

KARYA ILMIAH

IMPLEMENTASI ANDROID PADA PENGEMBANGAN APLIKASI PERSONAL ASSISTANT

OLEH

MAKSY SENDIANG



**KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MANADO
2017**



KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MANADO

Alamat Kantor : Kampus Politeknik Ds.Buha Manado-95254

Telp (0431)815288, 815212. Fax (0431)811566,815192

Website : www.polimdo.ac.id

LEMBAR PENGESAHAN

**IMPLEMENTASI ANDROID PADA PENGEMBANGAN
APLIKASI PERSONAL ASSISTANT**

OLEH

MAKSY SENDIANG

Setelah diperiksa dan dievaluasi, maka Karya Ilmiah dengan judul tersebut diterima dan dinyatakan sah sebagai hasil karya staf pengajar Politeknik Negeri Manado.

Manado, Oktober 2017



Kepala UPT Perpustakaan,

Ir. Telly Hetty Isye Kondo, M.Si

NIP. 19700517 200312 2 001

Ketua Jurusan Teknik Elektro,

Fanny J. Doringin, ST., MT

NIP.19670430 199203 1 003

KATA PENGANTAR

Puji dan syukur patut dinaikkan kepada yang Maha Besar Tuhan atas anugerah dan kasihNYA sehingga penyusunan karya ilmiah ini boleh terselesaikan dengan baik. Karya Ilmiah ini disusun dengan maksud untuk menjadi salah satu referensi bahan ajar kegiatan perkuliahan di Politeknik Negeri khususnya Program Studi D4 Teknik Informatika dan D3 Teknik Komputer

Pada kesempatan ini perkenankan penulis mengucapkan terima kasih kepada pihak yang telah membantu sehingga penulisan karya ilmiah ini dapat terselesaikan dengan baik. Ucapan terima kasih disampaikan kepada

1. Ir.Ever N.Slat,MT; Direktur Politeknik Negeri Manado
2. Dra.Mareyke Alelo,MBA; Pembantu direktur Bidang Akademik
3. Ir.Telly Hetty Isye Kondojo,M.Si; Kepala UPT Perpustakaan Politeknik Negeri Manado
4. Fanny Doringin,ST.,MT; Ketua Jurusan Teknik Elektro
5. Rekan – rekan dosen di Jurusan Teknik Elektro

Akhirnya semoga kehadiran karya ilmiah ini akan memberikan nilai tambah dalam upaya meningkatkan mutu kegiatan akademik di Politeknik Negeri Manado. Saran dari para pembaca untuk kesempurnaan karya ilmiah ini sangat diharapkan dan mari kita terus berkarya di bidang pendidikan untuk kejayaan tanah air tercinta.

Manado, Oktober 2017

DAFTAR ISI

Lembar Pengesahan	i
Kata Pengantar	ii
BAB I PENDAHULUAN	
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	1
1.3 Tujuan Penelitian	2
1.4 Manfaat Penelitian	2
BAB II TINJAUAN PUSTAKA	
2.1 Android	3
2.1.1 Arsitektur Android	3
2.1.2 Komponen Aplikasi Android	4
2.2 Personal Assistant	4
2.3 Java	5
2.4 Google Maps API	6
2.5 Database	6
BAB III METODOLOGI	
3.1 Metodologi RAD	8
3.2 Unified Modelling Language	8
BAB IV PEMBAHASAN	
4.1 Analisis Persyaratan	10
4.1.1 Spesifikasi Sistem	11
4.1.2 Manajemen Resiko	11
4.2 Use Case Model	12
4.3 Mengidentifikasi Hubungan Antar Objek	13
4.4 Implementasi Antarmuka Pengguna	13
4.5 Pengujian Aplikasi	21
BAB V KESIMPULAN DAN SARAN	
5.1 Kesimpulan	23
5.2 Saran	23
DAFTAR PUSTAKA	24

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Kemajuan teknologi pada masa sekarang ini sudah berkembang pesat. Salah satu contohnya adalah perkembangan teknologi komunikasi seperti telepon seluler. Berbagai jenis telepon seluler telah beredar pada masa sekarang ini dan juga beberapa inovasi dari masing – masing perusahaan telepon seluler yang terus dikembangkan untuk menarik minat dari para konsumen.

Telepon seluler pada saat ini tidak hanya sekedar digunakan untuk mengirimkan pesan, e-mail, telepon ataupun browsing internet melainkan sudah berkembang untuk dapat menunjang kehidupan sehari – hari seperti untuk hiburan ataupun pekerjaan. Android smartphone adalah salah satu dari sekian banyak telepon seluler yang beredar saat ini dengan beberapa keunggulannya. Keunggulan dari Android smartphone ditunjukkan dari tingginya pemasaran untuk saat ini. Beberapa aplikasi reminder ponsel pada umumnya hanya menampilkan notification reminder pada layar ponsel. Pada masa sekarang ini, mengingat banyaknya jadwal kegiatan/agenda yang tidak diatur dengan baik dikarenakan padatnya aktivitas yang dilakukan user, maka dibutuhkan aplikasi Personal Assistant yang dapat memudahkan pengguna dalam mengatur jadwal kegiatan atau agenda mereka, dimana aplikasi tersebut secara langsung dapat mengirimkan pesan atau e-mail otomatis pada waktu yang telah diatur. Juga pengaturan pengingat lokasi yang dituju sehingga dapat meningkatkan efektivitas penggunaan telepon seluler tersebut.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan di atas, maka rumusan masalahnya adalah bagaimana membangun aplikasi yang dapat membantu pengguna dalam pengaturan jadwal kegiatan dan pengingat tempat yang akan dituju melalui telepon seluler?

1.3. Tujuan Penelitian

Tujuannya adalah untuk membangun aplikasi yang dapat membantu pengguna dalam pengaturan jadwal kegiatan dan pengingat tempat yang akan dituju melalui telepon seluler.

1.4. Manfaat Penelitian

Manfaat yang bisa didapat oleh pengguna aplikasi :

1. Pengguna dapat mengatur jadwal kegiatan mereka dengan pengiriman pesan atau *e-mail* secara otomatis.
2. Pengguna dapat mengatur *profile* dari telepon seluler secara otomatis sehingga memudahkan pengguna dalam melakukan kegiatan.
3. Pengguna dapat mengatur jadwal yang diinginkan pada waktu kapanpun karena aplikasi tersebut diimplementasikan dalam telepon seluler.
4. Pengguna mendapatkan *notification* jika mendekati tempat yang akan dituju sebagai *reminder* sewaktu di perjalanan

BAB II

TINJAUAN PUSTAKA

2.1 Android

By Lessard and Kessler (2010), *Android is an operating system (OS) developed by the Open Handset Alliance (OHA). The Alliance is a coalition of more than 50 mobile technology companies ranging from handset manufactures and service providers to semiconductor manufacturers and software developers, including Acer, ARM, Google, eBay, HTC, Intel, LG Electronics, Qualcomm, Sprint, and T-Mobile.*

Menurut Supardi (2011), Android merupakan sebuah sistem operasi perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. Beberapa pengertian lain dari Android, yaitu :

1. Merupakan *platform* terbuka (*Open Source*) bagi para pengembang (*Programmer*) untuk membuat aplikasi.
2. Merupakan sistem operasi yang dibeli *Google Inc.* dari Android Inc.
3. Bukan bahasa pemrograman, akan tetapi hanya menyediakan lingkungan hidup atau *run time environment* yang disebut DVM (*Dalvik Virtual Machine*) yang telah dioptimasi untuk *device* / alat dengan sistem memori yang kecil.

Konsep Android dapat dijelaskan sebagai berikut :

2.1.1 Arsitektur Android

Arsitektur Android terdiri atas :

1. *Applications*

Lapisan ini adalah lapisan aplikasi, serangkaian aplikasi akan terdapat pada perangkat *mobile*. Aplikasi inti yang telah terdapat pada Android termasuk kalender, kontak, SMS, dan lain sebagainya. Aplikasi-aplikasi ini ditulis dengan bahasa pemrograman Java.

2. *Application Frameworks*

Pengembang aplikasi memiliki akses penuh ke Android sama dengan aplikasi inti yang telah tersedia. Pengembang dapat dengan mudah mengakses informasi lokasi, mengatur alarm, menambahkan pemberitahuan ke status bar dan lain sebagainya. Komponen-komponen yang termasuk di dalam *Application Frameworks* adalah sebagai berikut : *Views, Content Provider, Resource Manager, Notification Manager* dan *Activity Manager*.

3. *Libraries*

Satu set *library* dalam bahasa C/C++ yang digunakan oleh berbagai komponen pada sistem Android.

4. *Android Run Time*

Satu set *library* inti yang menyediakan sebagian besar fungsi yang tersedia di *libraries* inti dari bahasa pemrograman *Java*. Setiap aplikasi akan berjalan sebagai proses sendiri pada *Dalvik Virtual Machine*.

5. *Linux Kernel* Android bergantung pada *Linux Versi 2.6* untuk layanan sistem inti seperti keamanan, manajemen memori, manajemen proses, *network stack*, dan model *driver*. Kernel juga bertindak sebagai lapisan antara *hardware* dan seluruh *software*

2.1.2 **Komponen Aplikasi Android**

Ada empat komponen pada aplikasi Android sebagai berikut :

1. *Activities* dipakai untuk menampilkan aplikasi atau yang bertindak sebagai *user interface* (UI) saat aplikasi diperlihatkan kepada *user*.
2. *Service* merupakan komponen yang tidak memiliki *user interface* (tampilan program) tetapi *service* berjalan secara *backgrounds*.
3. *Broadcast Receiver* merupakan komponen yang berfungsi menerima dan bereaksi untuk menyampaikan notifikasi.
4. *Content Provider* merupakan komponen membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain.

2.2 **Personal Assistant**

According to Jeffress (2012), A personal assistant helps another individual accomplish daily tasks, manage finances, and maintain a reliable schedule.

Excellent organizational skills and the ability to multitask are essential qualities in a person who wants to become a personal assistant. An individual can master these traits by paying close attention to his or her own schedule and keeping a detailed journal of daily tasks and accomplishments. A good personal assistant must also be assertive, punctual, detail-oriented, and a good communicator.

2.3 Java

Java merupakan perangkat lunak produksi *Sun Microsystem Inc.* untuk pemrograman beberapa tujuan (*multi purpose*), *multiplatform* (dapat berjalan di beberapa sistem operasi), mudah dipelajari dan *powerful*. Aplikasi – aplikasi yang dapat dibuat dengan *Java*, meliputi *web programming* (pemrograman *web*), *Desktop Programming* (pemrograman desktop), *mobile programming* (pemrograman *mobile/handphone*).

Android menggunakan *Java* karena Android hanya menyediakan lingkungan *runtime*/sebagai *interpreter*, dimana kode sumber yang telah kita *compile* dengan *compiler Java* akan dioptimasi oleh *Delvik*. Sebuah *virtual machine* yang memang dibuat khusus untuk menjalankan kode – kode program yang dibuat dengan bahasa pemrograman *Java*, yang tentunya berbentuk sebuah *Class*. Kemudian oleh *dex tools* (merupakan bagian dari *Delvik Virtual Machine*) mengubah *Java Class* yang telah di-*compile* oleh *Java Compiler* ke lingkungan *native* yang berbentuk **.dex* format (*Dalvik executable*), yang teroptimasi untuk lingkungan perangkat keras dengan komputasi rendah.

Beberapa *software* utama yang digunakan diantaranya adalah sebagai berikut :

1. *JDK (Java Development Kit)*

JDK diperlukan untuk dapat men-*develop* aplikasi Android. *JDK* yang sudah termasuk di dalamnya *JRE (Java Runtime Environment)* untuk bisa menjalankan program yang dibuat.

2. *SDK Android (Software Development Kit)*

Android *SDK* merupakan alat atau *tool* yang digunakan untuk membuat aplikasi *platform* Android menggunakan bahasa pemrograman *Java*.

3. *IDE Android (Integrated Development Environment)*

Untuk membangun aplikasi berbasis Android menggunakan *editor* teks biasa, seperti *Notepad* atau *Wordpad* atau *DroidDraw*.

Google menyarankan kita untuk menggunakan *Integrated Development Environment* (IDE) *Eclipse* dengan tujuan untuk mengurangi waktu pembelajaran dan memudahkan dalam melakukan *compile*, dan juga sudah disediakan *emulator* kita untuk membuat, Android yang memudahkan dilakukan *running test* dan *debug* pada aplikasi yang sedang dibangun.

2.4 Google Maps API

Google Maps adalah layanan gratis *google* yang cukup populer. Kita dapat menambahkan fitur *Google Maps* yang sudah ada dalam web dengan *Google Maps API*.

Menurut Prabowo (2010), *Google Maps API* adalah *library JavaScript*. Menggunakan *Google Maps API* sangat mudah. Yang dibutuhkan hanyalah pengetahuan tentang HTML dan *JavaScript*, serta koneksi internet. Dengan menggunakan *Google Maps API*, kita dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga dapat fokus hanya pada data.

2.5 Database

Database yang digunakan adalah *SQLite*. Menurut Winarno dan Zakin (2011), *SQLite* merupakan salah satu RDBMS yang sama dengan RDBMS lainnya, seperti *MySQL*, *PostgreSQL*, *Oracle*. Aplikasi ini cukup ringan dan mudah dalam penginstalan, sehingga cocok digunakan untuk aplikasi yang membutuhkan *memory* dan *library* yang kecil seperti aplikasi Android.

Beberapa fitur yang ada pada *SQLite*, antara lain :

1. *Serverless*, artinya *SQLite* tidak memerlukan proses pada *server*, melainkan hanya sebuah *file* yang diakses oleh library *SQLite*.
2. *Zero Configuration*, artinya ketika membuat sebuah *database* seperti membuat *file* biasa.
3. *Cross-platform*, artinya semua *database* berada dalam sebuah *file cross-platform* dan tidak memerlukan administrasi.

4. *Self-contained*, artinya terdapat *library* yang mengandung keseluruhan *database* dan langsung terintegrasi pada aplikasi program.
5. *Transactional*, artinya *SQLite* memperbolehkan aksi penyimpanan melalui beberapa proses *thread*.
6. *Full featured*, artinya *SQLite* men-*support* sebagian besar standar SQL92 (SQL2).
7. *Highly reliable*, artinya tim pengembang *SQLite* telah mengembangkan dengan proses yang serius dan *testing* yang ketat.

BAB III

METODOLOGI

3.1. Metodologi RAD

Metodologi yang digunakan dalam pengembangan sistem adalah metodologi RAD. RAD adalah model proses pengembangan perangkat lunak yang bersifat inkremental terutama untuk waktu pengerjaan yang pendek. Model RAD adalah adaptasi dari model air terjun versi kecepatan tinggi dengan menggunakan model air terjun untuk pengembangan setiap komponen perangkat lunak.

Model RAD memiliki kelemahan sebagai berikut :

1. Untuk pembuatan sistem perangkat lunak dengan skala besar maka model RAD akan memerlukan sumber daya manusia yang cukup besar untuk membentuk tim-tim yang mengembangkan komponen-komponen.
2. Jika tidak ada persetujuan untuk mengembangkan perangkat lunak secara dengan cepat (*rapid*) maka proyek dengan model ini akan gagal karena hanya akan bingung mendefinisikan kebutuhan pelanggan (*customer*) atau *user*.
3. Jika sistem perangkat lunak yang akan dibuat tidak bisa dimodulkan (dibagi-bagi menjadi beberapa komponen) maka model RAD tidak dapat digunakan untuk membuat sistem perangkat lunak ini karena terlalu banyak campur tangan antar tim.
4. Model RAD tidak cocok digunakan untuk sistem perangkat lunak yang memiliki risiko teknis sangat tinggi, misalnya menggunakan teknologi baru yang belum banyak dikenal dan dikuasai pengembang.

3.2 *Unified Modelling Language (UML)*

Dalam pembuatan aplikasi ini, kaskas yang digunakan adalah UML 2.3. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

Ada beberapa diagram yaitu diantaranya : *use case diagram, class diagram, statechart diagram, activity diagram, sequence diagram, communication diagram, component diagram, object diagram, timing diagram, composite structure diagram, package diagram, interaction overview diagram* dan *deployment diagram*. Diagram yang digunakan untuk pengembangan perangkat lunak adalah *use case diagram, class diagram, activity diagram* dan *sequence diagram*.

BAB IV PEMBAHASAN

4.1. Analisis Persyaratan

Analisis persyaratan adalah fase pertama yang dilakukan dalam metodologi RAD. Fase ini terdiri dari pengumpulan dan pemrosesan data, spesifikasi pengguna (mengidentifikasi target pengguna dan menganalisis masalah serta kesempatan), spesifikasi sistem (mendefinisikan sistem dan mendokumentasikan spesifikasi fungsional sistem) dan studi kelayakan dalam bidang teknis, operasional, ekonomi dan penjadwalan serta manajemen resiko. Target pengguna dan prioritas persyaratan pengguna yang didapat dari hasil analisis ini adalah seperti pada table berikut :

Tabel 4.1 Daftar Target Pengguna

Pengguna	Tugas	Tanggung Jawab
<i>User yang menggunakan handphone Android (min versi Froyo)</i>	<i>User</i>	Orang yang menggunakan Aplikasi <i>Personal Assistant</i> pada Android

Tabel 4.2 Prioritas Persyaratan Pengguna

No	Kebutuhan Pengguna	Prioritas
1.	Pengguna dapat mengirimkan pesan/ <i>e-mail</i> dengan waktu yang telah diatur dan disimpan dalam pengaturan telepon seluler.	Tinggi
2.	Pengguna mendapatkan pemberitahuan atau pengingat lokasi sesuai dengan tempat yang dituju.	Tinggi
3.	Pengguna dapat mengatur aktivitas/kegiatan sehari-hari dengan baik.	Sedang

4.1.1 Spesifikasi Sistem

Aplikasi *Personal Assistant* pada Android ini bertujuan untuk mempermudah pengguna ponsel Android dalam mengatur kegiatan/aktivitas yang dijalankan setiap harinya. Juga mempermudah pengguna dalam mengingat tempat

yang akan dituju, sehingga aktivitas/kegiatan dapat dijadwalkan dan dijalankan dengan baik dan teratur sesuai dengan waktu yang telah diatur.

Adapun fitur-fitur yang disediakan adalah fitur *login*, fitur *add activity*, fitur *setting*, fitur *position line* dan fitur *logout*. Metodologi yang digunakan dalam pengembangan aplikasi ini adalah metodologi RAD dengan menggunakan UML sebagai pemodelan objek. Aplikasi ini menggunakan bahasa pemrograman *Java*, *SQLite* sebagai *database* dan *Eclipse* sebagai *editor* pemrograman *Java*.

Persyaratan fungsional system adalah :

1. Sistem harus mempunyai fitur *login* untuk melakukan verifikasi terhadap pengguna dan juga untuk menjaga keamanan.
2. Sistem harus dapat mengetahui ada berapa banyak aktivitas yang telah dilakukan sesuai tahun, bulan dan tanggal.
3. Sistem harus menspesifikasikan aktivitas yang *diinput* berdasarkan masing-masing *action*.
4. Sistem harus dapat menyimpan lebih dari 1 aktivitas pada setiap *action* yang tersedia.
5. Sistem harus mempunyai *notification*/pemberitahuan jika pesan/*e-mail* tidak tersampaikan pada orang yang dituju.
6. Sistem harus dapat memberikan *reminder* pada saat *user* mendekati radius lokasi yang akan dituju.

Adapun batasan masalah adalah sebagai berikut :

1. Aplikasi ini hanya dapat berjalan pada telepon seluler dalam keadaan aktif dalam kondisi koneksi internet yang baik dan pulsa yang mencukupi.
2. Aplikasi ini hanya dapat dijalankan pada telepon seluler berbasis Android minimal pada *Android versi 2.2 (Froyo)*.

4.1.2 Manajemen Resiko

Manajemen resiko diperlihatkan pada table berikut ini :

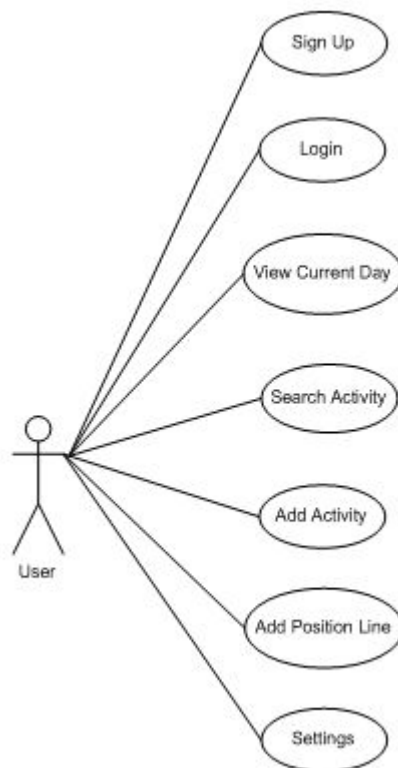
Tabel 4.3 Manajemen Resiko

No	Resiko	Penyebab	Antisipasi	Akibat
1	Fitur <i>Position line</i> tidak berjalan	Tidak adanya	<i>GPS</i> harus tetap dalam	Fitur <i>Position line</i>

	dengan baik	jaringan <i>GPS</i>	keadaan menyala	tidak dapat dijalankan dan harus menampilkan <i>message</i>
2	Terdapat kesalahan dalam aplikasi	Kurang teliti dalam pembuatan aplikasi	Memperbaiki kesalahan dalam aplikasi	Aplikasi tidak dapat berjalan dengan baik

4.2 Use Case Model

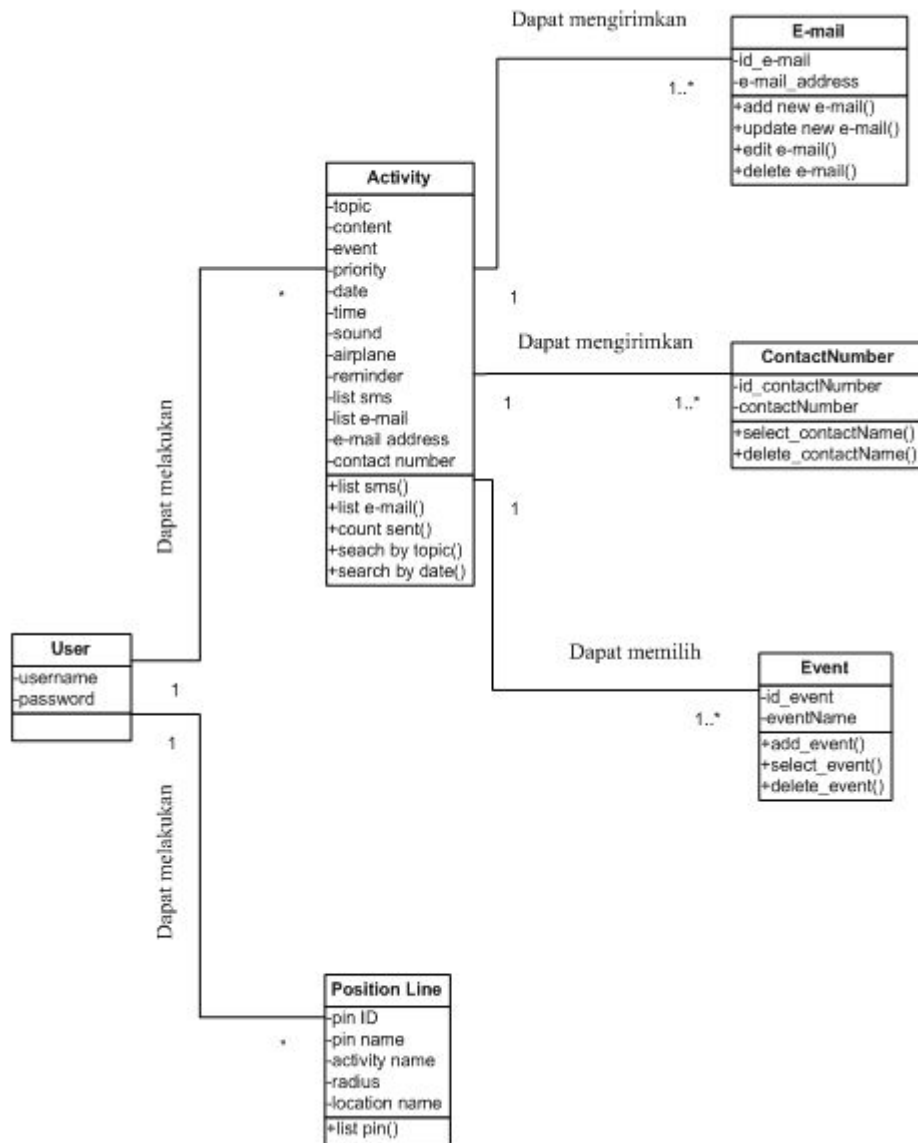
Diagram *use case (actor-use case)* sistem dapat dilihat pada gambar berikut ini



Gambar 4.1 Diagram Actor - Use Case

4.3 Mengidentifikasi Hubungan Antar Objek

Pada tahap ini, obyek-obyek yang telah teridentifikasi akan diorganisir untuk memperlihatkan relasi diantara obyek seperti yang digambarkan pada Class diagram berikut.



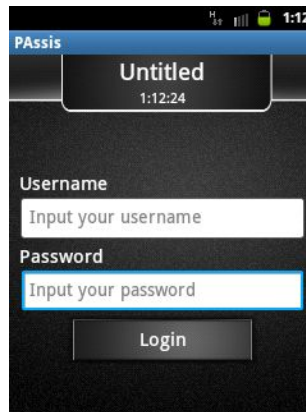
Gambar 4.2 Class Diagram

4.4 Implementasi Antarmuka Pengguna

Antarmuka pengguna ditunjukkan pada gambar di bawah ini.

Scene 01: Login

a. Antarmuka



Gambar 4.3 Antarmuka Login

b. Coding

Tabel 4.4. Scene Login

Tujuan Script	Keterangan
<i>Script untuk menampilkan Scene Login</i> <pre>private void Save(){ if(pwd.getText().toString().equals(cpwd.getText().toString())){ db.OpenDatabases(); db.InsertData("users", new String[]{uid.getText().toString(),pwd.getText().toString() ,"1"}, new String[]{"uid","pwd","status"}); } }</pre>	

Scene 02: View Current Day

a. Antarmuka



Gambar 4.4
Antarmuka View Current Day

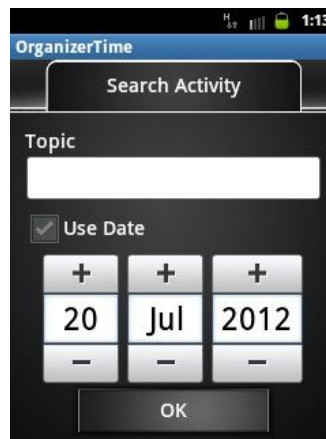
b. Coding

Tabel 4.5. Scene View Current Day

Tujuan Script	Keterangan
<p><i>Script untuk menampilkan Scene View Current Day</i></p> <pre> private void StartService(){ service_running = isMyServiceRunning(); //Toast.makeText(getApplicationContext(), Boolean.toString(service_running), Toast.LENGTH_LONG).show(); if(!service_running){ Intent intent = new Intent(Current_List.this, art.of.something.OT.NotifyService.class); Current_List.this.startService(intent); //service_start =true; //Toast.makeText(getApplicationContext(), Boolean.toString(service_running), Toast.LENGTH_LONG).show(); } </pre>	

Scene 03: Search Activity

a. Antarmuka



Gambar 4.5 Antarmuka Search Activity

b. Coding

Tabel 4.6. Scene Search Activity

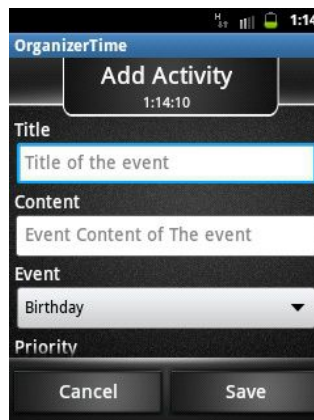
Tujuan Script	Keterangan
<p><i>Script untuk menampilkan Scene Search Activity</i></p> <pre> private void GetData(String filter){ </pre>	

Tujuan Script	Keterangan
<pre> db.OpenDatabases(); //dt=' "+toDateString(c)+"' d = db.GetData("todolist", new String[]{"e_type", "title", "dt", "tm", "_id", "status"}, filter, "dt ASC"); if(d.moveToFirst()){ int i = 1; if(!alist.isEmpty()) alist.clear(); do{ //(String title,String id,String dt,String tm,int counter,String tipe) boolean dump = false; String state = "1"; int Hour = c.get(Calendar.HOUR_OF_DAY); int Minute = c.get(Calendar.MINUTE); int Day = c.get(Calendar.DAY_OF_MONTH); int Month = c.get(Calendar.MONTH)+1; int Year = c.get(Calendar.YEAR); String []tm = d.getString(3).split(":"); String []dtm = d.getString(2).split("-"); if(Integer.parseInt(dtm[0])<Year) state="0"; else if(Integer.parseInt(dtm[0])>Year) state="1"; else{ if(Integer.parseInt(dtm[1])<Month) state="0"; else if(Integer.parseInt(dtm[1])>Month) state="1"; else{ if(Integer.parseInt(dtm[2])<Day) state="0"; else if(Integer.parseInt(dtm[2])>Day) state="1"; else{ if(Integer.parseInt(tm[0])<Hour) state = "0"; else if(Integer.parseInt(tm[0])==Hour){ </pre>	

Tujuan Script	Keterangan
<pre> if(Integer.parseInt(tm[1])<Minute) state="0"; } } } } List_Data tmp = new List_Data(d.getString(1),d.getString(4),d.getString(2),d.g etString(3),i,d.getString(0),state,dump); alist.add(tmp); i+=1; }while(d.moveToNext()); isEmpty = false; list.setAdapter(new Content_Adapter(alist)); }else{ isEmpty = true; String [] icount = new String[]{"No Activity Match"}; ArrayAdapter<String> aa = new ArrayAdapter<String>(Search_Activity.this,android.R.layout .simple_list_item_1,icount); list.setAdapter(aa); } } </pre>	

Scene 04 : Add Activity

- a. Antarmuka



Gambar 4.6 Antarmuka Add Activity

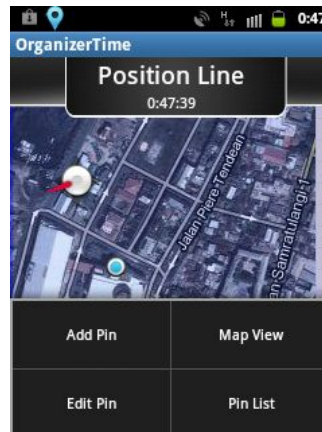
b. Coding

Tabel 4.7. Scene Add Activity

Tujuan Script	Keterangan
<p><i>Script untuk menampilkan Scene Add Activity</i></p> <pre> private void Save(){ if(CheckInput()){ //_id,e_type,priority,title,content,dt,tm,sound,air ,retry,reminder,point,user,status"); db.OpenDatabases(); String so = "1"; if(sound.getText().equals("OFF")) so = "0"; String ai = "0"; if(air.getText().equals("ON")) ai = "1"; String prior = getPriority(); String retry = getRetry(); String type = getType(); String remind = getReminder(); db.InsertData("todolist", new String[]{type,prior,title.getText().toString() ,content.getText().toString(),dtp,ttp,so,ai,retry,r emind,"1","-2"}, new String[]{"e_type" ,"priority","title","content","dt","tm","sound","ai r","retry","reminder","status","point"}); db.OpenDatabases(); String id = db.GetLastData("todolist", "_id", "", "_id ASC"); db.OpenDatabases(); if(sms>0) db.EditData("reciepent", new String[]{"todo"}, new String[]{id}, "todo=-1 AND tipe=1"); if(email>0) db.EditData("reciepent", new String[]{"todo"}, new String[]{id}, "todo=-1 AND tipe=2"); } } </pre>	

Scene 05: Position Line

a. Antarmuka



Gambar 4.7 Antarmuka *Position Line*

b. Coding

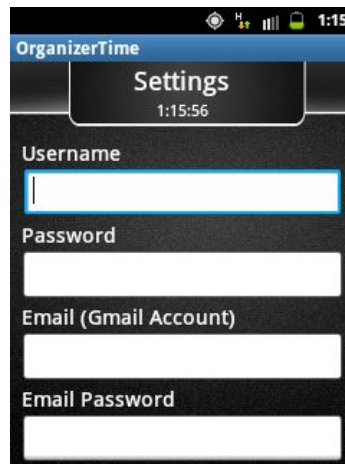
Tabel 4.8. Scene Action *Position Line*

Tujuan Script	Keterangan
<p><i>Script</i> untuk menampilkan Scene <i>Position Line</i></p> <pre> public void testProviders(){ Criteria criteria = new Criteria(); criteria.setAccuracy(Criteria.ACCURACY_FINE); criteria.setPowerRequirement(Criteria.POWER_LOW); criteria.setAltitudeRequired(false); criteria.setBearingRequired(false); criteria.setSpeedRequired(false); criteria.setCostAllowed(true); List<String>providers = locationManager.getProviders(criteria,false); for(String provider:providers){ locationManager.requestLocationUpdates(provider, 1000, 10, this); Location location = locationManager.getLastKnownLocation(provider); if(location!=null){ double lat = location.getLatitude(); double lng = location.getLongitude(); p = new GeoPoint((int)(lat * 1E6),(int)(lng * 1E6)); map.getController().animateTo(p); map.getController().setZoom(17); } } } </pre>	

Tujuan Script	Keterangan
<pre>map.setBuiltInZoomControls(true); } } }</pre>	

Scene 06: Setting

a. Antarmuka



Gambar 4.8 Antarmuka Setting

b. Coding

Tabel 4.9. Scene Setting

Tujuan Script	Keterangan
<p><i>Script untuk menampilkan Scene Setting</i></p> <pre>public void onClick(View v) { // TODO Auto-generated method stub if(v.getId()==R.id.btnsave){ db.OpenDatabases(); db.EditData("users", new String[]{"uid","pwd","email","pemail"}, new String[]{uid.getText().toString(), pwd.getText().toString(),email.getText().toString() ,epwd.getText().toString()},"status=1"); finish(); } }</pre>	

4.5 Pengujian Aplikasi

Pengujian aplikasi dilakukan seperti pada table berikut ini :

Tabel 4.10. Pengujian Aplikasi

Fungsi yang diuji	Hasil Harapan	Hasil Aktual (Keluaran)
Testing Menu Login		
Menekan tombol <i>Login</i>	Sistem menampilkan halaman <i>View Current Day</i>	Sistem menampilkan halaman <i>View Current Day</i>
Testing Menu Add Activity		
Memilih menu <i>add SMS</i>	Sistem menampilkan tampilan untuk mengisi nomor kontak	Sistem menampilkan tampilan <i>select contact</i> dari daftar kontak telepon
Memilih menu <i>add E-mail</i>	Sistem menampilkan tampilan untuk mengisi alamat <i>E-mail</i>	Sistem menampilkan tampilan <i>select E-mail Address</i> dari daftar kontak telepon
Testing Menu Setting		
Memilih menu <i>setting</i>	Menampilkan <i>field username, password, E-mail</i> dan <i>E-mail password</i>	Menampilkan <i>field username, password, E-mail</i> dan <i>E-mail password</i>
Testing Menu Position Line		
Memilih menu <i>position line</i>	Sistem menampilkan peta yang telah terhubung dengan GPS	Sistem menampilkan peta yang telah terhubung dengan GPS
Memilih <i>add pin</i>	Sistem menampilkan pin yang dapat digeser sesuai dengan lokasi yang diinginkan	Sistem menampilkan pin yang dapat digeser sesuai dengan lokasi yang diinginkan
Memilih <i>list pin</i>	Sistem menampilkan tampilan untuk mengisi	Sistem menampilkan tampilan untuk mengisi

Fungsi yang diuji	Hasil Harapan	Hasil Aktual (Keluaran)
	nama pin dan mengisi radius yang ingin ditetapkan	nama pin dan mengisi radius yang ingin ditetapkan
Cek GPS	Sistem mengecek apakah GPS dalam keadaan aktif atau tidak	Sistem mengecek apakah GPS dalam keadaan aktif atau tidak

Berdasarkan hasil pengujian, Aplikasi *Personal Assistant* pada Android dapat berjalan dengan baik dan juga memenuhi semua kebutuhan pengguna. Aplikasi ini mudah diaplikasikan pada telepon seluler berbasis android dengan versi *Froyo*. Fungsi-fungsi yang telah dibuat dapat berjalan dengan baik sesuai dengan yang diharapkan. Dengan demikian kebutuhan *user* dan sistem dapat terpenuhi dengan baik, tetapi masih diperlukan pengembangan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari keseluruhan kegiatan penelitian yang dilakukan ini, maka dapat ditarik kesimpulan, bahwa:

1. Aplikasi ini membantu dalam pengingatan agenda/kegiatan acara, dimana agenda tersebut dapat secara langsung dikirimkan melalui *sms/e-mail* sesuai waktu yang telah kita atur sendiri. Dapat dikatakan seperti asisten pribadi pada ponsel yang dapat melakukan apa yang telah kita atur dengan tepat waktu.
2. Aplikasi ini sangat membantu dalam pengingatan lokasi yang dituju dengan adanya *reminder* lokasi.

5.2 Saran

1. Menambahkan fitur pilihan nada dering pada menu *position line* agar pada saat mendekati radius lokasi yang ditetapkan, telepon seluler dapat berbunyi sesuai dengan nada dering yang dipilih *user*.
2. Menambahkan fitur pada *short messaging broadcast* untuk dapat langsung menginput nomor penerima.
3. Menambahkan *notification* jika GPS telepon seluler dalam keadaan tidak aktif.
4. Menambahkan pilihan *vibration* pada fitur *add activity* untuk memberikan *reminder*.

DAFTAR PUSTAKA

1. Hermawan, S. (2011). *Mudah Membuat Aplikasi Android*. Salatiga : Andi.
2. Jeffress, D. (2012). *How Do I Become a Personal Assistant*. Conjecture Corporation.
3. Lessard, J. And Kessler, G. (2011). *Android Forensics : Simplifying Cell Phone Examinations*. Small Scale Digital Device Forensics. Vol.4, no.1, pp 1.
4. Prabowo, P. (2011). *Rancang Bangun Aplikasi Pemantau Penyelewengan Kendaraan Dinas Dengan Menggunakan Modul GIS*. Yogyakarta : Sekolah Tinggi Manajemen Informatika & Komputer AMIKOM.
5. Priyanta, F. (2011). *Pemrograman Android Untuk Pemula*. Surabaya : Cerdas Pustaka Publisher.
6. Rosa, A.S. and Shalahuddin, M. (2011). *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur & Berorientasi Objek)*. Bandung : Modula.
7. Safaat, N. (2011). *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung : Informatika.
8. Supardi, Y. (2011). *Semua Bisa Menjadi Programmer Android Basic*. Jakarta : Elek Media Komputindo.
9. Winarno, E. and Zaki, A. (2011). *Membuat Sendiri Aplikasi Android Untuk Pemula*. Semarang : Elex Media Komputindo.