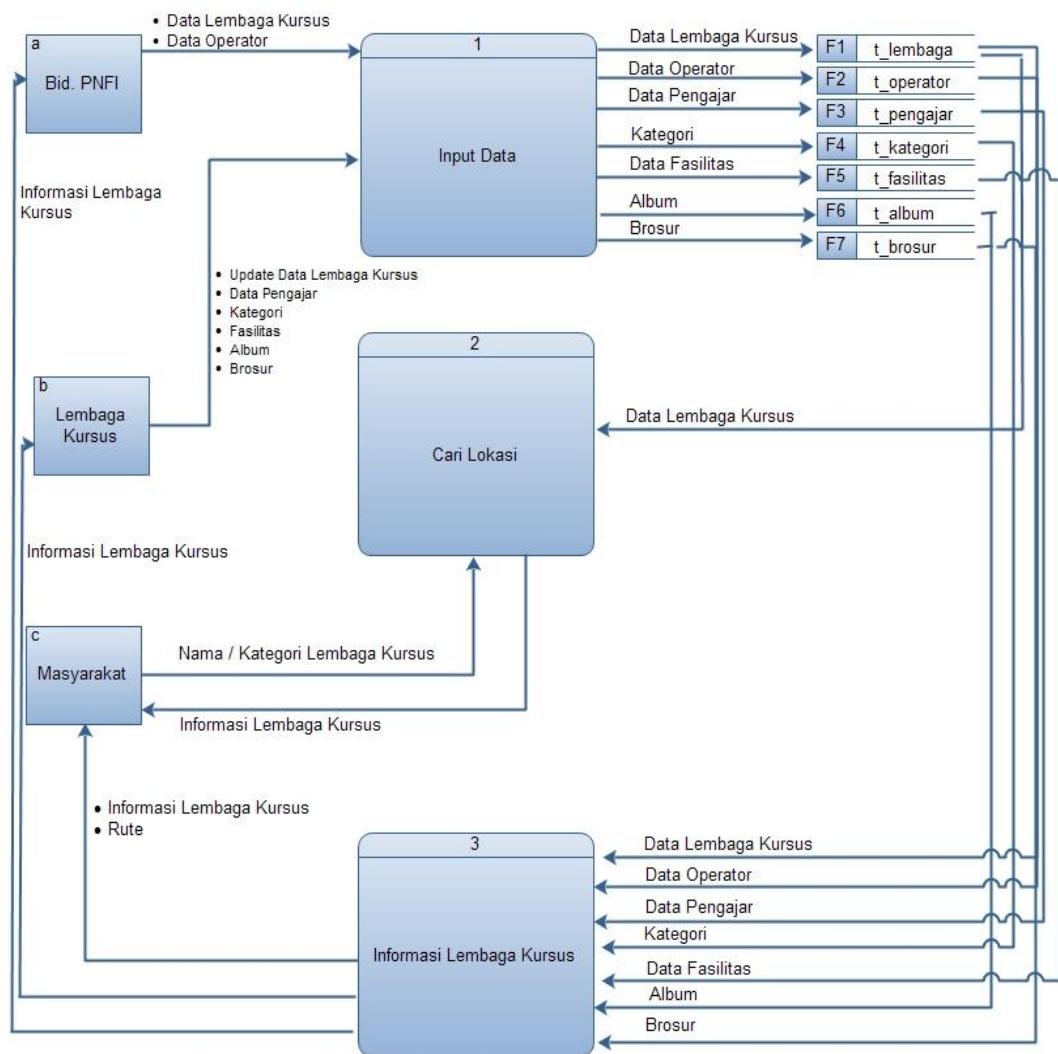
#### 4.2.1.2. Diagram Berjenjang



Gambar 4.3 Diagram Berjenjang

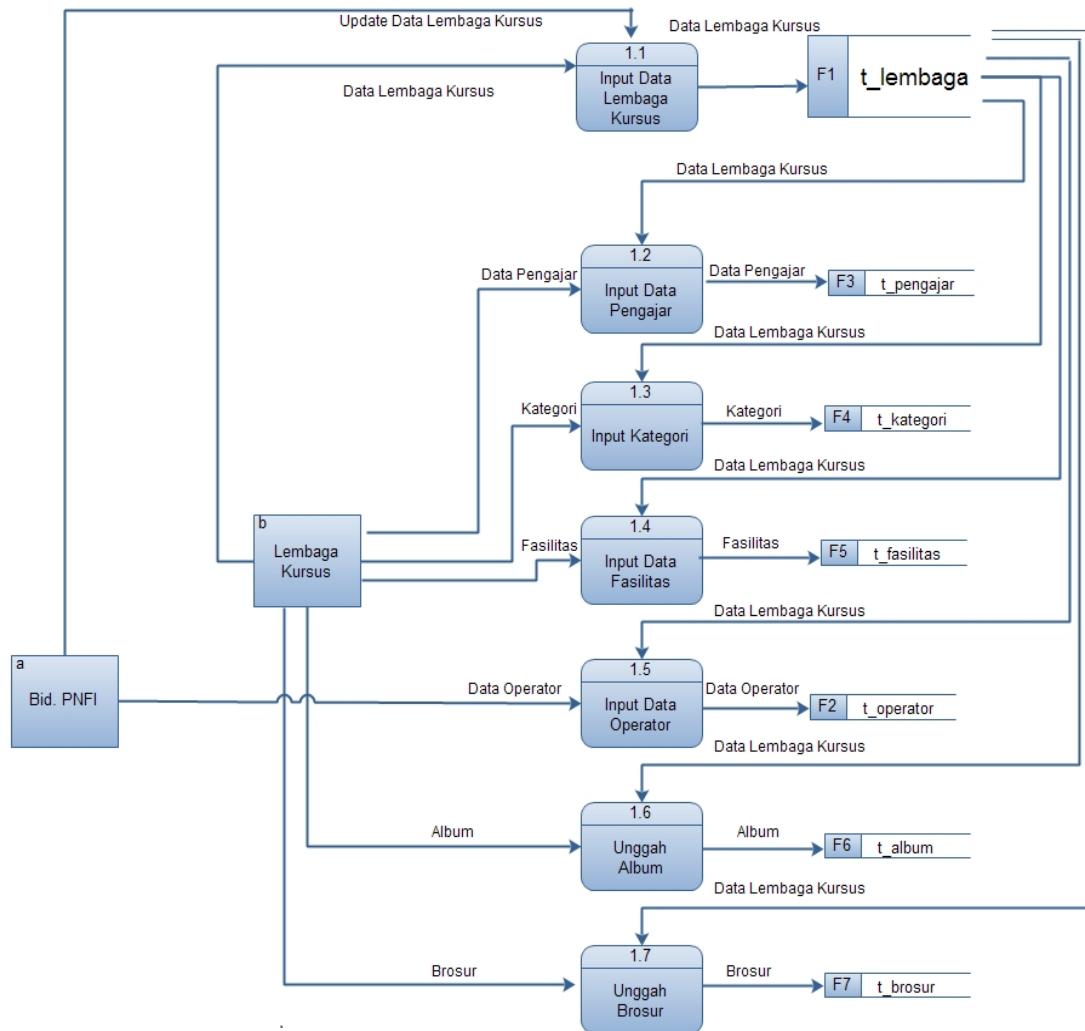
### 4.2.1.3. Diagram Arus Data (DAD)

#### 4.2.1.3.1. DAD Level 0



Gambar 4.4 DAD Level 0

#### 4.2.1.3.2. DAD Level 1



**Gambar 4.5 DAD Level 1**

#### 4.2.1.4. Kamus Data

Kamus data atau *Data Dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data digunakan untuk merancang *input*, *file-file/database* dan *output*. Kamus data

dibuat berdasarkan arus data yang mengalir pada DAD, dimana didalamnya terdapat struktur dari arus data secara detail.

**Tabel 4.1** Kamus Data Lembaga Kursus

<b>Kamus Data : t_lembaga</b>				
Nama Arus Data : Data Lembaga Kursus		Bentuk Data : Dokumen		
Penjelasan : Berisi data-data Lembaga Kursus		Arus Data : a-1,1-F1,F1-2,F1-3,a-1.1,1.1-F1		
Periode : Setiap ada penambahan data lembaga kursus (non periodik)				
Struktur Data :				
No	Nama Item Data	Type	Width	Description
1.	id	C	10	Id lembaga kursus
2.	alamat_lkp	C	100	Alamat Lembaga Kursus
3.	biaya	C	-	Biaya Kursus
4.	description	C	-	Deskripsi tentang Lembaga Kursus
5.	facebook	C	50	Akun Facebook Lembaga Kursus
6.	instagram	C	50	Akun Instagram Lembaga Kursus
7.	jam_pelayanan	N	10	Jam pelayanan lembaga kursus
8.	ketua	C	30	Nama ketua Lembaga Kursus
9.	lat	N	15	Titik koordinat lokasi
10.	longi	N	15	Titik koordinat lokasi
11.	nama	C	50	Nama Lembaga Kursus
12.	No_telp	N	30	No telepon lembaga kursus
13.	Program	C	-	Program Kursus

**Tabel 4.2** Kamus Data Operator

<b>Kamus Data : t_operator</b>				
Nama Arus Data	: Data Operator			Bentuk Data :
Penjelasan	: Berisi data-data operator			Dokumen
Periode	: Setiap ada penambahan data operator (non periodik)			Arus Data : a-1,1-F2,F2-3,a-1.5,1.5-F2
Struktur Data	:			
No	Nama Item Data	Type	Width	Description
1.	username	C	20	Nama Operator
2.	password	C	25	Kata sandi operator
3.	lembaga_id	C	10	Id lembaga kursus

**Tabel 4.3** Kamus Data Pengajar

<b>Kamus Data : t_pengajar</b>				
Nama Arus Data	: Data Pengajar			Bentuk Data :
Penjelasan	: Berisi data-data pengajar			Dokumen
Periode	: Setiap ada penambahan data pengajar (non periodik)			Arus Data : b-1,1-F3,F3-3,b-1.2,1.2P-F3
Struktur Data	:			
No	Nama Item Data	Type	Width	Description
1.	id	C	10	Id pengajar
2.	jabatan	C	20	Jabatan
3.	nama	C	30	Nama pengajar
4.	lembaga_id	C	10	Id lembaga kursus

**Tabel 4.4** Kamus Data Kategori

<b>Kamus Data : t_kategori</b>				
No	Nama Item Data	Type	Width	Description
1.	id_kategori	C	10	Id kategori
2.	id_lembaga	C	10	Id lembaga kursus

**Tabel 4.5** Kamus Data Fasilitas

<b>Kamus Data : t_fasilitas</b>				
No	Nama Item Data	Type	Width	Description
1.	id	C	5	Id fasilitas
2.	jumlah	C	10	Jumlah fasilitas
3.	nama	C	30	Nama fasilitas
4.	lembaga_id	C	10	Id lembaga kursus

**Tabel 4.6** Kamus Data Album

<b>Kamus Data : t_album</b>				
No	Nama Item Data	Type	Width	Description
1.	Id	C	5	Id album
2.	nama	C	30	Judul Album
3.	nama_file	C	30	Nama foto
4.	lembaga_id	C	10	Id lembaga kursus

**Tabel 4.7** Kamus Data Brosur

<b>Kamus Data : t_brosur</b>				
No	Nama Item Data	Type	Width	Description
1.	id	C	5	Id brosur
2.	nama_brosur	C	10	Judul Album
3.	nama_file	C	30	Nama file brosur
4.	lembaga_id	C	10	Id lembaga kursus

#### 4.2.1.5. Desain *Input* Secara Umum

##### **Desain Input Secara Umum**

- Untuk** : Dinas Pendidikan Kota Gorontalo  
**Sistem** : Pencarian Lokasi Lembaga Kursus Berbasis Android  
**Tahap** : Perancangan Sistem Secara Umum

**Tabel 4.8** Desain Input Secara Umum

Kode Input	Nama Input	Sumber	Tipe File	Periode
001	Data Lembaga Kursus	Lembaga Kursus	Indeks	Non Periodik
002	Data Operator	Bid. PNFI	Indeks	Non Periodik
003	Data Pengajar	Lembaga Kursus	Indeks	Non Periodik
004	Data Kategori	Lembaga Kursus	Indeks	Non Periodik
005	Data Fasilitas	Lembaga Kursus	Indeks	Non Periodik
006	Data Album	Lembaga Kursus	Indeks	Non Periodik
007	Data Brosur	Lembaga Kursus	Indeks	Non Periodik

#### 4.2.1.6. Desain Database Secara Umum

##### Desain Database Secara Umum

- Untuk** : Dinas Pendidikan Kota Gorontalo  
**Sistem** : Pencarian Lokasi Lembaga Kursus Berbasis Android  
**Tahap** : Perancangan Sistem Secara Umum

**Tabel 4.9** Desain Database Secara Umum

Kode File	Nama File	Tipe File	Media File	Organisasi File	Field Kunci
F1	t_lembaga	Master	Harddisk	Indeks	Id_lembaga
F2	t_operator	Master	Harddisk	Indeks	Username
F3	t_pengajar	Master	Harddisk	Indeks	Id_pengajar
F4	t_kategori	Master	Harddisk	Indeks	Id_kategori
F5	t_fasilitas	Master	Harddisk	Indeks	Id_fasilitas
F6	t_album	Master	Harddisk	Indeks	Id_album
F7	t_brosur	Master	Harddisk	Indeks	Id_brosur

#### 4.2.1.7. Desain Output Secara Umum

**Tabel 4.10** Desain Output Secara Umum

Kode Output	Nama Output	Format	Media Output	Distribusi	Periode
001	Data Lembaga Kursus	Teks	Monitor	Bid. PNFI dan Lembaga Kursus	Non Periodik

002	Profil Lembaga Kursus	Teks dan Gambar	<i>Smartphone</i> dan Monitor	Masyarakat	Non Periodik
003	Rute Lembaga Kursus	Gambar	<i>Smartphone</i>	Masyarakat	Non Periodik

#### 4.2.2. Desain Sistem Secara Terinci

##### 4.2.2.1 Desain *Input* Terinci

The screenshot shows a user interface for adding a new organization. At the top, there is a map with a green polygon representing the organization's location. Below the map, there is a section titled "Data Lembaga" containing four input fields: "Latitude", "Longitude", "Nama", and "Ketua". At the bottom of the form are two buttons: "< Kembali" (Back) and "Simpan" (Save).

**Gambar 4.6** Desain Input Data Lembaga Kursus

Struktur data form ini adalah *String*. Tipe data latitude, longitude, nama, dan ketua adalah *varchar*. Terdapat *button* Kembali untuk kembali ke halaman data lembaga dan Simpan untuk menyimpan data yang telah di *input*.

Tambah Operator

Isi Data		< Kembali
Nama Operator	<input type="text"/>	
Username	<input type="text"/>	
Password	<input type="text"/>	
Role	<input type="text"/>	
Lembaga	<input type="text"/>	
	<b>xxxxxxxxxxxxxx</b>	
<b>Simpan</b>		

**Gambar 4.7** Desain Input Data Operator

Struktur data form tambah operator adalah *String* dan tipe data dari masing-masing field adalah *varchar*. Di *combobox* Lembaga akan tampil nama-nama lembaga kursus yang sudah tersimpan di sistem, sehingga admin tinggal memilih akan menambahkan operator dari lembaga kursus mana. *Button* simpan untuk menyimpan data yang sudah di *input* ke form.

Tambah Pengajar

Isi Data Pengajar		< Kembali
Nama	<input type="text"/>	
Jabatan	<input type="text"/>	
<b>Simpan</b>		

**Gambar 4.8** Desain Input Data Pengajar

Struktur data form ini adalah *String* dan tipe data Nama dan Jabatan adalah *varchar* serta *button* Simpan untuk menyimpan data yang sudah di *input* ke form.

### Tambah Kategori

The screenshot shows a window titled "Isi Data Kategori". In the top right corner, there is a small "Kembali" button. On the left, there is a label "Kategori" followed by a dropdown menu. The dropdown menu contains four items, all of which are represented by the text "xxxxxxxxxx" in a monospaced font. At the bottom of the window is a blue rectangular button labeled "Simpan".

**Gambar 4.9** Desain Input Data Kategori

Struktur data form tambah kategori adalah *String* dan tipe data dari masing-masing field adalah *varchar*. Di dalam *combo box* akan tampil berbagai macam kategori, operator tinggal memilih dengan meng-klik kategori yang dimaksud. *Button Simpan* untuk menyimpan data yang sudah di *input* ke form.

### Tambah Fasilitas

The screenshot shows a window titled "Isi Data Fasilitas". In the top right corner, there is a small "Kembali" button. The main area contains two text input fields. The first field is labeled "Fasilitas" and the second is labeled "Jumlah", both preceded by a horizontal line for input. At the bottom of the window is a blue rectangular button labeled "Simpan".

**Gambar 4.10** Desain Input Data Fasilitas

Struktur data form ini adalah *String* dan tipe data Fasilitas dan Jumlah adalah *varchar* serta *button Simpan* untuk menyimpan data yang sudah di *input* ke form.

### Tambah Album

Upload Foto < Kembali

Nama Foto:

File:

**Gambar 4.11** Desain Input Tambah Album

Struktur data form tambah album adalah *String*. Tipe data Nama Foto dan File adalah *varchar*. Dalam field File diberi *button* Choose File untuk memasukkan gambar. Serta *button* Upload untuk upload foto yang sudah dipilih.

### Brosur

Foto Brosur

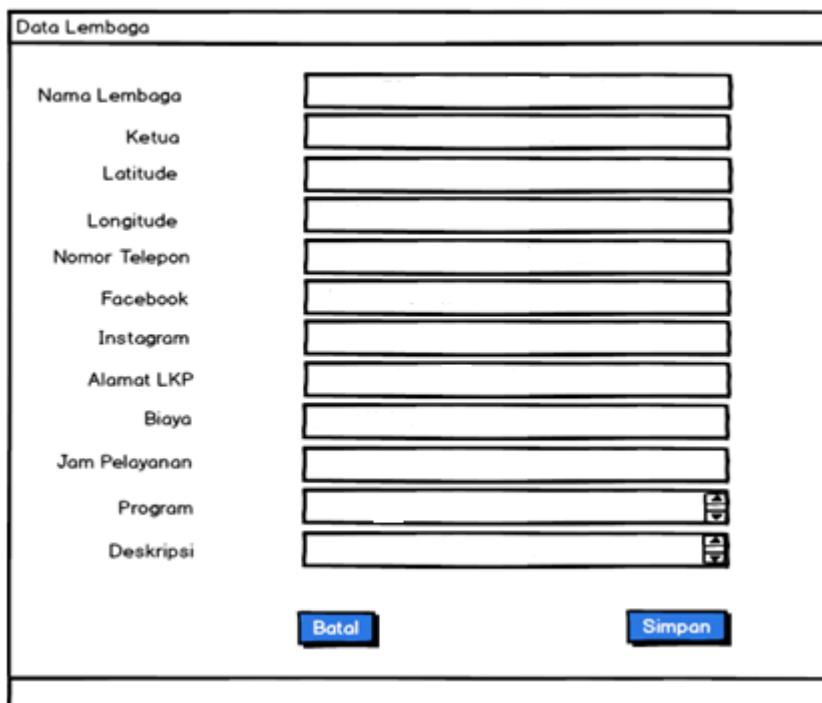
Nama Brosur:

File:

**Gambar 4.12** Desain Input Upload Brosur

Struktur data form Brosur adalah *String*. Tipe data Nama Brosur dan File adalah *varchar*. Sama seperti form upload album, di dalam field File diberi *button* Choose File untuk memasukkan gambar. Serta *button* Upload untuk upload foto brosur yang sudah dipilih.

## Profile



The image shows a user interface for editing a profile. At the top, it says "Data Lembaga". Below that is a table with 11 rows, each containing a label and an input field. The labels are: Nama Lembaga, Ketua, Latitude, Longitude, Nomor Telepon, Facebook, Instagram, Alamat LKP, Biaya, Jam Pelayanan, Program, and Deskripsi. The "Program" and "Deskripsi" fields have small dropdown arrows at their right ends. At the bottom left is a blue "Batal" button, and at the bottom right is a blue "Simpan" button.

**Gambar 4.13** Desain Input Edit Profil Lembaga Kursus

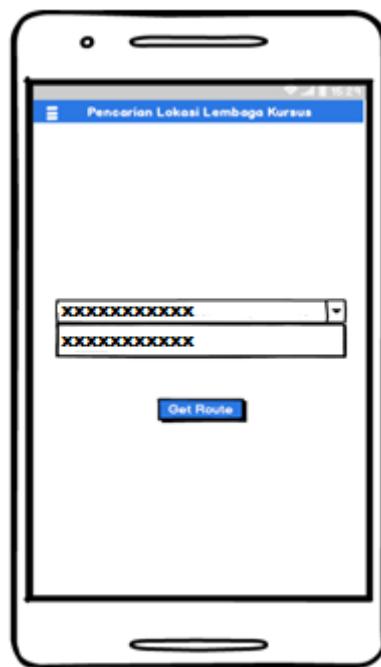
Struktur data form Profile adalah *String*. Tipe data *field* Biaya, Program dan Deskripsi adalah *longtext* sedangkan tipe data *field* lainnya adalah *varchar*. Terdapat *button* Batal untuk membatalkan *update* Profil lembaga kursus dan Simpan untuk menyimpan data yang telah di *update*.



**Gambar 4.14** Desain Pencarian Lokasi Berdasarkan Nama



**Gambar 4.15** Desain Input Pencarian Lokasi Berdasarkan Kategori



**Gambar 4.16** Desain Input Pencarian Rute Lembaga Kursus

#### 4.2.2.2 Desain Database Terinci

**Tabel 4.11** Tabel Data Lembaga Kursus

Nama File : t_lembaga Tipe File : Induk Organisasi : Indeks				
No	Field Name	Type	Width	Indeks
1.	Id	Varchar	10	Primary Key
2.	Alamat_LKP	Varchar	100	
3.	Biaya	Longtext	-	
4.	Description	Longtext	-	
5.	Facebook	Varchar	50	
6.	Instagram	Varchar	50	
7.	Ketua	Varchar	30	
8.	Lat	Varchar	15	
9.	Long	Varchar	15	

10.	Nama	Varchar	50	
11.	No_telp	Varchar	30	
12.	Program	Longtext	-	
13.	Jam_pelayanan	Varchar	10	

**Tabel 4.12** Tabel Data Operator

Nama File : t_user Tipe File : Induk Organisasi : Indeks				
No	Field Name	Type	Width	Indeks
1.	Id	Varchar	10	Primary Key
2.	Default_pass	Varchar	20	
3.	Display_name	Varchar	20	
4.	Enable	Bit	1	
5.	Password	Varchar	20	
6.	Username	Varchar	20	

**Tabel 4.13** Tabel Data Pengajar

Nama File : t_pengajar Tipe File : Induk Organisasi : Indeks				
No	Field Name	Type	Width	Indeks
1.	Id	Varchar	10	Primary Key
2.	Jabatan	Varchar	20	
3.	Nama	Varchar	30	
4.	Lembaga_id	Varchar	10	

**Tabel 4.14** Tabel Kategori

Nama File	:	t_kategori		
Tipe File	:	Induk		
Organisasi	:	Indeks		
No	Field Name	Type	Width	Indeks
1.	Id_kategori	Varchar	3	Primary Key
2.	Id_Lembaga	Varchar	10	

**Tabel 4.15** Tabel Fasilitas

Nama File	:	t_fasilitas		
Tipe File	:	Induk		
Organisasi	:	Indeks		
No	Field Name	Type	Width	Indeks
1.	Id	Varchar	5	Primary Key
2.	Jumlah	Varchar	10	
3.	Nama	Varchar	30	
4.	Lembaga_id	Varchar	10	

**Tabel 4.16** Tabel Album

Nama File	:	t_album		
Tipe File	:	Induk		
Organisasi	:	Indeks		
No	Field Name	Type	Width	Indeks
1.	Id	Varchar	5	Primary Key
2.	Nama	Varchar	30	
3.	Nama_file	Varchar	30	
4.	Lembaga_id	Varchar	10	

**Tabel 4.17** Tabel Brosur

Nama File	:	t_brosur		
Tipe File	:	Induk		
Organisasi	:	Indeks		
No	Field Name	Type	Width	Indeks
1.	Id	Varchar	5	Primary Key
2.	Nama_brosur	Varchar	10	
3.	Nama_file	Varchar	30	
4.	Lembaga_id	Varchar	10	

#### 4.2.2.3 Desain *Output* Terinci

##### Daftar Lembaga

Data Lembaga				+ Tambah Lembaga
Lembaga	No. Telepon	Ketua	Action	
Kids 2 Success	0435 - 8712999	Ventriwati	<a href="#">Hapus</a>	<a href="#">Edit</a>
MY LIBERTY International English School	085242999943/0435-8525515	Muhammad Busyaeri Jafar, SH.	<a href="#">Hapus</a>	<a href="#">Edit</a>

**Gambar 4.17** Desain Output Daftar Lembaga Kursus

## Daftar Operator

Data Operator				+ Tambah Operator
Nama Operator	Username	Lembaga	Action	
Ringgo	Ringgo	MY LIBERTY International English School	<a href="#">Hapus</a>	
Ventri	Ventri	Kids 2 Success	<a href="#">Hapus</a>	

Gambar 4.18 Desain Output Daftar Operator

## Daftar Kategori

Data Kategori		+ Tambah
Kategori	Action	
Bahasa Inggris	<a href="#">Hapus</a>	
Bimbingan Belajar	<a href="#">Hapus</a>	

Gambar 4.19 Desain Output Daftar Kategori

## Daftar Fasilitas

Data Fasilitas			+ Tambah
Fasilitas	Jumlah	Action	
LCD	4	<a href="#">Hapus</a>	<a href="#">Edit</a>
Ruang Kelas	6	<a href="#">Hapus</a>	<a href="#">Edit</a>

**Gambar 4.20** Desain Output Daftar Fasilitas

## Daftar Pengajar

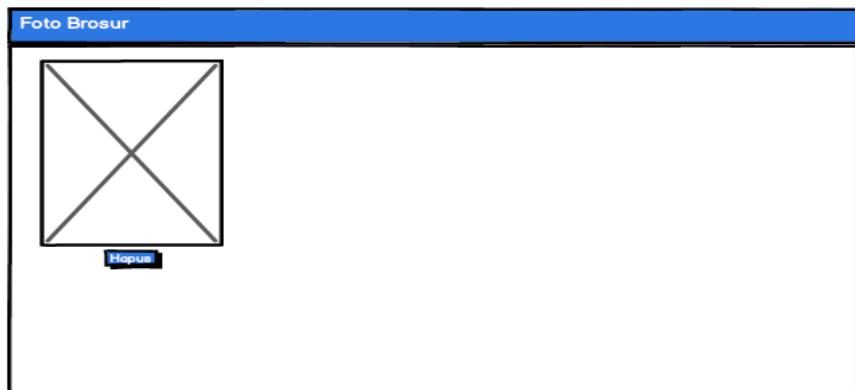
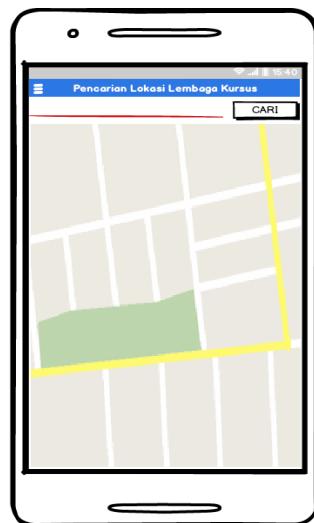
Data Pengajar			+ Tambah
Nama	Jabatan	Action	
Brother Redhy	Instruktur	<a href="#">Hapus</a>	<a href="#">Edit</a>
Sister Dewi	Instruktur	<a href="#">Hapus</a>	<a href="#">Edit</a>

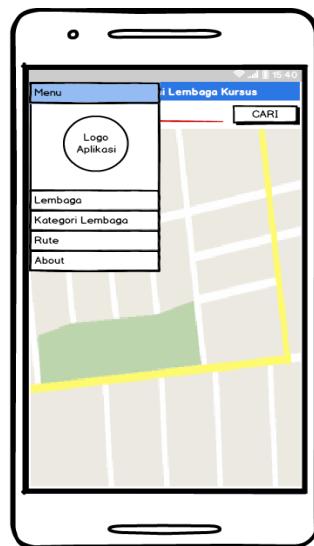
**Gambar 4.21** Desain Output Daftar Pengajar

## Album

Daftar Album Foto								+ Tambah
<a href="#">Hapus</a>								
<a href="#">Hapus</a>								

**Gambar 4.22** Desain Output Daftar Album

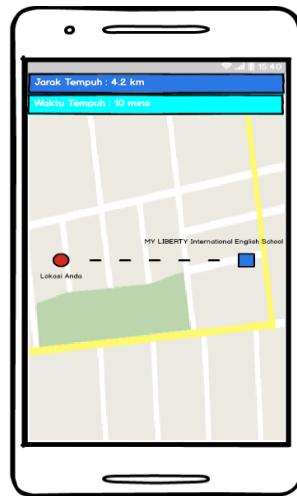
**Brosur****Gambar 4.23** Desain Output Foto Brosur**Gambar 4.24** Desain Output Tampilan Awal Aplikasi *Mobile*



**Gambar 4.25** Desain Output Menu

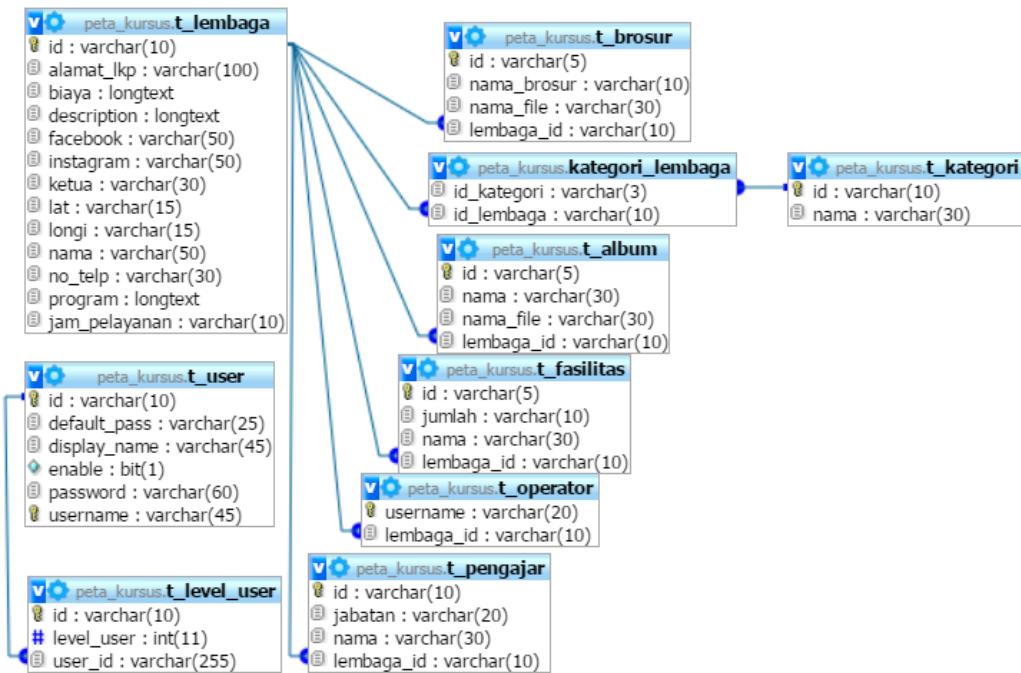


**Gambar 4.26** Desain Output Profil Lembaga Kursus



**Gambar 4.27 Desain Output Rute Lembaga Kursus**

#### 4.2.3. Desain Relasi Antar Tabel



**Gambar 4.28 Relasi Antar Tabel**

#### 4.2.4. Desain Menu Utama

Home	Bid. PNFI	Lembaga Kursus	Masyarakat
	Daftar Lembaga Kursus Daftar Operator	Kategori Pengajar Fasilitas Album Brosur Profil Lembaga Kursus	Cari Informasi Lembaga Kursus

**Gambar 4.29** Desain Menu Utama

## **BAB V**

### **HASIL PENELITIAN DAN PEMBAHASAN**

#### **5.1. Hasil Penelitian**

##### **5.1.1 Gambaran Umum Lokasi Penelitian**

Berdasarkan Latar Belakang dan Kerangka Pemikiran seperti yang telah diuraikan dalam Bab I dan Bab II, maka yang menjadi objek penelitian adalah **Pencarian Lokasi Lembaga Kursus.**

###### **5.1.1.1 Dinas Pendidikan Nasional Kota Gorontalo**

Awalnya di Kota Gorontalo Tahun 1975 ada 4 (empat) unit Kantor Pembinaan (KABIN) untuk lingkungan/jajaran Pendidikan dan Kebudayaan sebagai berikut :

1. Kabin Pendidikan Dasar dan Luar Biasa (PLPB)
2. Kabin Pendidikan dan Olahraga dan Generasi Muda
3. Kabin Pendidikan Masyarakat
4. Kabin Kebudayaan

Selanjutnya berdasarkan Keputusan Menteri Pendidikan dan Kebudayaan RI Nomor : 0134/0/1975 Kabin-Kabin tersebut diubah menjadi Kantor Departemen Pendidikan dan Kebudayaan Kotamadya Gorontalo termasuk koordinasi SLTP/SLTA di bawah pimpinan Bapak H.Dj. Rachman.

Di Kecamatan dikenal dengan Kantor Depdikbud Kecamatan yang terdiri dari :

Kantor Depdikbud Kec. Kota Selatan

1. Kantor Depdikbud Kec. Kota Utara
2. Kantor Depdikbud Kec. Kota Barat, yang masing-masing di Kepala oleh Kepala Kantor Depdikbud Kecamatan

Untuk Kantor Depdikbud Kotamadya Gorontalo dipimpin oleh Kepala Kantor Departemen Dikbud yang dibawahnya Sub Bagian Tata Usaha dan 4 (empat) seksi yaitu :

1. Seksi Pendidikan Dasar
2. Seksi Pendidikan Masyarakat
3. Seksi Pembinaan Generasi Muda dan Ke Olahragaan (Binmudora)
4. Seksi Kebudayaan

Pada bulan Agustus 1983 mengalami perubahan sesuai SK Mendikbud RI No. 0173/0/1983 mempunyai tugas yaitu :

1. Memberi pelayanan teknis dan administrasi kepada semua unsur di lingkungan Kantor Depdikbud Koatamdy Gorontalo
2. Membina dan mengurus sekolah TK/SD dan usaha wajib belajar serta sekolah luar biasa
3. Membina dan mengurus Pendidikan Masyarakat dalam kegiatan pembinaan generasi muda dan pembinaan kegiatan keolahragaan
4. Membina dan mengurus kegiatan pengembangan kebudayaan.

Dalam Keputusan ini Kantor Depdikbud Kotamadya Gorontalo terbagi 5 Sub Bagian dan 4 Seksi yaitu :

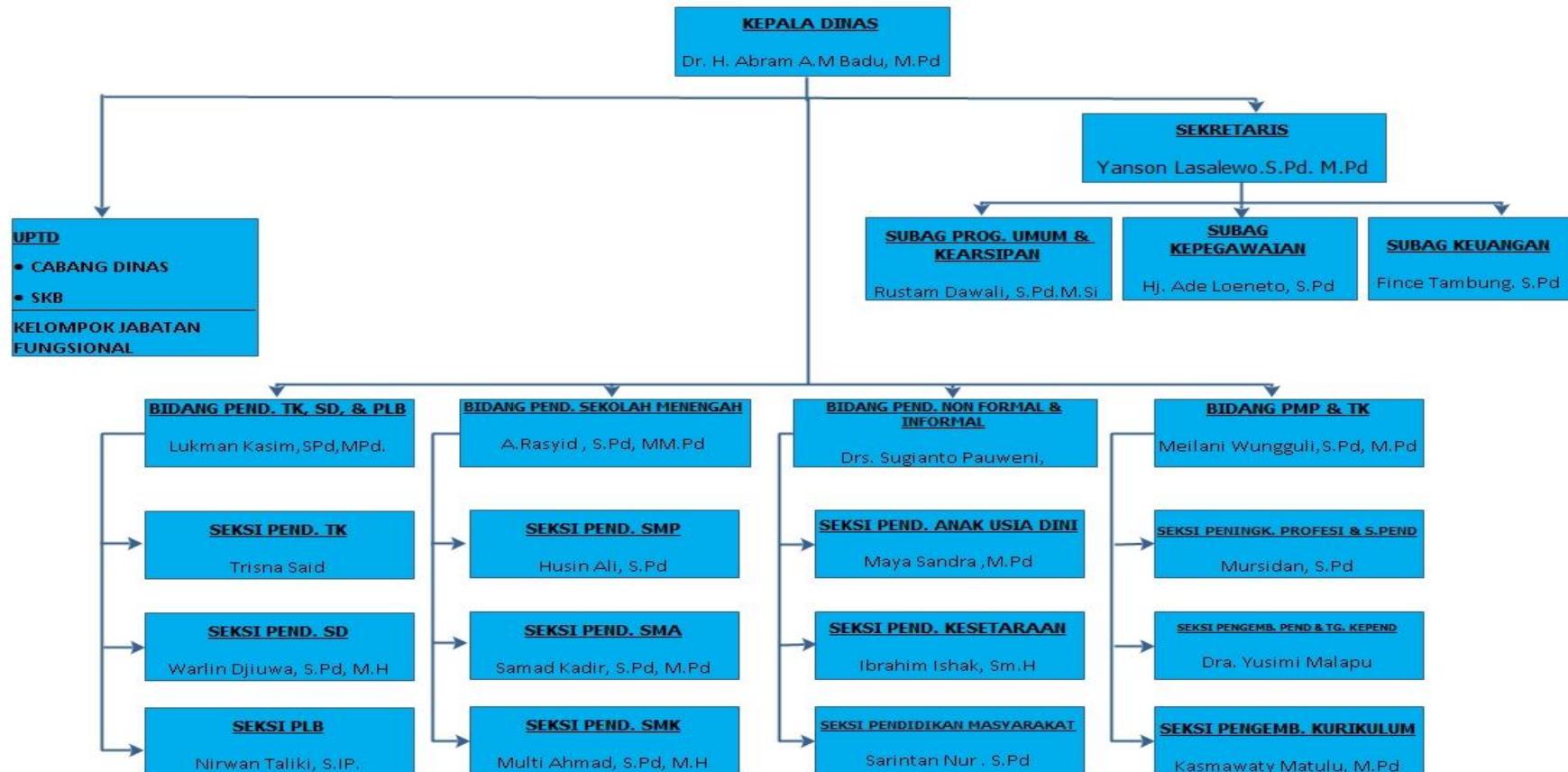
1. Sub Bagian Tata Usaha
  - a. Urusan Persuratan
  - b. Urusan Rumah Tangga
2. Sub Bagian Penyusunan Rencana dan Program
  - a. Urusan Pengumpulan Data
  - b. Urusan Penyusunan Program
  - c. Urusan Monitor pelaksanaan Rencana dan Program
3. Sub Bagian Kepegawaian
  - a. Urusan KPA
  - b. Urusan KPE
  - c. Urusan Pemensiun
4. Sub Bagian Keuangan
  - a. Urusan Penyusunan Anggaran
  - b. Urusan Pelaksanaan Anggaran
  - c. Urusan Monitoring Pelaksanaan Anggaran
5. Sub Bagian Perlengkapan
  - a. Urusan Gedung dan Perabot
  - b. Urusan Peralatan Teknis dan Perbukuan
  - c. Urusan Inventaris dan Penghapusan Barang
6. Seksi Pendidikan Dasar
7. Seksi Pendidikan Masyarakat
8. Seksi Pembinaan Pemuda dan Ke Olahragaan
9. Seksi Kebudayaan

Kantor Depdikbud Kecamatan mempunyai tugas untuk melaksanakan sebagian tugas Kandep Dikbud Koatamdy Gorontalo.

Sejak bulan Januari 2001 mengacu pada UU No. 22 Tahun 1999 tentang Pemerintahaan Daerah dan Peraturan Pemerintah RI No. 84 Tahun 2000 tentang Pedoman Organisasi perangkat daerah, maka Dinas Pendidikan dan Kebudayaan Kota Gorontalo berdasarkan Perda No. 19 Tahun 2000 tentang Organisasi dan Tata Kerja Dinas Daerah Kota Gorontalo.

Selanjutnya dengan telah ditetapkan PP No. 8 Tahun 2003 sebagai pengganti PP No. 84 Tahun 2000 tentang Pedoman Organisasi Perangkat Daerah, maka Perda No. 19 Tahun 2000 tentang Organisasi dan Tata Kerja Dinas Pendidikan dan Kebudayaan Kota Gorontalo tidak sesuai lagi dengan keadaan dan perkembangan. Penataan organisasi kelembagaan pada Pemerintah Daerah Kota Gorontalo disesuaikan dengan penerapan peraturan Pemerintah No. 8 Tahun 2003, dan susunan organisasi dan tata kerja Dinas Pendidikan dan Kebudayaan Kota Gorontalo di bentuk dan di tetapkan dengan Peraturan Daerah Kota Gorontalo Nomor 25 Tahun 2004. Sesuai Perda Nomor 5 Tahun 2009 Dinas Pendidikan dan Kebudayaan Kota Gorontalo berubah Nomenklatur menjadi Dinas Pendidikan Kota Gorontalo sampai dengan sekarang.

### 5.1.1.2 Struktur Organisasi



Gambar 5.1 Struktur Organisasi Dinas Pendidikan Kota Gorontalo

### **5.1.1.3 Job Deskripsi**

#### **1. KEPALA DINAS**

Kepala DINAS PENDIDIKAN mempunyai tugas :

Melaksanakan sebagian tugas Pemerintah Daerah dibidang pengembangan Pendidikan berdasarkan peraturan perundang-undangan untuk kelancaran tugas.

Dalam melaksanakan tugas Kepala Dinas Pendidikan menyelenggarakan fungsi :

- a. Merencanakan pengembangan pendidikan secara berkesinambungan untuk pemantapan sistim pendidikan yang berkualitas
- b. Merumuskan kebijakan pengembangan pendidikan berdasarkan skala prioritas untuk peningkatan SDM
- c. Menyusun kebijakan teknis pengembangan pendidikan secara berjenjang sebagai dasar pelaksanaan tugas unit.
- d. Mengorganisir pengembangan pendidikan melalui mekanisme / prosedur kerja dalam suatu unit untuk tertibnya pelaksanaan tugas
- e. Mengendalikan pelaksanaan sistem pengembangan pendidikan secara terpadu untuk peningkatan mutu pendidikan
- f. Mengarahkan pelaksanaan program sistem pengembangan mutu pendidikan melalui proses belajar mengajar yanmg inovatif dan kreatif untuk peningkatan mutu pendidikan
- g. Membina pelaksanaan kegiatan unit secara menyeluruh untuk kelancaran tugas

- h. Mendistribusikan tugas sesuai bidang masing-masing untuk kelancaran pelaksanaan tugas unit.
- i. Mengawasi pelaksanaan tugas baik intern dan ekstern secara berkala untuk efektifitas dan efisiensi kegiatan unit.
- j. mengevaluasi seluruh kegiatan unit secara terpadu untuk mengetahui perkembangan pelaksanaan tugas
- k. Mengkonsultasikan dan mengkoordinasikan pelaksanaan tugas dengan unit terkait melalui rapat koordinasi untuk penyatuan pendapat.
- l. Melaporkan hasil pelaksanaan tugas unit secara berkala sebagai bahan evaluasi
- m. Melakukan tugas pembantuan dan tugas-tugas lainnya yang diperintahkan oleh atasan untuk kelancaran tugas kedinasan.

## **2. SEKRETARIS**

Sekretaris mempunyai tugas :

Melaksankan tugas penatausahaan dibidang perencanaan, program dan keuangan, pelayanan kepegawaian, umum dan karsipan berdasarkan peraturan perundang-undangan untuk kelancaran pelaksanaan tugas.

Dalam melaksanakan tugas sekretaris menyelenggarakan fungsi :

- a. Menghimpun kebijakan teknis dibidang penyusunan program, kepegawaian, umum dan karsipan sesuai kebutuhan sebagai dasar pelaksanaan tugas
- b. Menyusun rencana kegiatan sesuai kebutuhan untuk menjadi program unit

- c. Melaksanakan tugas pengelolaan administrasi berdasarkan pedoman untuk peningkatan pelayanan
- d. Melaksanakan tugas pengelolaan kepegawaian berdasarkan petunjuk pelaksanaan/petunjuk teknis untuk tertibnya penataan admnistrasi kepegawaian
- e. Melaksanakan tugas pengelolaan perlengakapan dan kearsipan sesuai kebutuhan untuk kelancaran kegiatan unit.
- f. Melakukan pembinaan pegawai secara berkala untuk peningkatan kinerja aparatur
- g. Mengkonsultasikan tugas dengan atasan secara lisan maupun tulian untuk beroleh petujuk
- h. Mengkoordinasikan pelaksanaan tugas dengan Kepala-Kepala Bidang melalui rapat/pertemuan untuk penyatuan pendapat
- i. Mendistribusikan tugas kepada bawahan sesuai job untuk tertibnya pelaksanaan tugas.
- j. Menyusun laporan pelaksanaan tugas secara berkala sebagai bahan evaluasi
- k. Melaksanakan tugas lain yang diperintahkan oleh atasan untuk kelancaran tugas kedinasan

### **3. BIDANG PENDIDIKAN TK, SD DAN PLB**

Melaksanakan tugas dibidang pengembangan pendidikan serta Kurikulum TK, SD dan PLB berdasarkan Peraturan perundang-undangan untuk pelancaran pelaksanaan tugas.

Dalam melaksanakan tugas Bidang TK, SD dan PLB menyelenggarakan fungsi :

- a. Menghimpun kebijakan teknis dibidang pendidikan TK, SD dan PLB berdasarkan kebutuhan sebagai dasar pelakanaan tugas.
- b. Menghimpun dan mengelola data pendidikan TK, SD dan PLB sesuai jenis untuk mengetahui jumlah dan perkembangannya.
- c. Menyusun rencana kegiatan dibidang TK, SD dan PLB berdasarkan Petunjuk Pelaksanaan dan Petunjuk Teknis untuk menjadi program unit
- d. Mengembangkan standar kompetensi guru TK, SD dan PLB melalui diklat untuk peingkatan sumber daya manusia
- e. Menghimpun pedoman dan kurikulum TK, SD dan PLB sesuai kebutuhan sebagai dasar pelaksanaan proses belajar mengajar
- f. Mendistribusikan pedoman pelaksanaan proses belajar mengajar melalui sekolah-sekolah sebagai bahan kurikulum di sekolah.
- g. Meningkatkan kompetensi siswa melalui lomba kreatifitas untuk peningkatan prestasi siswa
- h. Mengusulkan pemberian beasiswa bagi siswa yang berprestasi melalui beasiswa untuk meotivasi minat belajar siswa
- i. Mengusulkan pemberian beasiswa bagi siswa ekonomi lemah yang berprestasi melalui Diknas Provinsi untuk membantu beban biaya pendidikan
- j. Mengkonsultasikan pelaksanaan tugas dengan atasan baik lisan maupun tertulis untuk beroleh petunjuk lebih lanjut

- k. Mengkoordinasikan pelaksanaan tugas dengan Sekretaris dan Kepala-Kepala Bidang melalui rapat/pertemuan untuk penyatuan pendapat.
- l. Mendistribusikan tugas kepada bawahan sesuai bidang untuk tertibnya pelaksanaan tugas
- m. Menyusun laporan pelaksanaan tugas secara berkala sebagai bahan evaluasi
- n. Melaksanakan tugas lain yang diperintahkan oleh atasan untuk kelancaran tugas kedinasan.

#### **4. BIDANG PENDIDIKAN SEKOLAH MENENGAH**

Melaksanakan tugas dibidang pengembangan Pendidikan Menengah berdasarkan peraturan perundang-undangan untuk peningkatan mutu pendidikan.

Dalam melaksanakan tugas Bidang Pendidikan Pendidikan Sekolah Menengah menyelenggarakan fungsi :

- a. Menghimpun kebijakan teknis dibidang pengembangan Pendidikan Menengah sesuai kebutuhan .
- b. Menghimpun data pendidikan menengah sesuai jenis untuk mengetahui jumlah dan perkembangannya.
- c. Menyusun rencana kegiatan dibidang Pendidikan Menengah berdasarkan Petunjuk Pelaksanaan dan Petunjuk Teknis untuk menjadi Program Unit.
- d. Mendistribusikan pedoman pelaksanaan proses belajar mengajar melalui sekolah-sekolah sebagai bahan acuan guru dalam peningkatan mutu pendidikan .

- e. Meningkatkan kompetensi siswa melalui lomba kreatifitas untuk peningkatan prestasi siswa.
- f. Mengusulkan pemberian penghargaan bagi siswa yang berprestasi melalui program beasiswa untuk memotivasi minat belajar siswa.
- g. Mengusulkan Pemberian beasiswa bagi siswa ekonomi lemah yang berprestasi melalui Diknas Provinsi untuk memotivasi minat belajar siswa.
- h. Mengkonsultasikan pelaksanaan tugas dengan atasan baik lisan maupun tertulis untuk beroleh petunjuk lebih lanjut.
- i. Mengkoordinasikan pelaksanaan tugas dengan Sekertaris dan kepala-kepala Bidang melalui rapat/pertemuan untuk penyatuan pendapat.
- j. Mendistribusikan tugas kepada bawaan sesuai bidang untuk tertibnya pelaksanaan tugas.
- k. Menyusun laporan pelaksanaan tugas secara berkala sebagai bahan evaluasi.
- l. Melaksanakan tugas lain yang diperintahkan oleh atasan untuk kelancaran tugas kedinasan.

## **5. BIDANG PENINGKATAN MUTU PENDIDIK DAN TENAGA KEPENDIDIKAN (PMPTK)**

Bidang PMPTK, mempunyai tugas :

Melaksanakan tugas dibidang pengembangan mutu pendidik dan tenaga kendidikan berdasarkan peraturan perundang-undangan untuk kelancaran pelaksanaaan tugas.

Dalam melaksanakan tugas dibidang PMPTK menyelenggarakan fungsi :

- a. Menghimpun kebijakan teknis dibidang pengembangan peningkatan mutu pendidik dan tenaga kependidikan
- b. Menyiapkan bahan perumusan kebijakan dibidang PMPTK meliputi pembinaan profesi dan sertifikasi tenaga pendidik dan kependidikan serta pengembangan kurikulum
- c. Menghimpun , mengelola data dan pemetaan pengembangan mutu pendidik dan tenaga kependidikan formal untuk mengetahui perkembangan kompetensi pendidik dan tenaga kependidikan. Serta pengembangan kurikulum.
- d. Menyusun rencana kegiatan dibidang PMPTK berdasarkan petunjuk pelaksanaan dan petunjuk teknis untuk menjadi program unit .
- e. Memberikan bimbingan teknis, supervise dan evaluasi dibidang pembinaan profesi serta sertifikasi pendidik .
- f. Memberikan bimbingan teknis, supervisi dan evaluasi bagi tenaga kependidikan
- g. Mengkoordinasikan dan melaksanakan pengembangan kurikulum pendidikan dasar dan menengah dengan bidang TK, SD, dan SM
- h. Mengkonsultasikan tugas dengan atasan secara lisan maupun tertulis untuk beroleh petunjuk lebih lanjut.
- i. Mengkoordinasikan pelaksanaan tugas dengan sekretaris dan kepala- kepala bidang melalui rapat/pertemuan untuk penyatuan pendapat
- j. Mendistribusikan tugas kepada bawahan sesuai bidang untuk tertibnya pelaksanaan tugas

- k. Menyusun laporan pelaksanaan tugas secara berkala sebagai bahan evaluasi
- l. Melaksanakan tugas lain yang diperintahkan oleh atasan untuk kelancaran tugas kedinasan.

## **6. BIDANG PENDIDIKAN NON FORMAL DAN INFORMAL**

Melaksanakan tugas dibidang pengembangan Pendidikan Informal dan Non Formal berdasarkan peraturan perundang-undangan untuk peningkatan mutu pendidikan.

Dalam melaksanakan tugas Bidang Pendidikan Informal dan Non Formal menyelenggarakan fungsi:

- a. Menghimpun kebijakan teknis dibidang pengembangan Pendidikan Informal dan Non Formal sesuai kebutuhan sebagai pedoman pelaksanaan tugas.
- b. Menghimpun data siswa Pendidikan Informal dan Non Formal melalui format untuk mengetahui jumlah dan perkembangannya.
- c. Menyusun rencana kegiatan dibidang pengembangan Pendidikan Informal dan Non Formal berdasarkan Petunjuk Pelaksanaan dan Petunjuk Teknis untuk menjadi program unit.
- d. Melakukan koordinasi penyelenggaraan Pendidikan Informal dan Non Formal secara lintas sektoral untuk pembinaan kegiatan belajar mengajar.
- e. Melakukan koordinasi pelaksanaan penuntasan Wajib Belajar 9 tahun dan pemberantasan Buta Aksara pada Pendidikan Informal dan Non Formal secara terpadu untuk peningkatan mutu pendidikan.

- f. Melaksanakan monitoring dan evaluasi terhadap penyelenggaraan program Pendidikan Informal dan Non Formal secara rutin untuk mengetahui kegiatannya.
- g. Mengkoordinasikan pelaksanaan peningkatan mutu pendidik dan tenaga kePendidikan Informal dan Non Formal secara lintas sektoral untuk mengetahui kegiatannya.
- h. Melakukan akreditas lembaga-lembaga Pendidikan Informal dan Non Formal sesuai ketentuan untuk kejelasan status lembaga pendidikan.
- i. Mengkoordinasikan penerbitan izin operasional penyelenggaraan Pendidikan Informal dan Non Formal secara sektoral untuk legalitas lembaga pendidikan
- j. Mengkonsultasikan pelaksanaan tugas dengan atasan baik lisan maupun tertulis untuk beroleh petunjuk lebih lanjut.
- k. Mengkoodinasikan pelaksanaan tugas dengan sekretaris dan kepala-kepala bidang melalui rapat/pertemuan untuk kesatuan pendapat.
- l. Mendistribusikan tugas kepada bawahan sesuai bidang untuk tertibnya pelaksanaan tugas.
- m. Menyusun laporan pelaksanaan tugas secara berkala sebagai bahan evaluasi.
- n. Melaksanakan tugas lain yang diperintahkan oleh atasan untuk kelancaran tugas kedinasan.

Untuk memudahkan pelayanan, koordinasi, dan kelancaran pelaksanaan tugas, DINAS PENDIDIKAN Kota Gorontalo dibantu 9 UPTD Cabang Dinas

Pendidikan Kecamatan dan 1 UPTD PNFI, yaitu :

1. UPTD Cabang Dinas Pendidikan Kecamatan Kota Utara
2. UPTD Cabang Dinas Pendidikan Kecamatan Kota Timur
3. UPTD Cabang Dinas Pendidikan Kecamatan Kota Selatan
4. UPTD Cabang Dinas Pendidikan Kecamatan Kota Barat
5. UPTD Cabang Dinas Pendidikan Kecamatan Dungingi
6. UPTD Cabang Dinas Pendidikan Kecamatan Kota Tengah
7. UPTD Cabang Dinas Pendidikan Kecamatan Sipatana
8. UPTD Cabang Dinas Pendidikan Kecamatan Dumbo Raya
9. UPTD Cabang Dinas Pendidikan Kecamatan Hulontalangi
10. UPTD Sanggar Kegiatan Belajar (SKB) Kota Gorontalo

### **5.1.2 Hasil Pengujian Sistem**

#### **5.1.2.1 Pengujian White Box**

1. Proses Pencarian Lokasi

```
var $map = array();
var $infiniteDistance = 0;
var $numberOfNodes = 0;
var $bestPath = 0;
var $matrixWidth = 0;
```

```
function findShortestPath($start, $to) {
    $this -> startnode = $start;
    for ($i=0; $i<$this -> numberOfNodes; $i++) {
        if ($i == $this -> startnode) {
            $this -> visited[$i] = true;
            $this -> distance[$i] = 0;
        } else {
            $this -> visited[$i] = false;
            $this -> distance[$i] = isset($this -> map[$this -> startnode][$i])
                ? $this -> map[$this -> startnode][$i]
                : $infiniteDistance;
        }
    }
}
```

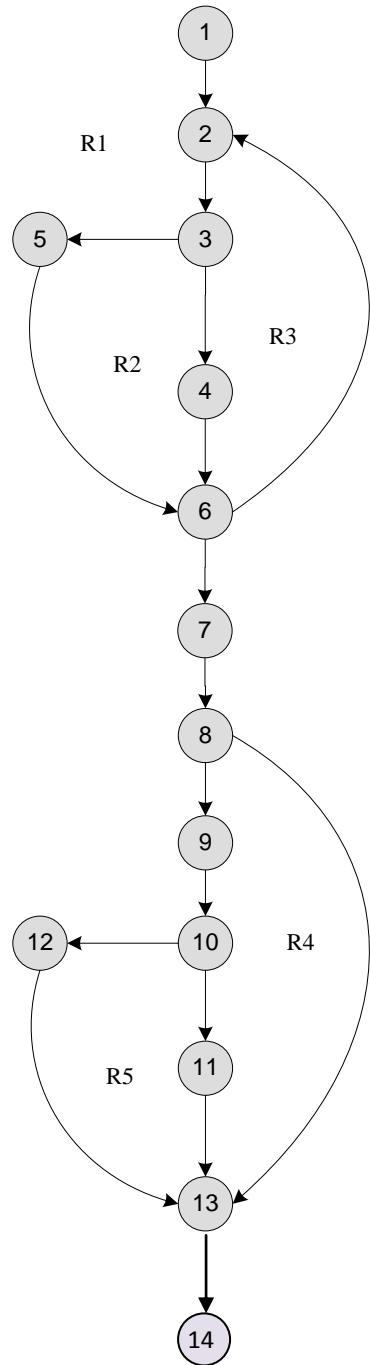
```

        : $this ->infiniteDistance;
    }
$this ->previousNode[$i] = $this ->startnode; ← (6)
}

(7) { $maxTries = $this ->numberOfNodes;
       $tries = 0;
while (in_array(false,$this ->visited,true) && $tries <= $maxTries) { ← (8)
    $this ->bestPath = $this->findBestPath($this->distance,array_keys($this ->visited,false,true));
    if($to !== null && $this ->bestPath === $to) { ← (10)
        break; ← (11)
    }
(12) { $this ->updateDistanceAndPrevious($this ->bestPath);
        $this ->visited[$this ->bestPath] = true;
        $tries++; ← (13)
    }
} ← (14)

```

## 2. Flowgraph Proses Pencarian Lokasi



**Gambar 5.2** Flowgraph Proses Pencarian Lokasi

a. Menghitung Nilai *Cyclomatic Complexity* (CC)

Dimana :

$$\text{Region}(R) = 5$$

$$\text{Node}(N) = 14$$

$$\text{Edge}(E) = 17$$

$$\text{Predicate Node}(P) = 4$$

$$V(G) = E - N + 2$$

$$= 17 - 14 + 2$$

$$= 5$$

$$V(G) = P + 1$$

$$= 4 + 1$$

$$= 5$$

Jadi *cyclomatic complexity* untuk *flowgraph* proses pencarian lokasi lembaga kursus adalah 5. Berdasarkan tabel hubungan antara *cyclomatic complexity* dan Resiko menurut Mc Cabe, menunjukkan bahwa nilai CC 5 masuk dalam *type of procedure a well structured and stable procedure* (strukturnya baik dan prosedur stabil) serta resikonya *Low* (rendah)

b. Menentukan Basis Path :

Basis set yang dihasilkan dari jalur independent path secara linier adalah jalur sebagai berikut :

Jalur 1 : 1-2-3-4-6-7-8-9-10-11-13

Jalur 2 : 1-2-3-5-6-7-8-10-11-13

Jalur 3 : 1-2-3-4-6-2-...

Jalur 4 : 1-2-3-4-6-7-8-13

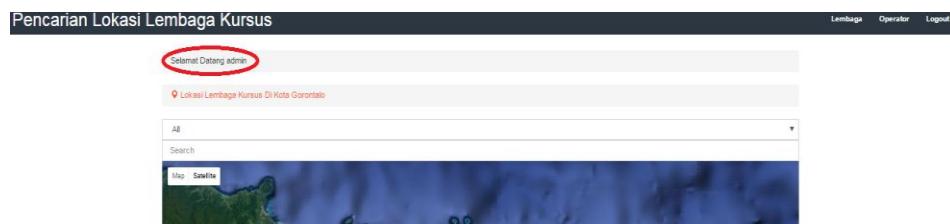
Jalur 5 : 1-2-3-4-6-7-8-9-10-12-13

Ketika aplikasi dijalankan, maka terlihat bahwa semua basis path yang dihasilkan telah dieksekusi satu kali. Berdasarkan ketentuan tersebut dari segi kelayakan *software*, system ini telah memenuhi syarat.

### 5.1.2.2 Pengujian *Black Box*

#### 1. Form Login

**Gambar 5.3** Tampilan Form Login



**Gambar 5.4** Tampilan Home

**Tabel 5.1** Tabel Pengujian *Black Box* Form Login

Input/Event	Fungsi	Hasil	Hasil Uji
Masukkan username salah, klik Login	Menguji validasi username	Tidak dapat login menu utama	Sesuai
Masukkan password salah	Menguji validasi password	Tidak dapat login menu utama	Sesuai
Masukkan nama user dan Password yang benar	Menguji validasi proses login	Tampilan Menu Utama dan muncul "Selamat datang (nama user)"	Sesuai

## 2. Menu Lembaga

Daftar Lembaga

Data Lembaga				+ Tambah Lembaga
Lembaga	No Telepon	Ketua	Action	
Kids 2 Success		Ventriwati	 	
MY LIBERTY International English School	085242999943 / 0435 - 8525515	Muhammad Busyaeri Jafar, SH.	 	
C55	085256908982	Irwan Karim	 	

**Gambar 5.5** Tampilan Daftar Lembaga Kursus

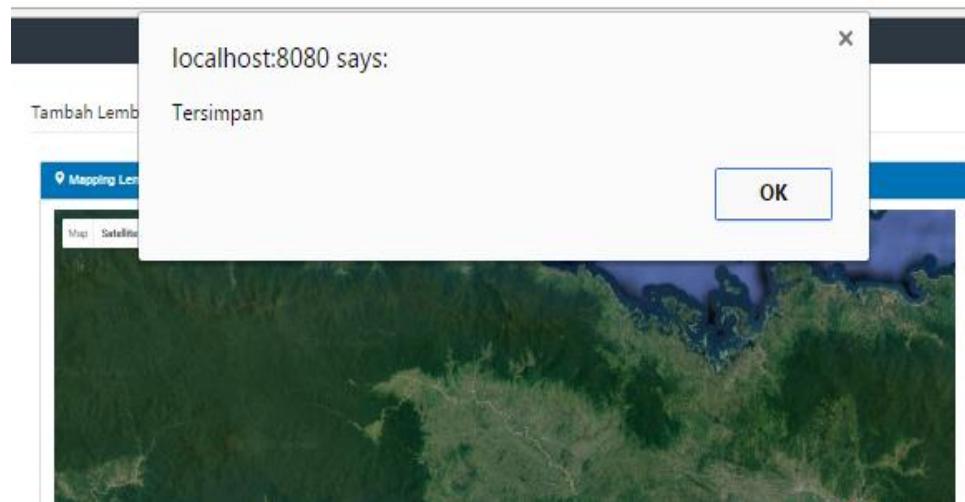
Tambah Lembaga

Map Satellite



Latitude: 0.54949  
Longitude: 123.06098  
Nama: LPK Kharisma  
Ketua: Dougg Henry Woodward, SE

**Gambar 5.6** Form Input Data Lembaga Kursus



**Gambar 5.7** Tampilan Pesan Tersimpan Data Lembaga Kursus

**Tabel 5.2** Tabel Pengujian *Black Box* Menu Lembaga

Input/Event	Fungsi	Hasil	Hasil Uji
Klik menu Lembaga	Menampilkan data lembaga kursus	Tampil Daftar Lembaga Kursus	Sesuai
Klik Tambah Lembaga	Menambahkan data lembaga	Tampil form Input Data Lembaga Kursus	Sesuai
Input data lembaga kursus, klik tombol Simpan	Menguji proses penyimpanan data lembaga kursus	Tampil pesan “Tersimpan”	Sesuai

### 3. Menu Operator

A screenshot of a web application showing a table titled 'Daftar Operator'. The table has columns for 'Nama Operator', 'Username', 'Lembaga', and 'Action'. There are three rows of data:

Nama Operator	Username	Lembaga	Action
Ringgo	Ringgo	MY LIBERTY International English School	
Ventri	Ventri	Kids 2 Success	
andi	andi	C55	

**Gambar 5.8** Tampilan Daftar Operator

Tambah Operator

Isi Data

Nama Operator	Rendi
Username	Rendi
Password	.....
Role	Lembaga
Lembaga	LPK Kharisma
<b>Simpan</b>	

**Gambar 5.9** Form Input Data Operator

**Tabel 5.3** Tabel Pengujian *Black Box* Menu Operator

Input/Event	Fungsi	Hasil	Hasil Uji
Klik menu Operator	Menampilkan data operator	Tampil Data Operator	Sesuai
Klik Tambah	Menambahkan data operator	Tampil form Input Data Operator	Sesuai

#### 4. Menu Kategori

Daftar Kategori

Data Kategori

Kategori	Action
Komputer	
Bahasa Inggris	

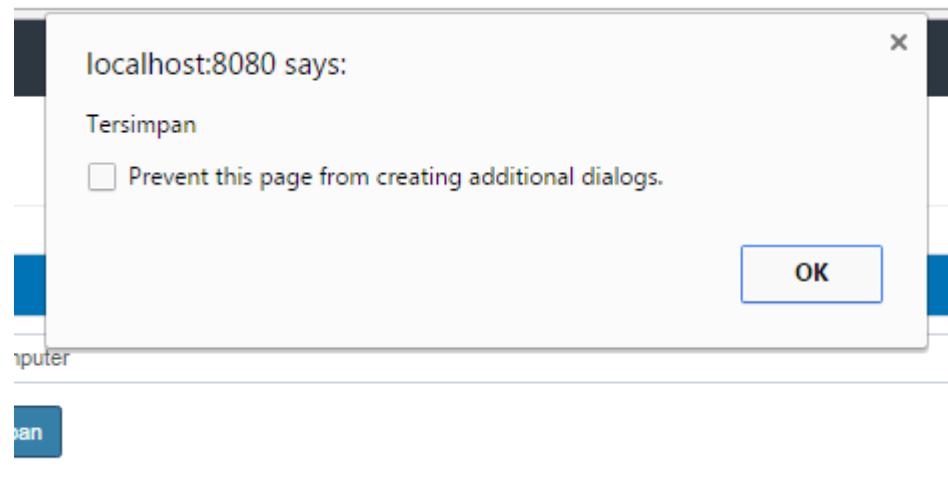
**Gambar 5.10** Tampilan Daftar Kategori

Tambah Kategori

Isi Data Kategori

Kategori	Komputer
<b>Simpan</b>	

**Gambar 5.11** Form Tambah Kategori



**Gambar 5.12** Tampilan Pesan Tersimpan Data Kategori

**Tabel 5.4** Tabel Pengujian *Black Box* Menu Kategori

Input/Event	Fungsi	Hasil	Hasil Uji
Klik menu Kategori	Menampilkan daftar kategori	Tampil Daftar Kategori	Sesuai
Klik Tambah	Menambahkan kategori	Tampil form tambah Kategori	Sesuai
Input kategori, klik tombol Simpan	Menguji proses penyimpanan kategori	Tampil pesan “Tersimpan”	Sesuai

## 5. Menu Pengajar

Data Pengajar			+ Tambah
Nama	Jabatan	Action	
Delyan	Instruktur Bahasa Inggris		
Romy	Instruktur Komputer		
Delviana	Instruktur Komputer		
Selvi	Instruktur Bahasa Inggris		

**Gambar 5.13** Tampilan Daftar Pengajar

Tambah Pengajar

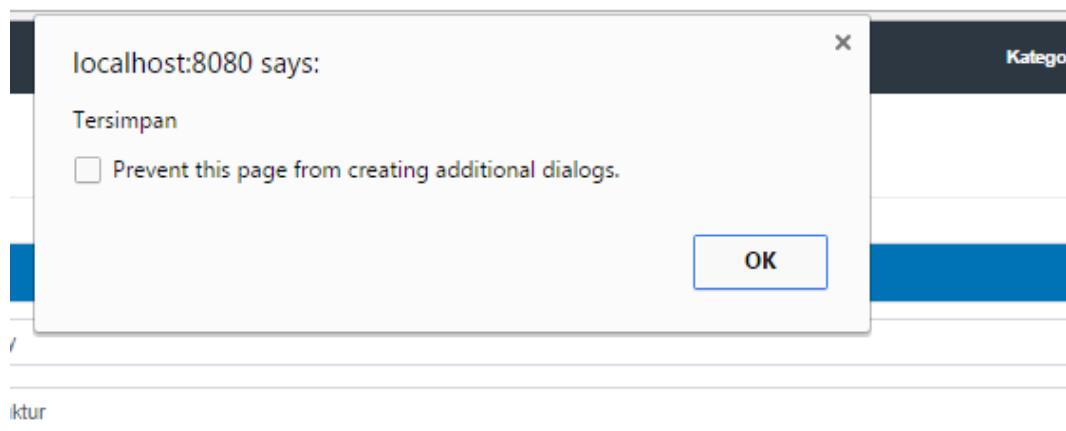
Isi Data Pengajar

Nama: Romy

Jabatan: Instruktur

**Simpan**

**Gambar 5.14** Form Tambah Pengajar



**Gambar 5.15** Tampilan Pesan Tersimpan Data Pengajar

**Tabel 5.5** Tabel Pengujian *Black Box* Menu Pengajar

Input/Event	Fungsi	Hasil	Hasil Uji
Klik menu Pengajar	Menampilkan daftar pengajar	Tampil Daftar Pengajar	Sesuai
Klik Tambah	Menambahkan pengajar	Tampil form tambah Pengajar	Sesuai
Input data pengajar, klik tombol Simpan	Menguji proses penyimpanan data pengajar	Tampil pesan "Tersimpan"	Sesuai

## 6. Menu Fasilitas

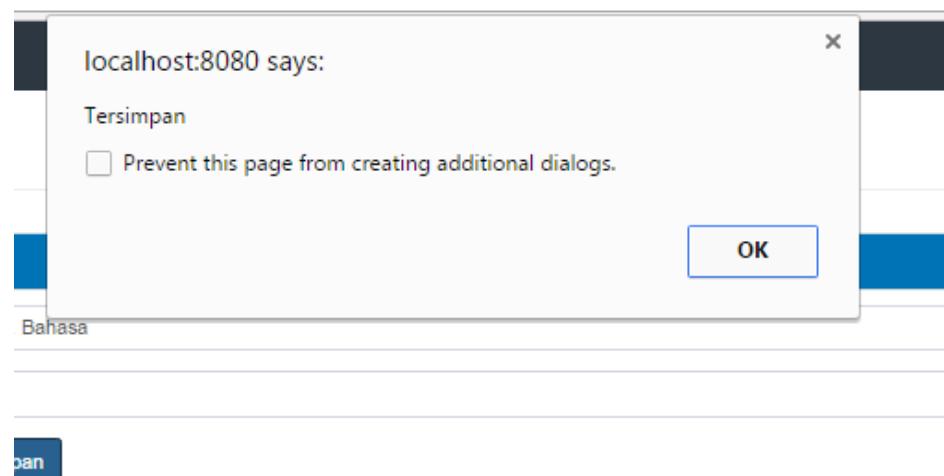
Daftar Fasilitas

Data Fasilitas			<a href="#">Tambah</a>
Fasilitas	Jumlah	Action	
Internet	100 Mbps	<a href="#"></a> <a href="#"></a>	
Lab. Komputer	3	<a href="#"></a> <a href="#"></a>	
Lab. Bahasa	3	<a href="#"></a> <a href="#"></a>	
LCD	6	<a href="#"></a> <a href="#"></a>	

**Gambar 5.16** Tampilan Daftar Fasilitas

Tambah Fasilitas

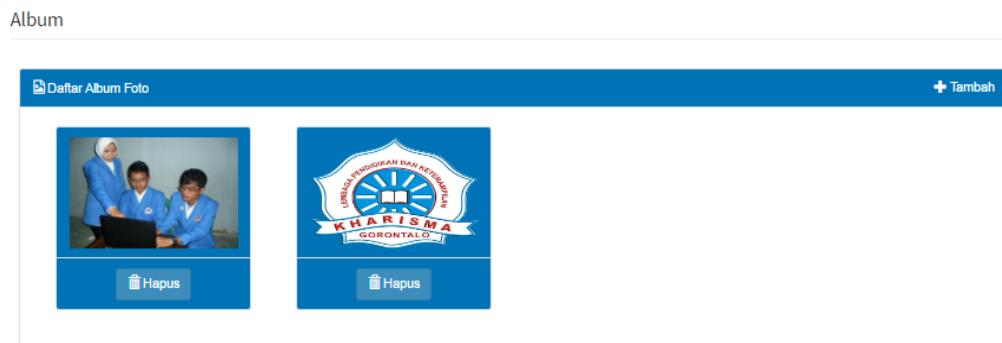
Isi Data Fasilitas		<a href="#">Kembali</a>
Fasilitas	<input type="text" value="Lab. Bahasa"/>	
Jumlah	<input type="text" value="3"/>	
<a href="#">Simpan</a>		

**Gambar 5.17** Form Tambah Fasilitas**Gambar 5.18** Tampilan Pesan Tersimpan Data Fasilitas

**Tabel 5.6** Tabel Pengujian *Black Box* Menu Fasilitas

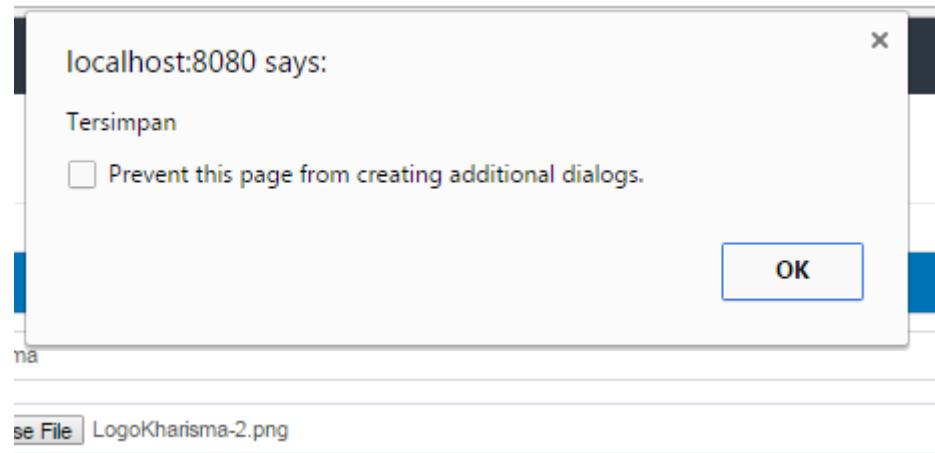
Input/Event	Fungsi	Hasil	Hasil Uji
Klik menu Fasilitas	Menampilkan data fasilitas	Tampil Data Fasilitas	Sesuai
Klik Tambah	Menambahkan data fasilitas	Tampil form Input Fasilitas	Sesuai
Input fasilitas, klik tombol Simpan	Menguji proses penyimpanan fasilitas	Tampil pesan “Tersimpan”	Sesuai

## 7. Menu Album

**Gambar 5.19** Tampilan Daftar Album Foto

The screenshot shows a 'Tambah Album' (Add Album) form. At the top, there's a blue header bar with a camera icon and the text 'Upload Foto' on the left, and a back arrow and 'Kembali' (Back) on the right. Below the header, the form has two main input fields. The first field is labeled 'Nama Foto' (Name Photo) with the value 'kharisma' entered. The second field is labeled 'File' with a 'Choose File' button, showing the file path 'LogoKharisma-2.png'. At the bottom right of the form is a blue 'Upload' button. The entire form is contained within a light gray box.

**Gambar 5.20** Form Upload Foto



**Gambar 5.21** Tampilan Pesan Foto Tersimpan

**Tabel 5.7** Tabel Pengujian *Black Box* Menu Album

Input/Event	Fungsi	Hasil	Hasil Uji
Klik menu Album	Menampilkan Album	Tampil album	Sesuai
Klik Tambah	Meng-upload album	Tampil form upload album	Sesuai
<i>Upload</i> album, klik tombol Simpan	Menguji proses <i>upload</i> album	Tampil pesan "Tersimpan"	Sesuai

## 8. Menu Brosur



**Gambar 5.22** Tampilan Foto Brosur

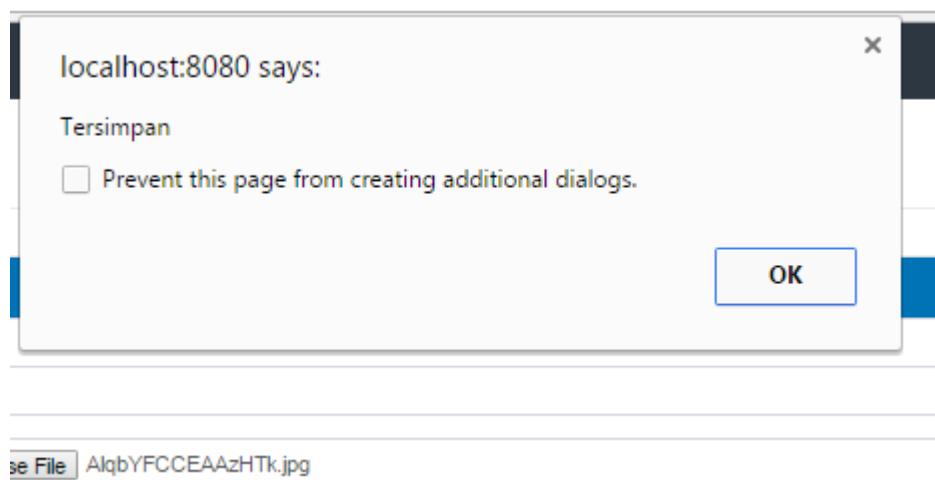
Brosur

Foto Brosur

Nama Brosur: brosur

File: Choose File AlqbYFCCEAAzHTk.jpg

Upload

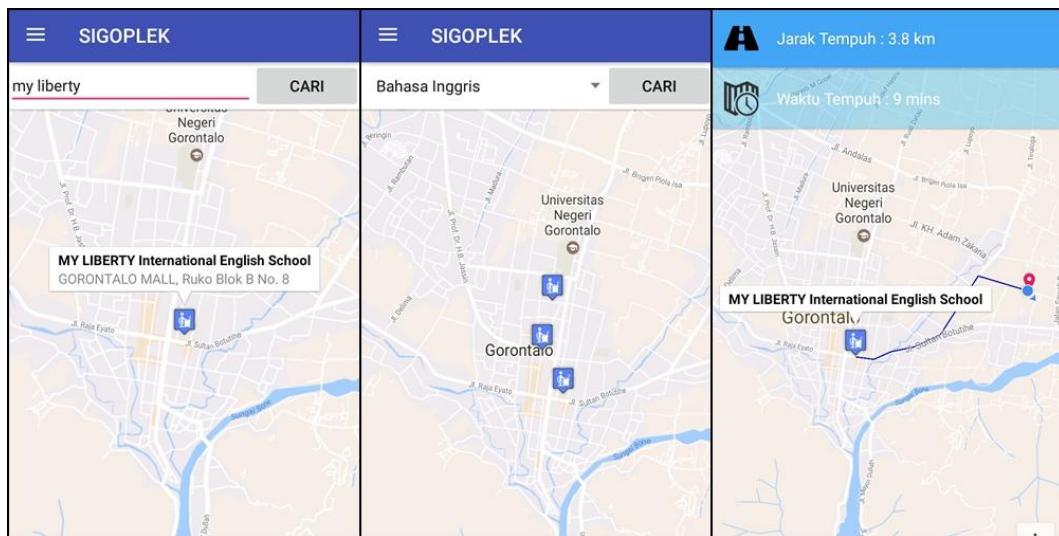
**Gambar 5.23** Form Upload Brosur

Choose File AlqbYFCCEAAzHTk.jpg

**Gambar 5.24** Tampilan Pesan Brosur Tersimpan**Tabel 5.8** Tabel Pengujian *Black Box* Menu Brosur

Input/Event	Fungsi	Hasil	Hasil Uji
Klik menu Brosur	Menampilkan Brosur	Tampil Brosur	Sesuai
Klik Tambah	Meng-upload brosur	Tampil form unggah brosur	Sesuai
Unggah brosur, klik tombol Simpan	Menguji proses unggah brosur	Tampil pesan "Tersimpan"	Sesuai

## 9. Pencarian Lokasi Lembaga Kursus



**Gambar 5.25** Pencarian Lokasi Lembaga Kursus

**Tabel 5.9** Tabel Pengujian *Black Box* Pencarian Lokasi Lembaga Kursus

Input/Event	Fungsi	Hasil	Hasil Uji
Klik kotak pencarian	Mencari alamat lembaga kursus	Tampil lembaga kursus yang akan di tuju	Sesuai
Input nama lembaga kursus atau kategori, klik Cari	Melakukan proses pencarian lokasi lembaga kursus	Info Alamat dan Profil Lembaga Kursus	Sesuai
Klik lembaga kursus yang muncul	Menampilkan hasil pencarian rute google	Tampil rute, jarak dan waktu tempuh menuju lembaga kursus	Sesuai

Ketika aplikasi dijalankan, maka terlihat bahwa semua pengujian *black box* yang dihasilkan telah dieksekusi satu kali. Berdasarkan ketentuan tersebut dari segi kelayakan aplikasi, sistem ini telah memenuhi syarat.

## 5.2 Pembahasan

### 5.2.1 Deskripsi Kebutuhan Hardware/Software

Penulis dalam mengembangkan Website ini menggunakan bahasa pemrograman *Java* dan Basis Data *MySQL*.

Pada dasarnya, untuk implementasi sistem ini membutuhkan beberapa konfigurasi dasar, diantaranya :

#### 1. *Hardware* dan *Software*

##### A. Spesifikasi yang disarankan untuk komputer (*server*)

- a. Processor setara Core 2 Duo atau lebih.
- b. RAM (Memory) 1 GB atau lebih.
- c. HDD 100 GB atau lebih.
- d. Monitor SVGA dengan Resolusi 1024 X 768
- e. Dan Peralatan I/O Lainnya.
- f. Windows XP, Vista atau Windows 7.
- g. Browser Mozilla Firefox, Google Chrome dan Opera untuk membuka Web.

##### B. Spesifikasi yang disarankan untuk *smartphone* (*client*)

- a. Processor setara Dual Core 1 Ghz atau lebih.
- b. RAM (Memory) 512 MB atau lebih.
- c. OS Android versi 4.0 (Ice Cream Sandwich) atau lebih untuk akses mobile
- d. Global Positioning System (GPS).

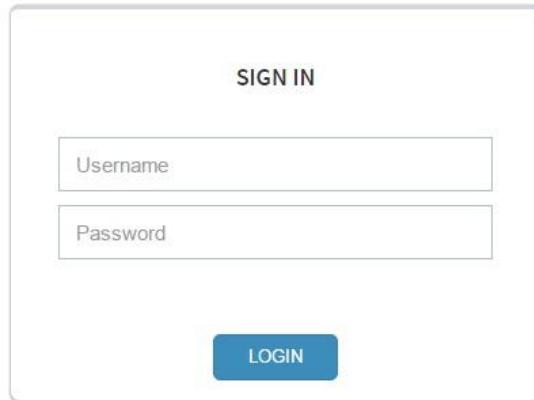
## 2. Brainware

Yaitu sumber daya manusia yang terlibat di dalam mengoperasikan serta mengatur sistem komputer. Sumber daya yang dibutuhkan dengan karakteristik sebagai berikut memiliki kemampuan dasar tentang komputer dan proses yang berlangsung di dalamnya.

### 5.2.2 Langkah – langkah Menjalankan Sistem

Untuk admin (Bidang PNFI) dan operator (Lembaga Kursus), cara menjalankan sistem adalah dengan mengetikkan alamat pada tab address kemudian dilanjutkan dengan melakukan proses *login*.

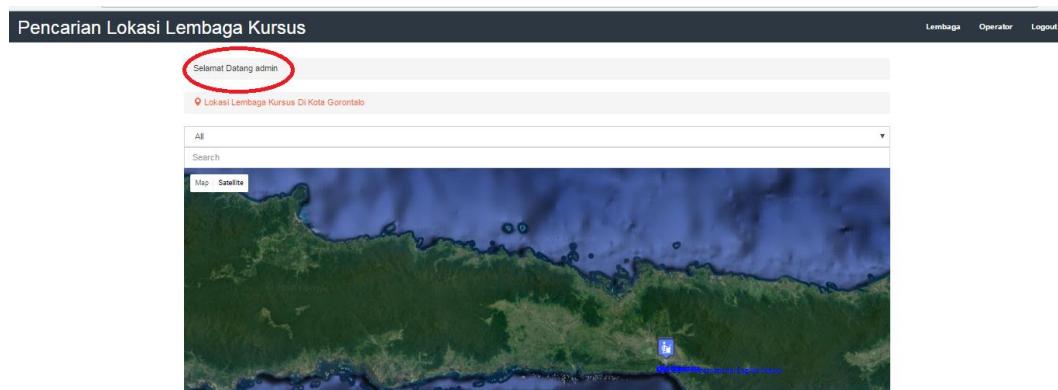
#### 5.2.2.1 Tampilan Halaman Login



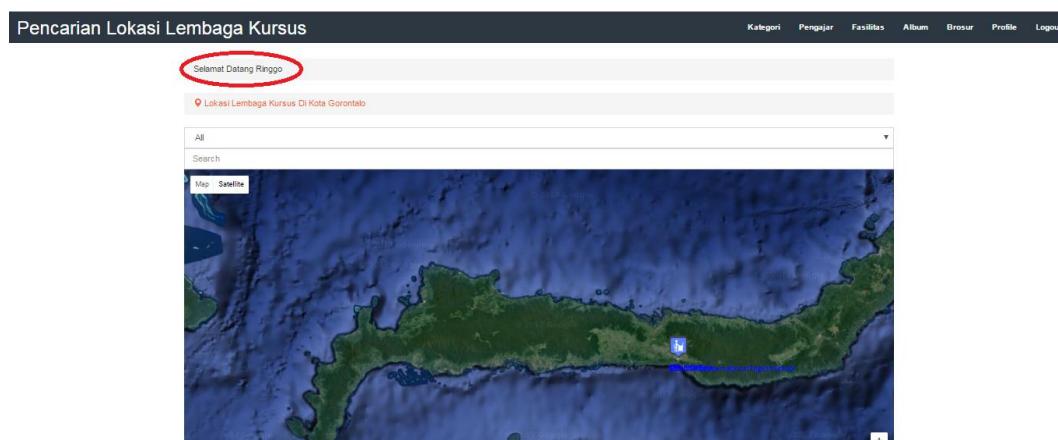
**Gambar 5.26** Form Login

Pada tampilan halaman *login* ini, admin (Bid. PNFI) dan operator (Lembaga Kursus) wajib mengisi *username* dan *password* lalu klik Login untuk masuk ke halaman menu utama. Apabila salah memasukkan *username* atau *password*, maka tidak akan bisa mengakses halaman utama.

### 5.2.2.2 Tampilan Home



**Gambar 5.27** Tampilan Home Admin (Bid. PNFI)



**Gambar 5.28** Tampilan Home Operator (Lembaga Kursus)

Halaman ini merupakan tampilan awal setelah proses *login*. Sekilas terlihat sama, yang membedakan antara tampilan *home* dari admin dan operator terletak pada menu yang tersedia.

Pada tampilan *home* admin hanya terdapat menu Lembaga, Operator dan Logout. Sedangkan pada tampilan *home* operator terdapat menu Kategori,

Pengajar, Fasilitas, Album, Brosur, Profil dan Logout. Masing-masing menu tersebut memiliki fungsi yang berbeda.

### 5.2.2.3 Tampilan Daftar Lembaga Kursus

Daftar Lembaga

Data Lembaga				+ Tambah Lembaga
Lembaga	No Telepon	Ketua	Action	
MY LIBERTY International English School	085242999943 / 0435 - 8525515	Muhammad Busyaeri Jafar, SH.	 	
C55	085256908982	Irwan Karim	 	

**Gambar 5.29** Tampilan Daftar Lembaga Kursus

Halaman ini menampilkan daftar lembaga kursus yang sudah di *input* ke dalam sistem. Untuk menambahkan data Lembaga Kursus yang baru klik "Tambah Lembaga". Untuk mengubah data pilih aksi *edit* dan untuk menghapus pilih aksi *hapus*.

#### 5.2.2.4 Tampilan Form Tambah Data Lembaga Kursus

The screenshot displays a web application for managing institutional data. At the top, there is a header titled "Tambah Lembaga". Below this is a map interface from "Mapping Lembaga" showing a coastal area with a red location pin. The map includes standard controls for zooming (+/-) and switching between "Map" and "Satellite" views. The main content area is titled "Data Lembaga" and contains the following form fields:

Latitude	0.54949
Longitude	123.06099
Nama	LPK Kharisma
Ketua	Asep Henry Widadap, SE

At the bottom of the form are two buttons: "<< Kembali" (Back) and "Simpan" (Save).

**Gambar 5.30** Tampilan Form Tambah Lembaga Kursus

Halaman ini digunakan untuk menambahkan data lembaga kursus yang baru. Dimulai dengan mengisi *Latitude*, *Longitude* Nama Lembaga dan Ketua. Untuk operasi data, gunakan tombol Simpan. Untuk membatalkan proses gunakan tombol <<Kembali.

### 5.2.2.5 Tampilan Daftar Operator

Daftar Operator			
Nama Operator	Username	Lembaga	Action
Ringgo	Ringgo	MY LIBERTY International English School	
andi	andi	C55	

**Gambar 5.31** Tampilan Daftar Operator

Halaman ini menampilkan daftar operator yang sudah di *input* ke dalam sistem. Untuk menambahkan operator yang baru klik "Tambah". Untuk mengubah data pilih aksi *edit* dan untuk menghapus pilih aksi *hapus*.

### 5.2.2.6 Tampilan Form Tambah Operator

Tambah Operator	
<b>Isi Data</b>	
Nama Operator	<input type="text" value="Ringgo"/>
Username	<input type="text" value="Ringgo"/>
Password	<input type="password" value="....."/>
Role	<input type="text" value="Lembaga"/> <ul style="list-style-type: none"> <li><input type="radio"/> Admin</li> <li><input checked="" type="radio"/> Lembaga</li> </ul>
Lembaga	<input type="button" value="Simpan"/>

**Gambar 5.32** Tampilan Form Tambah Operator

Form ini berfungsi untuk menambahkan pengguna baik dari Bid. PNFI (Admin) maupun Pihak Lembaga Kursus (Operator). Hal yang membedakan keduanya terdapat pada hak akses. Admin dapat melihat keseluruhan daftar operator sedangkan pihak lembaga kursus tidak. Namun, hanya operator dari

lembaga kursus yang bersangkutan yang dapat melakukan *update* data informasi lembaga kursus yang dikelolanya.

Tambah Operator

Isi Data	
Nama Operator	Ringgo
Username	Ringgo
Password	.....
Role	Lembaga
Lembaga	MY LIBERTY International English School

Simpan

**Gambar 5.33** Tampilan Form Tambah Operator

Form ini berfungsi untuk menambahkan pengguna, baik dari Bid. PNFI (Admin) maupun Pihak Lembaga Kursus (Operator). Dimulai dengan mengisi Nama Operator, *Username*, *Password*, *Role* serta Lembaga Kursus. Untuk operasi data, gunakan tombol Simpan. Untuk membatalkan proses gunakan tombol <<Kembali.

Hal yang membedakan antara Admin dan Operator terdapat pada hak akses. Admin dapat melihat keseluruhan daftar operator sedangkan pihak lembaga kursus tidak. Namun, hanya operator dari lembaga kursus yang bersangkutan yang dapat melakukan *update* data informasi lembaga kursus yang dikelolanya.

### 5.2.2.7 Tampilan Form Tambah Kategori

Tambah Kategori

The screenshot shows a form titled 'Tambah Kategori'. At the top left is a checkbox labeled 'Isi Data Kategori'. On the right is a link 'Kembali'. Below the title, there is a label 'Kategori' followed by a dropdown menu containing the following items: Mengemudi, Menjahit, Public Speaking, Bimbingan Belajar, Komputer, and Bahasa Inggris.

**Gambar 5.34** Tampilan Form Tambah Kategori

Tambah Kategori

The screenshot shows the same 'Tambah Kategori' form. The dropdown menu now only contains 'Bahasa Inggris'. Below the dropdown is a blue 'Simpan' button.

**Gambar 5.35** Tampilan Form Tambah Kategori

Form ini berfungsi untuk menambahkan kategori lembaga kursus. Operator akan memilih kategori sesuai dengan lembaga kursus yang dikelolanya. Untuk operasi data, gunakan tombol Simpan. Untuk membatalkan proses gunakan tombol <<Kembali.

### 5.2.2.8 Tampilan Daftar Kategori

Daftar Kategori

The screenshot shows a table titled 'Data Kategori'. The first column is 'Kategori' and the second is 'Action'. It lists two entries: 'Bahasa Inggris' and 'Bimbingan Belajar'. Each entry has a blue edit icon (pencil) and a blue delete icon (trash bin).

**Gambar 5.36** Tampilan Daftar Kategori

Halaman ini menampilkan daftar kategori yang sudah di *input* ke dalam sistem. Untuk menambahkan operator yang baru klik "Tambah". Untuk menghapus pilih aksi hapus.

### 5.2.2.9 Tampilan Form Tambah Pengajar

Isi Data Pengajar	
Nama	Sister Dewi
Jabatan	Instruktur
<b>Simpan</b>	

**Gambar 5.37** Tampilan Form Tambah Data Pengajar

Form ini berfungsi untuk menambahkan data pengajar. Operator akan memasukkan nama pengajar dan jabatan. Untuk menyimpan data, klik tombol Simpan.

### 5.2.2.10 Tampilan Daftar Pengajar

Data Pengajar			+ Tambah
Nama	Jabatan	Action	
Brother Redhy	Instruktur		
Sister Dewi	Instruktur		
Sister Ivon	Instruktur		
Brother David	Instruktur		

**Gambar 5.38** Tampilan Daftar Pengajar

Halaman ini menampilkan daftar pengajar yang sudah di *input* ke dalam sistem. Untuk menambahkan data baru klik "Tambah". Untuk mengubah data pilih aksi *edit* dan untuk menghapus pilih aksi hapus.

#### 5.2.2.11 Tampilan Form Tambah Fasilitas

Fasilitas	Jumlah
Ruang Kelas	6

**Gambar 5.39** Tampilan Form Tambah Fasilitas

Form ini berfungsi untuk menambahkan fasilitas yang disediakan oleh lembaga kursus. Operator akan memasukkan nama fasilitas dan jumlah. Untuk menyimpan data, klik tombol Simpan.

#### 5.2.2.12 Tampilan Daftar Fasilitas

Fasilitas	Jumlah	Action
LCD	4	
Ruang Kelas	6	
Internet	100 Mbps	
Kamar Mandi	2	

**Gambar 5.40** Tampilan Daftar Fasilitas

Halaman ini menampilkan daftar fasilitas yang sudah di *input* ke dalam sistem. Untuk menambahkan data baru klik "Tambah". Untuk mengubah data pilih aksi *edit* dan untuk menghapus pilih aksi *hapus*.

#### **5.2.2.13 Tampilan Form Upload Album (Foto)**

Tambah Album

**Gambar 5.41** Tampilan Form Upload Foto

Form ini digunakan untuk *upload* foto kegiatan di lembaga kursus. Operator akan memasukkan nama foto dan meng-*upload* foto dengan klik "Choose File". Untuk mengunggah foto, klik "Upload". Masing-masing lembaga kursus dapat meng-*upload* lebih dari 10 foto ke dalam album.

#### **5.2.2.14 Tampilan Album**

Album



**Gambar 5.42** Tampilan Album

Halaman ini menampilkan foto yang sudah di *upload* ke dalam album. Untuk menambahkan foto lainnya klik "Tambah" dan untuk menghapus klik Hapus.

#### 5.2.2.15 Tampilan Form Upload Brosur

The screenshot shows a web-based form titled "Foto Brosur". At the top left is a small icon of a camera. The form has two input fields: one for "Nama Brosur" containing the value "brosur1" and another for "File" with a "Choose File" button and a message "No file chosen". Below these fields is a blue rectangular button labeled "Upload". The entire form is set against a white background with a thin gray border.

**Gambar 5.43** Tampilan Form Upload Brosur

Form ini berfungsi untuk *upload* brosur lembaga kursus. Operator akan memasukkan nama brosur dan meng-*upload* brosur dengan klik "Choose File" lalu klik "Upload". Berbeda dengan album, di menu ini masing-masing lembaga kursus hanya bisa meng-*upload* brosur saja. Jika lembaga kursus akan mengganti brosur, maka operator diharuskan untuk menghapus brosur sebelumnya kemudian diganti dengan brosur yang baru.

### 5.2.2.16 Tampilan Brosur

Brosur



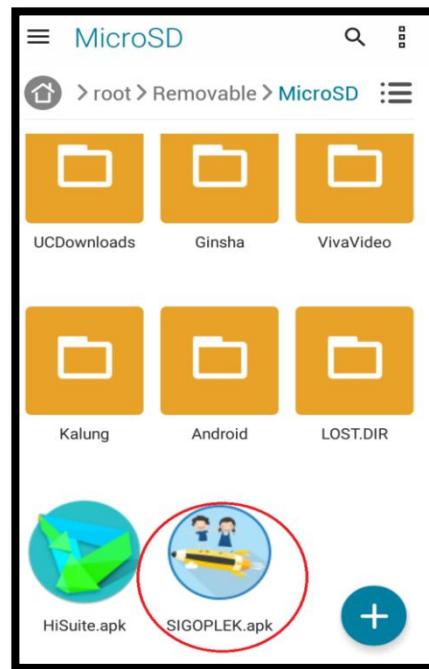
**Gambar 5.44** Tampilan Brosur

Halaman ini menampilkan brosur yang sudah di *upload*. Klik hapus untuk menghapus brosur yang ada.

### 5.2.2.17 Tampilan Aplikasi SIGOPLEK

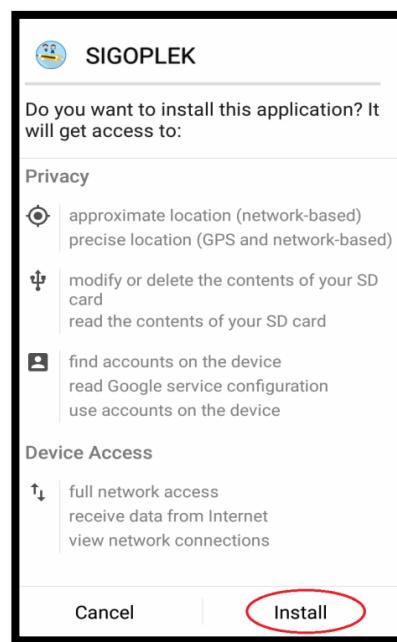
Untuk melakukan pencarian lokasi lembaga kursus pertama-tama masyarakat sebagai pengguna harus meng-*install* aplikasi 'SIGOPLEK.apk' (Sistem Informasi Geografis Pencarian Lokasi Lembaga Kursus) ke dalam *smartphone* terlebih dahulu. Karena aplikasi ini belum dibagikan secara umum, maka masyarakat bisa mendapatkannya dengan menghubungi pihak bidang PNFI, Dinas Pendidikan Kota Gorontalo. Langkah-langkah instalasi-nya adalah sebagai berikut:

1. Klik 'SIGOPLEK.apk' yang tersimpan di File Manager.



**Gambar 5.45** File SIGOPLEK.apk

2. Setelah klik 'SIGOPLEK.apk' maka akan tampilan seperti berikut, lanjutkan dengan klik Next.



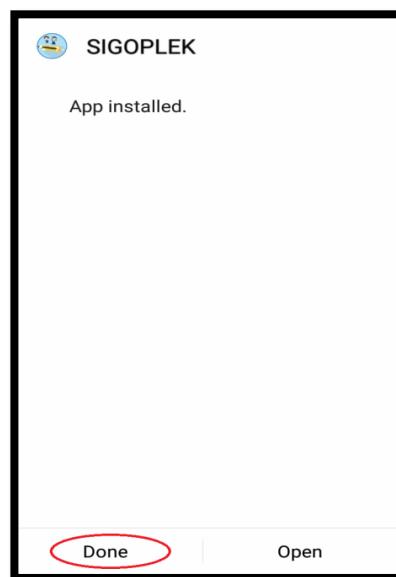
**Gambar 5.46** Install SIGOPLEK

3. Tunggu proses instalasi sampai selesai.



**Gambar 5.47** Proses Install SIGOPLEK

4. Setelah proses instalasi selesai, klik 'Done'. Sampai di tahap ini, aplikasi SIGOPLEK sudah ter-*install* ke dalam *smartphone* dan sudah bisa digunakan.



**Gambar 5.48** Proses Install SIGOPLEK selesai

Setelah aplikasi SIGOPLEK ter-*install*, barulah dilakukan pencarian lokasi lembaga kursus. Berikut ini tampilan awal aplikasi SIGOPLEK.

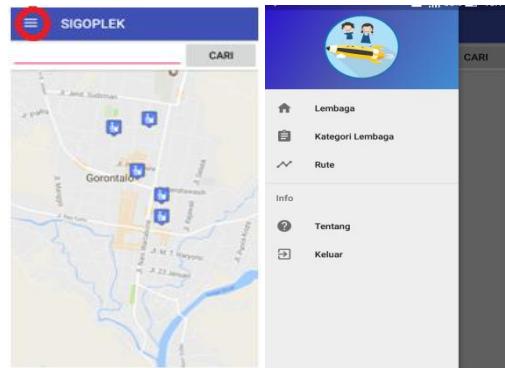


**Gambar 5.49** Tampilan Awal SIGOPLEK

Di tampilan awal ini, aplikasi SIGOPLEK menampilkan seluruh lembaga kursus yang sudah ter-*input* ke dalam sistem. Untuk dapat menggunakan aplikasi ini, masyarakat tidak perlu melakukan proses *login*. Begitu aplikasinya ter-*install* aplikasi bisa langsung digunakan.

#### 5.2.2.18 Tampilan Profil dan Lokasi Lembaga Kursus

Pencarian lokasi lembaga kursus dapat dilakukan dengan memasukkan nama lembaga kursus atau kategori lembaga kursus. Untuk melakukan pencarian lembaga kursus, klik menu.

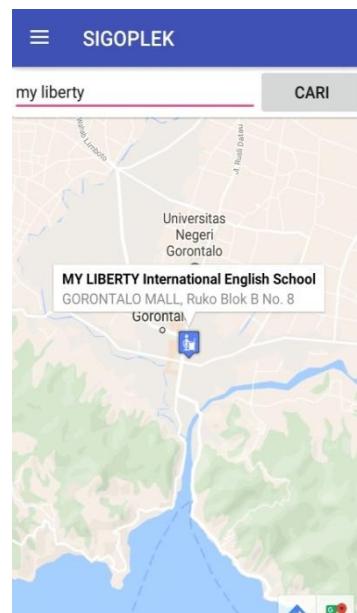


**Gambar 5.50** Tampilan Menu SIGOPLEK

Untuk mengetahui fungsi dari masing – masing menu tersebut, berikut penjelasannya:

### 5. Lembaga

Menu ini berfungsi untuk menampilkan lokasi lembaga kursus berdasarkan nama lembaga kursus. Gambar berikut menampilkan pencarian lokasi lembaga kursus berdasarkan nama lembaga:



**Gambar 5.51** Pencarian Lokasi Lembaga Kursus Berdasarkan Nama

Setelah nama lembaga kursus yang dicari ditemukan, klik pada ikon biru maka akan tampil profil lengkap mengenai lembaga kursus tersebut.



**Gambar 5.52 Profil Lembaga Kursus**

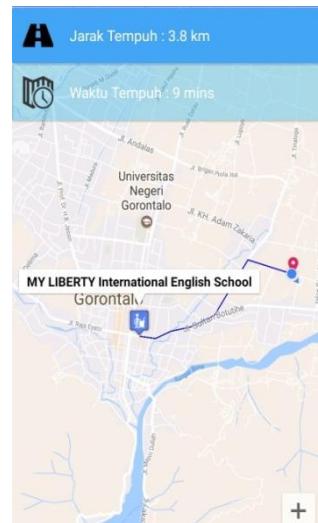
Untuk menampilkan rute menuju ke lokasi lembaga kursus yang dituju, cukup dengan klik ikon peta yang ada di sudut kanan bawah.



**Gambar 5.53 Ikon Peta**

Maka akan muncul tampilan rute dari tempat dimana pengguna berada menuju ke lokasi lembaga kursus yang dituju. Sebelum menggunakan fitur ini,

pastikan GPS pada *smartphone* dalam keadaan aktif. Selain menampilkan rute, fitur ini juga menampilkan jarak dan waktu tempuh ke lokasi lembaga kursus yang dituju.



**Gambar 5.54** Tampilan Rute Menuju Lokasi Lembaga Kursus

## 6. Kategori

Menu ini menampilkan lokasi lembaga kursus berdasarkan kategori lembaga kursus. Gambar berikut menampilkan pencarian lokasi lembaga kursus berdasarkan kategori Bahasa Inggris:



**Gambar 5.55** Pencarian Lokasi Lembaga Kursus Berdasarkan Kategori

Masyarakat dapat memilih kategori pada *combobox* yang tersedia kemudian klik Cari. Pada menu ini aplikasi menampilkan seluruh lembaga kursus yang termasuk dalam kategori yang dicari. Hal ini memudahkan masyarakat untuk mempertimbangkan lembaga kursus mana yang akan mereka pilih sesuai dengan keinginan dan kebutuhan mereka. Dengan meng-klik ikon biru, maka akan muncul profil lembaga kursus seperti pada gambar 5.48. Untuk menampilkan rute, lakukan hal yang sama seperti pada gambar 5.49.

## 7. Rute

Menu ini sangat membantu jika masyarakat ingin mencari informasi lembaga kursus namun lupa nama lembaga kursus yang dimaksud. Pada menu ini terdapat *combobox* yang menyimpan seluruh nama lembaga kursus yang sudah tersimpan di dalam sistem.



**Gambar 5.56** Menu Rute

Setelah menemukan nama lembaga kursus yang dicari, klik tombol Get Route, maka akan ditampilkan rute menuju ke lokasi lembaga kursus yang dimaksud seperti tampilan pada gambar 5.50.

## 8. About

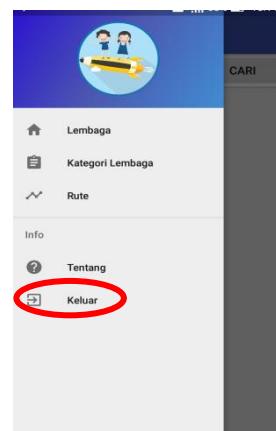
Menu ini menampilkan profil pembuat aplikasi SIGOPLEK.



**Gambar 5.57** Tampilan About

## 9. Keluar

Menu ini digunakan untuk keluar dari aplikasi SIGOPLEK.



**Gambar 5.58** Menu Keluar

## **BAB VI**

### **KESIMPULAN DAN SARAN**

#### **6.1 Kesimpulan**

Dari hasil pembahasan dan pembuatan aplikasi Sistem Informasi Geografis Pencarian Lokasi Lembaga Kursus, dapat disimpulkan bahwa :

1. Penelitian ini berhasil menerapkan Google Maps API untuk membangun Aplikasi SIG Pencarian Lokasi Lembaga Kursus di Kota Gorontalo.
2. Dengan adanya aplikasi ini memudahkan masyarakat dalam mencari informasi tentang Lembaga Kursus yang ada di Kota Gorontalo.
3. Dengan aplikasi ini memudahkan pihak Dinas Pendidikan dalam memantau lembaga kursus yang ada di Kota Gorontalo.

#### **6.2 Saran**

Berdasarkan hasil penelitian, aplikasi ini masih terdapat kekurangan, sehingga masih dapat dilakukan pengembangan. Beberapa hal yang dapat dikembangkan dari aplikasi ini adalah:

1. Dari segi penampilan. Aplikasi ini masih nampak sederhana sehingga masih membutuhkan desain yang lebih baik seperti penempatan menu-menu (*button*) serta tampilan utama bagi *user* agar terlihat lebih menarik dari segi warna, tulisan, *layout*, dll.
2. Menambahkan fasilitas pendaftaran *online*. Dengan penambahan fasilitas ini, diharapkan dapat lebih memudahkan masyarakat dalam hal pendaftaran,

sehingga masyarakat tidak perlu datang ke lembaga kursus untuk mendaftar.

3. Tambahan fitur cetak daftar lembaga kursus untuk memudahkan pihak PNFI sebagai laporan lembaga kursus yang ada di kota Gorontalo.
4. Aplikasi ini belum dibagikan secara umum sehingga diharapkan bagi peneliti selanjutnya untuk bisa membagikan aplikasi ini untuk kepentingan umum melalui *playstore*.
5. Titik koordinat aplikasi ini belum terpusat di Kota Gorontalo, sehingga masih dibutuhkan penambahan fitur agar titik lokasi terpusat langsung ke wilayah Kota Gorontalo.

## DAFTAR PUSTAKA

Bambang Yuwono, Agus Sasmito Aribowo, Febri Arif Setyawan. 2015. *Sistem Informasi Geografis Berbasis Android Untuk Pariwisata Di Daerah Magelang.* Skripsi tidak diterbitkan. Yogyakarta : UPN Veteran.

Bidang PNFI (Pendidikan Non Formal Informal). 2014. *Data Lembaga Kursus.* Gorontalo: Dinas Pendidikan Nasional Kota Gorontalo

Fadhoelor Rohman, Agung Budi Cahyono. 2013. *Mobile GIS Fasilitas Umum Untuk Pengguna Jalan Berbasis Android .* Skripsi tidak diterbitkan. Surabaya : Institut Teknologi Sepuluh Novermber (ITS).

Ifnu Bima. 2011. *Java Desktop Aplikasi POS Berarsitektur Three Tier Menggunakan Swing, Hibernate dan Spring.* E-Book. Singapore.

Ir. Aja Muhammad Ikram. 2012. *Lembaga Kursus dan Pelatihan.* (<http://bloganeuk-pidie.blogspot.co.id/2012/08/lembaga-kursus-dan-pelatihan.html> diakses pada 19 November 2016)

Jogiyanto, HM. 2005. *Data Flow, Proses, data store.* Yogyakarta.

Juwairiah, Ial Irwan Arahman, Budi Santosa. 2013. *Mobile GIS Layanan Informasi Lokasi Penting Kota Surakarta Berbasis Android.* Skripsi tidak diterbitkan. Yogyakarta : UPN Veteran.

Meides Palabiran, Dedi Cahyadi, Zainal Arifin. 2015. *Sistem Informasi Kuliner, Seni dan Budaya Kota Balikpapan Berbasis Android.* Skripsi tidak diterbitkan. Samarinda : Universitas Mulawarman.

Nurhadi TF. 2011. *Mari Kenali Pengertian Dan Kegunaan Aplikasi IDE.* (<http://boboy09.wordpress.com/2011/07/06/mari-kenali-pengertian-dan-kegunaan-aplikasi-ide/> diakses 3 Oktober 2016).

Pandu Dryad. 2015. *Urutan Tingkatan Versi Nama Android:* (<http://techijau.com/urutan-versi-nama-android/> diakses 26 September 2016).

Rahmat, Husein. 2007, *Konsep Dasar System Informasi Geografis:* Rahmat, Husein. 2007, *Konsep Dasar System Informasi Geografis:* (<http://ftsi.files.wordpress.com/2008/04/rahmat-sig.pdf/> diakses 17 September 2016).

Rastuti, Leon Andretti Abdillah, Eka Puji Agustini. 2015. *Sistem Informasi Geografis Potensi Wilayah Kabupaten Banyuasin Berbasis Web.* Skripsi tidak diterbitkan. Palembang : Universitas Bina Darma.

Roger S P. 2002. *Teknik Pengujian Sistem*. Yogyakarta: Andi Offset.  
Sewajarnya Saja. 2014. *Pengertian Netbeans*.  
[\(<http://sewajarnyasaja.blogspot.co.id/2014/10/pengertian-netbeans.html>\)](http://sewajarnyasaja.blogspot.co.id/2014/10/pengertian-netbeans.html) diakses 4 Oktober 2016).

Tim Penyusun . 2015. *Buku Pedoman Penulisan Proposal dan Skripsi*. Gorontalo : Universitas Ichsan Gorontalo.

Whiten, Jeffery L. et al. 2004. *Metode Desain & Analisis Sistem*. Edisi 6. Yogyakarta : Andi Offset.

Windu Gata, Grace Gata. 2013. *Sukses Membangun Aplikasi Penjualan dengan Java*, Jakarta : PT Elex Media Komputindo.

Yusuf Mufti. 2015. *Panduan Mudah Pengembangan Google Map Android*. Yogyakarta : ANDI Yogyakarta.

## LISTING PROGRAM

### 1. Halaman Utama

#### a. Tampilan (HTML)

```
<div ng-show="authenticated" class="col-xs-12 col-md-12">
<ol><li>SelamatDatang {{account.username}}</li>
</ol>
</div>
<div class="col-xs-12 col-md-12">
<ol>
<li><a href="javascript:void(0)"><i class="glyphicon glyphicon-map-marker"></i> LokasiLembagaKursus Di Kota Gorontalo </a></li>
</ol>
</div>
<div class="col-xs-12 col-md-12">
<select ng-model="searchKategori" class="form-control" ng-change="updateDataLembaga()">
<option value="">All</option>
<option ng-repeat="k in dataKategori" value="{{k.nama}}>{{k.nama}}</option>
</select>
<input type="text" class="form-control" placeholder="Search" ng-model="search" ng-keyup="updateDataLembaga()"/>
</div>
<ui-gmap-google-map class="col-md-12"
center="map.center"
zoom="map.zoom"
dragging="map.dragging"
bounds="map.bounds"
events="map.events"
options="map.options"
pan="true"
control="map.control">
<ui-gmap-marker ng-repeat="m in dataLembaga" coords="m" icon="m.icon"
idKey="m.id" options="m.options">
<ui-gmap-window>
<div class="col-lg-12 col-xs-12" style="padding-left: 0px;padding-right: 0px;padding-bottom: 0px;">
<!-- small box -->
<div class="small-box bg-blue">
<div class="inner">
<h4>Lembaga {{m.nama}}</h4>
<p>Ketua {{m.ketua}}</p>
</div>
<a href="#/lembaga-detail/{{m.id}}" class="small-box-footer">
More info <i class="fa fa-arrow-circle-right"></i>
</a>
</div>
</div>
</ui-gmap-window>
</ui-gmap-marker>
</ui-gmap-google-map>
```

## b. Java Script

```
$scope.updateDataLembaga();  
  
angular.extend($scope, {  
    map:{  
        show:true,  
        control:{},  
        version:"unknown",  
        heatLayerCallback:function(layer){  
            //set the heat layers backend data  
            var mockHeatLayer = newMockHeatLayer(layer);  
        },  
        showTraffic:true,  
        showBicycling:false,  
        showWeather:false,  
        showHeat:false,  
        center:{  
            latitude:0.702704,  
            longitude:122.493358  
        },  
        options:{  
            streetViewControl:false,  
            panControl:false,  
            maxZoom:20,  
            minZoom:3,  
            mapTypeId:google.maps.MapTypeId.SATELLITE  
        },  
        zoom:9,  
        dragging:false  
    }  
});  
})
```

```

.controller('home',function($scope, $location, $timeout,uiGmapLogger,
$http,uiGmapGoogleMapApi){
uiGmapGoogleMapApi.then(function(maps) {
    $scope.googleVersion = maps.version;
maps.visualRefresh = true;
));
    $scope.dataLembaga = [];
    $scope.dataKategori = [];
    $scope.showWindow = false;
    $scope.search = "";
    $scope.searchKategori = "";

    $scope.updateKategori = function(){
        $http.get("/api/kategori")
.success(function(response){
            $scope.dataKategori = response.content;
            console.log(response);
})
.error(function(response){
            alert(response.message);
})
);
};

    $scope.updateDataLembaga = function(){
        $http.get("/api/lembaga-kategori-nama?namaKategori=" +
$scope.searchKategori + "&namaLembaga=" + $scope.search)
.success(function(response){
            $scope.dataLembaga = [];
angular.forEach(response,function(value, index){
            $scope.dataLembaga.push({
                id: value.id,
                latitude:value.lat,
                longitude:value.longi,
                nama:value.nama,
                ketua:value.ketua,
                alamatLkp:value.alamatLkp,
                desc:value.desc,
                options:{
labelContent:<h6 style='color:blue;'>" + value.nama + "</h6>",
labelAnchor:"15 0",
icon:'image/workoffice.png'
}
));
});
});
};

    $scope.updateKategori();
}

);
});
);
);
.error(function(response){
            alert(response.message);
})
);
);
};

```

### c. Server

```
@RequestMapping(value = "/lembaga-kategori-nama", method =
RequestMethod.GET)
public Iterable<Lembaga> getByKategoriAndNama(
    @RequestParam(value = "namaKategori", required = true)
String namaKategori,
    @RequestParam(value = "namaLembaga", required = true)
String namaLembaga
) {
    returnId.findByKategoriAndNama(namaKategori, namaLembaga);
}
```

## 2. Login

### a. Server

```
@RequestMapping("/user")
public Principal user(Principal user) {
    return user;
}
```

### b. Tampilan Login (HTML)

```
<div class="col-md-4 col-md-offset-4">
<div class="login-wrapper">
<div class="box">
<div class="alert alert-danger" ng-show="controller.error">
    Username dan Password Salah
</div>
<div class="content-wrap">
<h6>Sign In</h6>
<form ng-submit="controller.login()">
<input class="form-control" ng-
model="controller.credentials.username" type="text" placeholder="Username">
<input class="form-control" ng-
model="controller.credentials.password" type="password" placeholder="Password">
<div class="action">
<button class="btn btn-primary
signup" type="submit">Login</button>
</div>
</form>
</div>
</div>
</div>
</div>
```

### c. Java Script

```
.controller('login',
function($rootScope, $http, $location, $route, $window, authService){
var self = this;
self.tab = function(route){
return $route.current&& route ===$route.current.controller;
};
var authenticate = function(credentials, callback){
var headers = credentials ?{
authorization:"Basic " + btoa(credentials.username + ":"+
credentials.password)
}: {};
$http.get('user',{
headers: headers
}).then(function(response){
if(response.data.name){
$rootScope.authenticated = true;
authService.loginConfirmed(response);
} else{
$rootScope.authenticated = false;
}
callback &&callback($rootScope.authenticated);
},function() {
$rootScope.authenticated = false;
callback && callback(false);
});
};
authenticate();
self.credentials = {};
self.login = function(){
authenticate(self.credentials,function(authenticated) {
if(authenticated){
console.log("Login succeeded");
$window.open("/", "_self");
self.error = false;
$rootScope.authenticated = true;
} else{
console.log("Login failed");
$location.path("/login");
self.error = true;
$rootScope.authenticated = false;
}
});
};
});});})
```

### 3. Input Data Lembaga Kursus

#### a. Tampilan Input (HTML)

```
<div class="col-xs-12 col-md-12"><h4 class="page-header">TambahLembaga</h4>
</div>
<div class="col-xs-12 col-md-12"><div class="row"></div>
<div class="box-body box-profile">
<ul class="list-group">
<li class="list-group-item bg-blue">
<i class="glyphicon glyphicon-map-marker"></i> MappingLembaga
</li><li class="list-group-item"><div class="row">
<ui-gmap-google-map class="col-xs-12 col-md-12"
center="map.center" zoom="map.zoom" dragging="map.dragging"
bounds="map.bounds" events="map.events" options="map.options"
pan="true" control="map.control">
<ui-gmap-marker coords="map.clickedMarker"
idKey="map.clickedMarker.id"
options="map.clickedMarker.options"></ui-gmap-marker></ui-gmap-
google-map>
</div></li><li class="list-group-item bg-
blue"><i class="glyphicon glyphicon-info-sign"> Data
Lembaga</i></li><li class="list-group-item">
<div class="row"><div class="col-xs-12 col-md-12">
<form class="form-horizontal" role="form" ng-submit="simpanLembaga()">
<div class="form-group"><label class="col-sm-2 control-
label">Latitude</label>
<div class="col-sm-10">
<input type="text" class="form-control" placeholder="Klik Maps" ng-
model="map.clickedMarker.latitude"/>
</div></div><div class="form-group">
<label class="col-sm-2 control-label">Longitude</label>
<div class="col-sm-10">
<input type="text" class="form-control" placeholder="Klik Maps" ng-
model="map.clickedMarker.longitude"/></div></div>
<div class="form-group">
<label class="col-sm-2 control-label">Nama</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="inputEmail3" ng-
model="lembaga.nama" placeholder="Nama Lembaga">
</div></div>
<div class="form-group">
<label class="col-sm-2 control-label">Ketua</label>
<div class="col-sm-10">
<input type="text" class="form-control" id="inputEmail3" ng-
model="lembaga.ketua" placeholder="KetuaLembaga">
</div></div><div class="col-sm-offset-2 col-sm-10">
<a href="#/list-lembaga" class="btn btn-
primary"><i class="glyphicon glyphicon-menu-left"></i> Kembali</a>
<button type="submit" class="btn btn-primary pull-right">Simpan</button>
</div></div></form></div>
</div>
</li>
</ul>
</div>
</div>
```

## b. Java Script

```
$scope.simpanLembaga = function() {
    $scope.data = {
        lat: $scope.map.clickedMarker.latitude,
        longi: $scope.map.clickedMarker.longitude,
        nama: $scope.lembaga.nama,
        ketua: $scope.lembaga.ketua
    };
    $http.post("/api/lembaga", $scope.data)
        .success(function(data) {
            $scope.lembaga = "";
            alert("Tersimpan");
        })
        .error(function(data) {
            alert(data.message);
        });
};
```

## c. Server

```
@RequestMapping(value = "/lembaga", method =
RequestMethod.POST)
    @ResponseStatus(HttpStatus.OK)
public void simpan(@RequestBody Lembaga lembaga) {
    ld.save(lembaga);
}
```

#### 4. Pencarian Rute

##### a. Tampilan Android (XML)

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/coordinatorFragmentPosisi"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/l1_map_posisi"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <fragment
            android:id="@+id/map_rute"
            android:name="com.google.android.gms.maps.SupportMapFragment"
            android:layout_width="match_parent"
            android:layout_height="match_parent"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <android.support.v7.widget.CardView xmlns:card_view="http://schemas.android.com/apk/res-auto"
            android:id="@+id/jarak_tempuh_stats"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="1dp"
            card_view:cardBackgroundColor="@color/blue">

            <RelativeLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:padding="10dp">

                <ImageView
                    android:id="@+id/jarak_tempuh_icon"
                    android:layout_width="40dp"
                    android:layout_height="40dp"
                    android:adjustViewBounds="true"
                    android:src="@mipmap/ic_road"/>
            
```

```
<TextView
    android:id="@+id/jarak_tempuh_tv"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerVertical="true"
    android:layout_marginLeft="16dp"
    android:layout_toRightOf="@+id/jarak_tempuh_icon"
    android:text="JarakTempuh"
    android:textColor="@color/white"
    android:textSize="16sp"/>
</RelativeLayout>
</android.support.v7.widget.CardView>
<android.support.v7.widget.CardView xmlns:card_view="http://schemas.android
.com/apk/res-auto"
    android:id="@+id/waktu_pejalan_stats"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="1dp"
    card_view:cardBackgroundColor="@color/wallet_highlighted_text_holo_light">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="10dp">

        <ImageView
            android:id="@+id/waktu_pejalan_icon"
            android:layout_width="40dp"
            android:layout_height="40dp"
            android:adjustViewBounds="true"
            android:src="@mipmap/ic_waktu_tempu"/>

        <TextView
            android:id="@+id/waktu_pejalan_tv"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerVertical="true"
            android:layout_marginLeft="16dp"
            android:layout_toRightOf="@+id/waktu_pejalan_icon"
            android:text="WaktuTempuh"
            android:textColor="@color/white"
            android:textSize="16sp"/>
    </RelativeLayout>
    </android.support.v7.widget.CardView>
</LinearLayout>
</android.support.design.widget.CoordinatorLayout>
```

## b. Java

```
package com.wordpress.ria.pemetaankursusclient.activity;

import android.app.AlertDialog;
import android.graphics.Color;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.FragmentActivity;
import android.util.Log;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.BitmapDescriptor;
import com.google.android.gms.maps.model.BitmapDescriptorFactory
;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.maps.model.PolylineOptions;
import com.google.gson.Gson;
import com.wordpress.ria.pemetaankursusclient.R;
import com.wordpress.ria.pemetaankursusclient.helper.VolleyController;
import com.wordpress.ria.pemetaankursusclient.json.Steps;
import com.wordpress.ria.pemetaankursusclient.util.RestUri;
import com.wordpress.ria.pemetaankursusclient.util.Utils;

import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

public class RuteActivity extends FragmentActivity{

    private static final String TAG = "RUTEACTIVITY";

    private GoogleMap mMap;
    private double latAwal, latTujuan, longiAwal, longiTujuan;
```

```
private com.wordpress.nofipriyanto.pemetaankursusclient.json.Res
ponse response;
private String tujuan;
private List<LatLng> latLngList;
private ProgressDialog progressDialog;
private Bundle data;
private TextView tvJarakTempuh, tvWaktuTempuh;

    @Override
protected void onCreate(@Nullable Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_rute);
findViewById();
latLngList=new ArrayList<>();
progressDialog=new ProgressDialog(this);
progressDialog.setMessage("Loading...");
setUpMapIfNeeded();
data = getIntent().getExtras();
latAwal=data.getDouble("latAwal");
longiAwal=data.getDouble("longiAwal");
latTujuan=data.getDouble("latTujuan");
longiTujuan=data.getDouble("longiTujuan");
tujuan=data.getString("tujuan");
getDirections(RestUri.maps.GET_GOOGLE_API(latAwal, longiAwal, lat
Tujuan, longiTujuan));
}

private void findViewById() {
tvJarakTempuh=(TextView)findViewById(R.id.jarak_tempuh_tv);
tvWaktuTempuh=(TextView)findViewById(R.id.waktu_pejalan_tv);
}

@Override
protected void onResume() {
super.onResume();
setUpMapIfNeeded();
}

private void setUpMapIfNeeded() {
// Do a null check to confirm that we have not already
instantiated the map.
if(mMap==null){
// Try to obtain the map from the SupportMapFragment.
mMap=((SupportMapFragment) getSupportFragmentManager().findFragm
entById(R.id.map_rute))
.getMap();
// Check if we were successful in obtaining the map.
if(mMap!=null){
setUpMap();
}
}
}
```

```
private void setUpMap() {
    mMap.setMyLocationEnabled(true);
    mMap.getUiSettings().setZoomControlsEnabled(true);
    //        mMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
}

private void getDirections(String URL) {
    JsonObjectRequest request
    =new JsonObjectRequest(Request.Method.GET,
                          URL,
    new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject jsonObject) {
            Log.e("Null", jsonObject.toString());
            progressDialog.dismiss();
            response
            =new Gson().fromJson(jsonObject.toString(),
com.wordpress.nofipriyanto.pemetaankursusclient.json.Response.class);
            Log.e(TAG,""+response.getRouteList().get(0).getLegsList().get(0)
            .getStepsList().get(0).getEndLocation().getLat());

            fetchDirection(response.getRouteList().get(0).getLegsList().get(0)
            .getStepsList());
            tvJarakTempuh.setText("JarakTempuh : "+response.getRouteList().get(0).getLegsList().get(0).getDistance()
            .getText());
            tvWaktuTempuh.setText("WaktuTempuh : "+response.getRouteList().get(0).getLegsList().get(0).getDuration()
            .getText());
        }
    },
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError volleyError) {
            progressDialog.dismiss();
            Log.d("error",""+volleyError.toString());
        }
    });
    progressDialog.show();
    VolleyController.getInstance().addToRequestQueue(request);
}
```

```
private void fetchDirection(List<Steps> stepsList) {
    for(Steps data:stepsList) {

        LatLngList.add(new LatLng(data.getStartLocation().getLat(),data.
        getStartLocation().getLng()));

        List<LatLng> decodePoly=Utils.decodePoly(data.getPolyline().getP
        oints());
        for(LatLng point:decodePoly)
            LatLngList.add(new LatLng(point.latitude,point.longitude));
        LatLngList.add(new LatLng(data.getEndLocation().getLat(),data.ge
        tEndLocation().getLng()));
    }
    drawDirectionToMap(LatLngList);
}

private void drawDirectionToMap(List<LatLng> LatLngList) {
    PolylineOptions line
    =new PolylineOptions().width(3).color(Color.BLUE);
    for(int i=0;i<LatLngList.size();i++) {
        line.add(LatLngList.get(i));
    }
    mMap.addPolyline(line);
    mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(
    LatLngList.get(0).latitude,LatLngList.get(0).longitude),14));
    mMap.addMarker(
        new MarkerOptions()
        .position(new LatLng(LatLngList.get(0).latitude,LatLngList.get(0)
        .longitude))
        .title("Posisi
Anda").icon(BitmapDescriptorFactory.fromResource(R.drawable.ic_
marker)));
    ;
    int index =LatLngList.size()-1;
    mMap.addMarker(
        new MarkerOptions()
        .position(new LatLng(LatLngList.get(index).latitude,LatLngList.g
        et(index).longitude))
        .title(tujuan).icon(BitmapDescriptorFactory.fromResource(R.draw
        able.ic_workoffice)));
    );
    Toast.makeText(this,"JalurBerhasil Di
Gambar",Toast.LENGTH_SHORT).show();
}
}
```



## PEMERINTAH KOTA GORONTALO DINAS PENDIDIKAN

Jalan. Dewi Sartika No. 13 ☎ (0435). 821441 Fax. 822625 Kode Pos 96128  
Website [Http : disdikgrn.schoolmed.id](http://disdikgrn.schoolmed.id)

### REKOMENDASI

Nomor : 423.4 / Disdik. Sekrt / 5161

#### TENTANG

PENGAMBILAN DATA PENELITIAN MAHASISWA  
UNIVERSTAS ICHSAN GORONTALO  
FAKULTAS ILMU KOMPUTER/TEKNIK INFORMASI  
DI DINAS PENDIDIKAN KOTA GORONTALO

Menunjuk surat dari Ketua Lemlit Fakultas Ilmu Komputer/Teknik Informatika Universitas Ichsan Gorontalo Nomor : 050/Lemlit-Unisan/1/2016 tanggal 7 Nopvember 2016, Perihal Permohonan Rekomendasi, maka dengan ini Kepala Dinas Pendidikan Kota Gorontalo memberikan izin/rekomendasi Kepada :

Nama	:	RIA PRATIWI ULOLI
NIM	:	T.31.14.294
Fak./Jrs.	:	Ilmu Komputer/Teknik Informatika

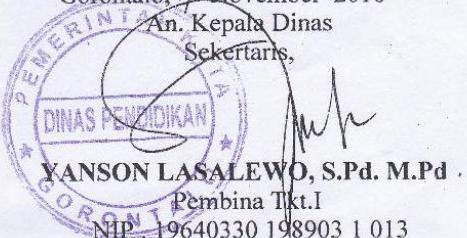
Untuk mengadakan Pengambilan Data Penelitian dengan rangka Proposal (*Skripsi*) yang berjudul : “*Aplikasi Sistm Informasi Geografis Pencarian Lokasi Lembaga Kursus di Kota Gorontalo Menggunakan Google Maps API berbasis Android. (Lokasi Pengambilan data penelitian di Dinas Pendidikan Kota Gorontalo)*”.

Sehubungan dengan hal tersebut diatas pihak kami menyetujui / tidak keberatan dengan ketentuan sebagai berikut :

- a. Kegiatan penelitian tersebut agar dikonsultasikan dengan Kepala Dinas
- b. Kegiatan penelitian tersebut tidak mengganggu kegiatan Aktifitas Dinas
- c. Rekomendasi ini berlaku sampai dengan tanggal Desember 2016.

Demikian rekomendasi ini diberikan untuk perlunya.

Gorontalo, 17 November 2016



Tembusan :

1. Ketua Jrs. Ilmu Komputer / Teknik Informatika
2. Arsip.

## **DAFTAR RIWAYAT HIDUP**



**Ria Pratiwi Uloli**, lahir di Jayapura pada tanggal 17 September 1993 Putri Sulung dari pasangan Bapak Abd. Rahman Uloli, S.Sos dan Ibu Dra. Hj. Salmawati Usuli.

### **Riwayat Pendidikan :**

1. Tahun 2005, menyelesaikan Pendidikan di Sekolah Dasar Negeri 2 Duano, Kec. Suwawa Tengah Kab. Bone Bolango.
2. Tahun 2008, menyelesaikan Pendidikan di Sekolah Menengah Pertama Negeri 1 Suwawa, Kec. Suwawa Kab. Bone Bolango.
3. Tahun 2011 menyelesaikan Pendidikan di Sekolah Menengah Kejuruan Negeri 3 Gorontalo, Kec. Kota Tengah, Kota Gorontalo.
4. Tahun 2014 menyelesaikan Pendidikan D-III di Politeknik Gorontalo, Kecamatan Botupingge, Kabupaten Bone Bolango.

Tahun 2014, mendaftar dan diterima menjadi Mahasiswa Lanjut Studi di Fakultas Ilmu Komputer, Universitas Ichsan Gorontalo.