

Mobile Programming

STMIK Nusa Mandiri

Ridwansyah, M.Kom



KATA PENGANTAR

Alhamdulillah saya ucapkan untuk kedua kali dapat menyelesaikan dan membuat sebuah modul.

Saya ingin juga mengucapkan terima kasih kepada : Allah SWT yang telah menganugerahkan akal dan pikiran ini, serta kekuatan fisik saya, Kedua Orang tua dan Kakak dan adik saya yang telah memberikan dukungannya baik moral dan spiritual, Istri dan anak saya serta Seluruh staff dan karyawan STMIK Nusa Mandiri, Dosen-dosen pengajar Mobile Programming atas subangsihnya pada diri saya. Besar harapan saya agar para mahasiswa dapat mudah untuk mempelajari Mobile Programming. Dan mudahan-mudahan buku ini dapat berguna bagi para pembaca dan pengguna sekalian.

Penulis

Ridwansyah, M.Kom

DAFTAR ISI

Kata Pengantar	1
Daftar Isi	2
BAB I : Pengenalan Android	
1.1 Pengenalan Android.....	3
1.2 Sejarah Android	3
1.3 Kelebihan dan Kekurangan Android.....	4
1.4 Aplikasi Untuk Android.....	5
BAB II : Database	
2.1 Database Android	7
2.2 Bekerja dengan Database Android.....	7
2.3 Android Sqlite Database.....	9
2.4 Membuat Database.....	12
BAB III : Program Sederhana Android	
3.1 GPS	22
3.2 Game Sederhana	29
BAB IV : Membangun Aplikasi Android	
4.1 Membangun Aplikasi	31

PENGENALAN ANDROID

1.1. Apakah Android itu?

Android merupakan sistem operasi yang berbasis Linux untuk telepon seluler, dengan menyediakan platform bersifat *open source* bagi para pengembang untuk menciptakan aplikasi yang mereka inginkan. Awalnya, Google Inc. mengakuisi Android Inc., pendatang baru yang membuat *software* untuk ponsel yang berada di Palo Alto, California Amerika Serikat. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan *hardware*, *software*, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

1.2. Sejarah Android

Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD).

Para pendiri Android Inc. bekerja pada Google, di antaranya Andy Rubi, Rich Miner, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android Inc. hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak

memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler.

1.3.Kelebihan dan Kekurangan Android

Berdasarkan pengembangannya, yang membedakan platform android dengan Platform lainnya dan merupakan kelebihan Android adalah :

1. Open Source

Android menyediakan akses ke fungsi dasar perangkat mobile menggunakan standard panggilan ke API.

2. Konektifitas Tanpa Batasan

Dapat menggabungkan informasi dari Internet ke dalam telepon, seperti informasi kontak, atau data pada lokasi geografis untuk mendapatkan kesempatan baru.

3. Sederhana

Dalam SDK yang diterapkan oleh Android memiliki semua yang di butuhkan untuk membuat dan menjalankan aplikasi Android, termasuk simulator instrumen, dan alat debugging.

4. Object Oriented

Android menggunakan bahasa pemrograman yang berorientasi objek yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali. Pemrograman berorientasi objek memodelkan dunia nyata kedalam objek dan melakukan interaksi antar objek-objek tersebut.

5. Stabilitas

Android lebih stabil, dengan banyak aplikasi terinstall di dalamnya.

Berikut ini fitur-fitur yang dimiliki oleh Android :

1. Application framework yang komponen-komponennya dapat diganti dan diubah ulang.
2. Dalvik virtual machine untuk mengoptimalkan mobile device.
3. Integrated browser based on the open source WebKit engine.
4. Optimized graphics support a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification (hardware acceleration optional).
5. SQLite sebagai penyimpanan struktur data.
6. Dukungan media secara umum mendukung format audio, video dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
7. GSM Telephony (hardware dependent).
8. Media koneksi melalui bluetooth, EDGE, 3G dan WiFi.
9. Camera, GPS, compass, and accelerometer.
10. Lingkungan development yang mencakup perangkat emulator, tools untuk debug, memory dan performance profiling serta plugin untuk Eclipse IDE.

1.4. Aplikasi Untuk Android

Untuk membangun aplikasi Android, seorang programmer dituntut telah akrab dengan bahasa pemrograman Java. Bila syarat pertama ini telah dipenuhi, langkah selanjutnya adalah men-download software development kit (SDK) yang disediakan Android. SDK ini

membukakan jalan bagi programmer untuk mengakses *application programming interface* (API) Android.

SDK yang diinstall di PC ini mencakup beberapa tool, termasuk aplikasi contoh dan emulator Android. Emulator Android adalah program yang menduplikasi fungsi-fungsi smartphone yang berjalan di atas platform Android. Emulator juga berfungsi sebagai ajang pengujian aplikasi di PC, sebelum disematkan ke dalam smartphone Android.

Google menjabarkan aplikasi Android ke dalam empat blok bangunan dasar (tidak semua aplikasi mempunyai keempatnya) yaitu :

- *Activities*

yaitu ketika sebuah aplikasi memunculkan screen di layar. Sebagai contoh, sebuah aplikasi GPS mempunyai screen peta dasar, screen rencana perjalanan, dan screen rute di atasnya. Ketiga penampakan screen ini disebut activities.

- *Intents*

yaitu mekanisme perpindahan dari suatu activity ke activity lainnya. Sebagai contoh ketika merencanakan perjalanan pada aplikasi GPS, intent akan menginterpretasi input dan mengaktifkan screen rute di atas screen peta.

- *Services*

yaitu serupa dengan service di PC dan server, program yang berjalan di belakang layar, tanpa interferensi dari pengguna.

- *Content Provider*

yaitu mekanisme yang memungkinkan sebuah aplikasi berbagi informasi dengan aplikasi lainnya.

DATABASE ANDROID

2.1. Database Android

Android mengembangkan full relational database dengan kemampuan berhubungan dengan *SQL Library*. Dengan menggunakan *SQLite* dapat membuat *relational database* tersendiri untuk setiap aplikasi., Gunakan itu untuk menyimpan dan mengatur secara kompleks struktur data aplikasi. Seluruh *database* Android disimpan pada `/data/data/<nama package>/database` folder pada *device*-nya (emulator). Secara *default* semua *database* adalah *private*, dan dapat diakses hanya dengan aplikasi yang membuat *database* tersebut. Untuk membagi *database* diluar aplikasi gunakan *Content Provider*.

2.2. Bekerja dengan Database Android

Memikirkan pembuatan *database adapter*, bagian yang mana dari sebuah abstrak layer yang merupakan enkapsulasi integrasi database. Ini seharusnya memberikan sebuah intuisi, tipe method yang kuat untuk menambah, menghapus, dan merubah item. Sebuah database adapter seharusnya juga menangani query dan membuat wrap, membuka wrap, dan menutup wrap.

Ini juga sering kali digunakan sebagai lokasi yang sesuai dengan dari tempat dimana melakukan statik *publish* pada *database* konstan. Dengan menyertakan nama tabel, nama column, dan nama indek.

Berikut kode program yang digunakan pada Android database:

```
import android.content.Context;
import android.database.*;
import android.database.sqlite.*;
import android.database.sqlite.SQLiteDatabase.CursorFactory;
import android.util.Log;
public class MyDBAdapter {
private static final String DATABASE_NAME = "myDatabase.db";
```



```

private static final String DATABASE_TABLE = "mainTable";
private static final int DATABASE_VERSION = 1;
// The index (key) column name for use in where clauses.
public static final String KEY_ID="_id";
// The name and column index of each column in your database.
public static final String KEY_NAME="name";
public static final int NAME_COLUMN = 1;
// TODO: Create public field for each column in your table.
// SQL Statement to create a new database.
private static final String DATABASE_CREATE = "create table " +
DATABASE_TABLE + " (" + KEY_ID +
" integer primary key autoincrement, " +
KEY_NAME + " text not null);";
// Variable to hold the database instance
private SQLiteDatabase db;
// Context of the application using the database.
private final Context context;
// Database open/upgrade helper
private myDbHelper dbHelper;
public MyDBAdapter(Context _context) {
context = _context;
dbHelper = new myDbHelper(context, DATABASE_NAME, null,
DATABASE_VERSION);
}
public MyDBAdapter open() throws SQLException {
db = dbHelper.getWritableDatabase();
return this;
}
public void close() {
db.close();
}
public long insertEntry(MyObject _myObject) {
ContentValues contentValues = new ContentValues();
// TODO fill in ContentValues to represent the new row
return db.insert(DATABASE_TABLE, null, contentValues);
}
public boolean removeEntry(long _rowIndex) {
return db.delete(DATABASE_TABLE, KEY_ID +

```

```

"=" + _rowIndex, null) > 0;
}

public Cursor getAllEntries () {
return db.query(DATABASE_TABLE, new String[] {KEY_ID, KEY_NAME},
null, null, null, null, null);
}

public MyObject getEntry(long _rowIndex) {
MyObject objectInstance = new MyObject();
// TODO Return a cursor to a row from the database and
// use the values to populate an instance of MyObject
return objectInstance;
}

public int updateEntry(long _rowIndex, MyObject _myObject) {
String where = KEY_ID + "=" + _rowIndex;
ContentValues contentValues = new ContentValues();
// TODO fill in the ContentValues based on the new object
return db.update(DATABASE_TABLE, contentValues, where, null);
}

private static class myDbHelper extends SQLiteOpenHelper {
public myDbHelper(Context context, String name,
CursorFactory factory, int version) {
super(context, name, factory, version);
}

// Called when no database exists in
// disk and the helper class needs
// to create a new one.
@Override
public void onCreate(SQLiteDatabase _db) {
_db.execSQL(DATABASE_CREATE);
}

// Called when there is a database version mismatch meaning that
// the version of the database on disk needs to be upgraded to
// the current version.
@Override
public void onUpgrade(SQLiteDatabase _db, int _oldVersion,
int _newVersion) {
// Log the version upgrade.
Log.w("TaskDBAdapter", "Upgrading from version " +

```

```

_oldVersion + " to " +
_newVersion +
", which will destroy all old data");
// Upgrade the existing database to conform to the new version.
// Multiple previous versions can be handled by comparing
// _oldVersion and _newVersion values.
// The simplest case is to drop the old table and create a
// new one.
_db.execSQL("DROP TABLE IF EXISTS " + DATABASE_TABLE);
// Create a new one.
onCreate(_db);
}
}
}

```

2.3. Android Sqlite Database

SQLite adalah database yang bisa dibangun di android. SQLite memiliki fitur relasional database, hampir sama dengan SQL pada desktop hanya saja SQLite membutuhkan memori yang sedikit. SQLite terdapat pada semua perangkat android, Kamu cukup mendefinisikan perintah SQL untuk meng-create atau meng-update database, selanjutnya system pada android akan menangani hal-hal yang berhubungan dengan database. SQLite database otomatis akan tersimpan didalam path data/data/nama_package/database/nama_database.

Sebelum mulai membuat aplikasi SQLite database, ada 3 kelas yang harus Kamu ketahui yaitu:

1. SQLiteDatabase

SQLiteDatabase merupakan kelas yang mempunyai method seperti:

- a. Insert() untuk menambahkan baris ke database
- b. Update() untuk memperbarui baris pada database
- c. Delete() untuk menghapus baris pada database

- d. `execSQL()` untuk mengeksekusi sintak SQL

2. SQLiteOpenHelper

SQLiteOpenHelper adalah subclass yang memiliki beberapa

- a. `onCreate()` dijalankan jika sebelumnya belum ada database
- b. `onUpgrade()` dijalankan jika sebelumnya sudah ditemukan database yang sama namun beda versi.
- c. Method ini bisa dimanfaatkan untuk mengubah skema database
- d. `onOpen()` dijalankan jika database sudah dalam keadaan open method seperti:
 - `getWritableDatabase()` memanggil database agar bisa dimasuki data
 - `getReadableDatabase()` memanggil database agar bisa membaca datanya

3. Cursor

Setiap query yang dieksekusi pasti membawa nilai kembalian atau feedback. Feedback yang dihasilkan query ini disebut cursor. Jadi dengan kata lain, cursor merepresentasikan hasil query yang dieksekusi pada baris dan kolom tertentu. Gambar 12.1 berikut ini akan membantu Kamu memahami istilah cursor pada SQLite.

Tabel. II.1. Analogi cursor pada SQLite Database

Nama	Hobi
Ayib	Makan
Fitri	Masak
Ridhani	Baca

Cursor diwakili dengan tKamu panah berada pada posisi baris ke 2 dengan membawa data [fitri,masak]. Berikut perintah dibawah ini

- moveToFirst() untuk pindah ke baris pertama isAfterLast() akan mengirim pesan balik jika posisi cursor sudah berada di baris terakhir
- getLong() untuk mengambil data pada kolom yang mempunyai tipe data long
- getString() untuk mengambil data pada kolom yang mempunyai tipe data String

2.4. Membuat Database

Kita akan membuat database hobi seseorang. Data dimasukkan melalui 2 buah edittext, kemudian tombol add dipakai untuk menyimpan data kedalam database sekaligus mengupdate tabel jika data berhasil disimpan.



Gambar II.1. Database Android

1. Siapkan project baru dengan nama sebagai berikut

Tabel II.2. Keterangan Project

Project Name	DatabaseAndroid
Build Target	Android 2.2
Application Name	Database Android #1
Package Name	Com.db.satu
Create Activity	DatabaseAndroidSatu
Min SDK Version	8

Ubah baigan **String.xml** seperti berikut

```
<?xml version="1.0" encoding="utf-8"?>

<resources>

<string name="hello">Masukkan nama dan hobi</string>

<string name="app_name">Database Android #1</string>

<string name="btnAddtxt">Add</string>
```

```
<string name="namaLabel">Nama</string>

<string name="hobiLabel">hobi</string>

<string name="nomorLabel">No.</string>

</resources>
```

Ubah **main.xml** seperti berikut ini

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"

android:layout_width="fill_parent"

android:layout_height="fill_parent">

<TextView android:layout_width="fill_parent"

android:layout_height="wrap_content"

android:text="@string/hello" />

<LinearLayout android:layout_width="match_parent"

android:layout_height="wrap_content"

android:id="@+id/linearLayout1">

<EditText android:id="@+id/inNama"

android:layout_height="wrap_content"

android:layout_width="100dip"></EditText>

<EditText android:id="@+id/inHobi"

android:layout_height="wrap_content"

android:layout_width="100dip"></EditText>
```

```

<Button android:layout_width="wrap_content"
android:id="@+id/btnAdd"
android:layout_height="wrap_content"
android:text="@string/btnAddtxt"></Button>

</LinearLayout>

<TableLayout android:layout_height="wrap_content"
android:layout_width="match_parent"
android:id="@+id/tabel_data">

<TableRow android:id="@+id/tableRow1"
android:layout_width="wrap_content"
android:layout_height="wrap_content">

<TextView android:layout_height="wrap_content"
android:layout_width="50dip"
android:text="@string/nomorLabel"
android:id="@+id/no_id"></TextView>

<TextView android:layout_height="wrap_content"
android:layout_width="100dip"
android:text="@string/namaLabel"
android:id="@+id/nama_id"></TextView>

<TextView android:layout_width="100dip"
android:layout_height="wrap_content"
android:text="@string/hobiLabel"
android:id="@+id/hobi_id"></TextView>

```

```
</TableRow>

</TableLayout>

</LinearLayout>
```

Buat kelas baru kemudian beri nama **DatabaseManager.java**, setelah itu ketikkan baris kode berikut ini.

```
1: package com.db.satu;
2:
3: import java.util.ArrayList;
4:
5: import android.content.ContentValues;
6:
7: import android.content.Context;
8:
9: import android.database.Cursor;
10:
11: import android.database.sqlite.SQLiteDatabase;
12:
13: import android.database.sqlite.SQLiteOpenHelper;
14:
15: import android.util.Log;
16:
17: 11:
18: 12: public class DatabaseManager
19: { 13:
20:     private static final String ROW_ID = "_id";
21:
22:     private static final String ROW_NAMA = "nama";
23:
24:     private static final String ROW_HOBI = "hobi";
25:
26:     private static final String NAMA_DB = "DatabaseAndroidSatu";
27:
28:     private static final String NAMA_TABEL = "hobiku";
29:
30:     private static final int DB_VERSION = 1;
31:
32: 21:
33: 22: private static final String CREATE_TABLE = "create table
34: "+NAMA_TABEL+" (" +ROW_ID+" integer PRIMARY KEY autoincrement,
35: "+ROW_NAMA+" text,"+ROW_HOBI+" text)";
36:
37: 23:
```



```

private final Context context;

private DatabaseOpenHelper dbHelper;

private SQLiteDatabase db;
27:
public DatabaseManager(Context ctx) {

    this.context = ctx;

    dbHelper = new DatabaseOpenHelper(context);

    db = dbHelper.getWritableDatabase();
    }
33:
34: private static class DatabaseOpenHelper extends
35: SQLiteOpenHelper {
    public DatabaseOpenHelper(Context context) {
37:
    super(context, NAMA_DB, null, DB_VERSION);
38:
    // TODO Auto-generated constructor stub
39: }
40:
    @Override

    public void onCreate(SQLiteDatabase db) {
    // TODO Auto-generated method stub
44:
    db.execSQL(CREATE_TABLE);
45: }
46:
    @Override

    public void onUpgrade(SQLiteDatabase db, int oldVer, int
newVer) {
49:
    // TODO Auto-generated method stub
50:
    db.execSQL("DROP TABLE IF EXISTS "+NAMA_DB);
51: onCreate(db);
52:
    }
    }

```

```

public void close() {
    dbHelper.close();
}
58:
public void addRow(String nama, String hobi) {
    ContentValues values = new ContentValues();
    values.put(ROW_NAMA, nama);
    values.put(ROW_HOBI, hobi);
    try {
        db.insert(NAMA_TABEL, null, values);
    } catch (Exception e) {
        Log.e("DB ERROR", e.toString());
        e.printStackTrace();
    }
}

public ArrayList<ArrayList<Object>> ambilSemuaBaris() {
    ArrayList<ArrayList<Object>> dataArray =
new ArrayList<ArrayList<Object>>();
    Cursor cur;
    try {
        cur = db.query(NAMA_TABEL,
            new String[] { ROW_ID, ROW_NAMA, ROW_HOBI }, null, null,
            null, null, null);
        cur.moveToFirst();
        if (!cur.isAfterLast()) {
            do {
                ArrayList<Object> dataList = new ArrayList<Object>();
                dataList.add(cur.getLong(0));
                dataList.add(cur.getString(1));
                dataList.add(cur.getString(2));
                dataArray.add(dataList);
            } while (cur.moveToNext());
        }
        catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            Log.e("DEBE ERROR", e.toString());
        }
        return dataArray;
    }
}

```

Nah sekarang kita menambahkan kode di activity **DatabaseAndroidSatu.java** seperti dibawah ini.

```
1: package com.db.satu;
2:
3: import java.util.ArrayList;
4:
import android.app.Activity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TableLayout;

import android.widget.TableRow;

import android.widget.TextView;

import android.widget.Toast;


public class DatabaseAndroidSatu extends Activity {

    DatabaseManager dm;

    EditText nama, hobi;

    Button addBtn;

    TableLayout tabel4data;// tabel for data

    /** Called when the activity is first created. */

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
```

```

setContentView(R.layout.main);
26:
dm = new DatabaseManager(this);

tabel4data = (TableLayout) findViewById(R.id.tabel_data);

nama = (EditText) findViewById(R.id.inNama);

hobi = (EditText) findViewById(R.id.inHobi);

addBtn = (Button) findViewById(R.id.btnAdd);

addBtn.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {
        35: simpKamuta();

    });

    updateTable();
}
40:
protected void simpKamuta() {

    try {

        dm.addRow(nama.getText().toString(),hobi.getText().toString());

        Toast.makeText(getApplicationContext(),

            nama.getText().toString() + ", berhasil disimpan",

            Toast.LENGTH_SHORT).show();

        updateTable();

        kosongkanField();

    } catch (Exception e) {

        e.printStackTrace();

        Toast.makeText(getApplicationContext(), "gagal simpan, " +

```

```

e.toString(),Toast.LENGTH_LONG).show();
}
}
protected void kosongkanField(){

nama.setText("");

hobi.setText("");
}

protected void updateTable() {

// TODO Auto-generated method stub

while (tabel4data.getChildCount() > 1) {

tabel4data.removeViewAt(1);

}

63:
64: ArrayList<ArrayList<Object>> data = dm.ambilSemuaBaris();//
65:
for (int posisi = 0; posisi < data.size(); posisi++) {

TableRow tabelBaris = new TableRow(this);

        ArrayList<Object> baris = data.get(posisi);

TextView idTxt = new TextView(this);

idTxt.setText(baris.get(0).toString());

tabelBaris.addView(idTxt);

73:
TextView namaTxt = new TextView(this);

namaTxt.setText(baris.get(1).toString());

tabelBaris.addView(namaTxt);
77:
TextView hobiTxt = new TextView(this);

hobiTxt.setText(baris.get(2).toString());

```

```
tabelBaris.addView(hobiTxt);  
81:  
tabel4data.addView(tabelBaris);  
}  
}  
}
```

PROGRAM SEDERHANA ANDROID

3.1. GPS

Trend teknologi sekarang dan kedepan salah satunya adalah teknologi mobile dengan fitur LBS nya. Dengan LBS, kita bisa mencari suatu lokasi dengan sangat mudah, mencari posisi teman, mempublish lokasi keberadaan kita sendiri, mencari rute jalan atau lainnya. Suatu gadget katakanlah Android mendapatkan lokasi bisa dari GPS , Network Location Provider atau bisa juga keduanya. Cell-id (Simcard) dan wifi adalah contoh Network Location Provider.

Menentukan lokasi menggunakan Android itu gampang-gampang susah, karena semakin akurat suatu lokasi semakin boros juga baterainya. Berikut ini adalah hal-hal yang menjadi kendala untuk menentukan lokasi

1. Multitude sumber lokasi

GPS, simcard dan wifi dapat dikombinasikan untuk mendapatkan lokasi yang akurat, namun efeknya jatuh ke baterai.

2. Perpindahan pengguna

Karena seorang pengguna melakukan perpindahan, maka aplikasi juga akan mererefresh lokasi secara berulang.

3. Akurasi

Akurasi terhadap posisi suatu lokasi tidak konsisten. Posisi 10 menit yang lalu bisa jadi lebih akurat dibandingkan posisi yang terbaru. Baik, agar cepat menjadi jago android, sekarang saatnya membuat aplikasi menentukan Latitude dan Longitude. Skenarionya kita bikin aplikasi yang bisa menangkap perubahan nilai longitude dan latitude. Setiap terjadi perubahan, nilai

latitude dan longitude yang baru ditampilkan melalui Toast dan TextView. Kita bisa mengirim lokasi ke emulator menggunakan DDMS. DDMS dibahas lebih dalam dibagian xx. Hasil akhirnya tampak seperti gambar 14.1.

1. Buat project baru

Tabel II.3 Keterangan Project

Project Name	Pengenalan GPS
Build Target	Android 2.2
Application Name	Pengenalan GPS
Package Name	Com.gps.sederhana
Create Activity	Lokasiku
Min SDK Version	8

Tambahkan beberapa baris kode pada Strings.xml

```
<?xml version="1.0" encoding="utf-8"?>

<resources>

<string name="hello">Lokasiku!</string>

<string name="app_name">pengenalan GPS</string>

<string name="latText">Latitude :</string>

<string name="lonText">Longitude :</string>

</resources>
```

Buat tampilan main.xml seperti berikut

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout_width="fill_parent"

android:layout_height="fill_parent">
```



```

<TextView android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textStyle="bold"
    android:text="@string/lonText"></TextView>

<TextView android:text="unknown"
    android:id="@+id/longituddeTxt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"></TextView>

<TextView android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textStyle="bold"
    android:text="@string/latText"></TextView>

<TextView android:text="unknown"
    android:id="@+id/latitudeTxt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"></TextView>

</LinearLayout>
26:

```

Sekarang kita ke activity Lokasiku.java, tambahkan kode berikut

```

package com.gps.sederhana;
2:
import android.app.Activity;
import android.content.Context;
import android.location.Location;

```

```

import android.location.LocationListener;

import android.location.LocationManager;

import android.os.Bundle;

import android.widget.TextView;

import android.widget.Toast;
11:
public class Lokasiku extends Activity {

    private LocationManager lm;

    private LocationListener locListener;

    private TextView latTxt,lonTxt;
16:
    /** Called when the activity is first created. */

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.main);
22:
        latTxt = (TextView) findViewById(R.id.latitudeTxt);

        lonTxt = (TextView) findViewById(R.id.longitudeTxt);

        lm = (LocationManager)

            getSystemService(Context.LOCATION_SERVICE);

        locListener = new MyLocationListener();

        lm.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0,

            200, locListener);
    }
31:
    private class MyLocationListener implements LocationListener

```

```

{

@Override

public void onLocationChanged(Location loc) {

if (loc != null) {

latTxt.setText(String.valueOf(loc.getLatitude()));

lonTxt.setText(String.valueOf(loc.getLongitude()));

Toast.makeText(getApplicationContext(),

"Location Changed : Lat " + loc.getLatitude() +

"lgt: "+loc.getLongitude(), Toast.LENGTH_SHORT).show();

}

}

@Override

public void onProviderDisabled(String arg0) {}

@Override

public void onProviderEnabled(String arg0) {}

@Override

public void onStatusChanged(String provider, int status,

Bundle extras){}

}

}

```

Terakhir, tambahkan user.permission untuk mengakses GPS ke dalam Manifest.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
package="com.gps.sederhana" android:versionCode="1"
android:versionName="1.0">

<application android:icon="@drawable/icon"
android:label="@string/app_name">

<activity android:name=".Lokasiku"
android:label="@string/app_name">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category
android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

<uses-sdk android:minSdkVersion="8" />

<uses-permission

android:name="android.permission.ACCESS_FINE_LOCATION">

</uses-permission>

<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION">
</uses-permission>

<uses-permission
android:name="android.permission.INTERNET">
</uses-permission>

</manifest>
```

3.2. Game Sederhana

1. Game 2D

Pada pembahasan kali akan membahas tentang pembuatan game 2D. Tetapi dalam pembuatan game ini. Kita menggunakan engine Candroid (<http://code.google.com/p/candroidengine/>) untuk mempermudah dalam proses pembuatan game tersebut.

2. Basic, Texture and Sprites.

Sebelum membuat Sprite, diharuskan untuk membuat sebuah Texture dari sebuah gambar kemudian di renderer dengan SurfaceRenderer(). Class yang kita gunakan pun merupakan turunan dari CandroidSurfaceView.

```
//Membuat Renderer baru untuk menggambar game
spriteRenderer = new SurfaceRenderer();

//TextureAtlas seperti sebuah container untuk menambah texture
//yang dibutuhkan.
TextureAtlas atlas = new TextureAtlas();

//Texture adalah class yang paling penting.Parameter yang dibutuhkan
//adalah lokasi dari gambar yang terletak di dalam folder assets
candroidTexture = new Texture("graphics/candroid_logo.png");
seamolec = new Texture("graphics/seamolec.png");
jeni = new Texture("graphics/jeni.png");

//Menambahkan semua texture ke dalam TextureAtlas
atlas.addTexture(candroidTexture);
atlas.addTexture(seamolec);
```

```

atlas.addTexture(jeni);

//Kemudian gunakan method static load() untuk memuat semua Texture
TextureManager.load(context, atlas);

//Sprites adalah sebuah object dari gambar
candroidSprite = new Sprite(candroidTexture,100,100);
seamolecSprite = new Sprite(seamolec,20,20);
jenisprite = new Sprite(jeni,100,250);

//Menambahkan sprite pada Renderer.
spriteRenderer.addRenderable(candroidSprite);
spriteRenderer.addRenderable(seamolecSprite);
spriteRenderer.addRenderable(jenisprite);

setRendererAndStart(spriteRenderer);

```

3. Animations

Untuk membuat sebuah animasi di dalam game,digunakan TileAnimation. TileAnimation adalah animasi untuk game.

```

TileAnimation ta = new TileAnimation(Texture tex, int x, int y, int tileColumns, int tileRows,
int frameskipdelay);

```

Parameter yang digunakan adalah texture , kordinat x , kordinat y , jumlah kolom dari gambar yang akan dipotong , jumlah baris dari gambar yang akan dipotong , jumlah waktu yang digunakan untuk pindah ke frame berikutnya.

```

explosionTex = new Texture("graphics/explosion.png");

```

```
marioTex = new Texture("graphics/mario_sprite.png");
atlas.addTexture(explosionTex);
atlas.addTexture(marioTex);
TextureManager.load(context, atlas);

explosionAni = new TileAnimation(explosionTex, 2, 2, 5, 5, 100);
marioSprite = new Sprite(marioTex, 10, 90);
//Menambahkan ke dalam sprite (8 kolom dan 1 baris)
marioAni = new TileAnimation(marioTex, 10, 150, 8, 1, 100);
//Menambahkan ke dalam Renderer
spriteRenderer.addRenderable(explosionAni);
spriteRenderer.addRenderable(marioSprite);
spriteRenderer.addRenderable(marioAni);
```

MEMBANGUN APLIKASI ANDROID

4.1. Membangun Aplikasi

1. Aplikasi Diskon

Pembahasan materi sekarang hari ini kita belajar merancang sebuah aplikasi yang berguna bagi orang umum. Dalam mengembangkan aplikasi, kita berangkat dari masalah sederhana dalam kehidupan sehari-hari. Contohnya tidak semua orang bisa menghitung harga setelah kena diskon dengan cepat. Nah, kita akan membuat aplikasi untuk membantu orang dalam menghitung dengan cepat. Inilah ide sederhana untuk membuat aplikasi DISKON!!.

Skenarionya apabila aplikasi di-launch, disediakan 2 buah editteks untuk memasukkan harga awal dan besar diskon. Kemudian sebuah tombol, jika diklik maka muncul nilai berapa harga setelah kena diskon. Tuliskan resources di Strings.xml seperti berikut

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<string name="hello">HargaDiskon!</string>
<string name="app_name">Diskon!</string>
<color name="trans">#00330000</color>
<drawable name="klik">#CCFFFFFF</drawable>
</resources>
```

Sekarang buat di main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView android:id="@+id/scrollView1"
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="wrap_content"
```



```

android:layout_height="wrap_content">
<TableLayout android:orientation="vertical"
android:layout_height="wrap_content"
android:layout_width="fill_parent">
<TableRow android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:id="@+id/tableRow1"
android:layout_weight="1">
<TextView android:id="@+id/textView1"
14:
android:layout_width="fill_parent"
15: android:layout_height="wrap_content"
16:
android:text="Harga Asli Rp."></TextView>
17: <EditText android:text=" "
18:
android:layout_height="wrap_content"
19: android:id="@+id/harga"
20:
android:inputType="numberDecimal"
21: android:layout_width="40dip"></EditText>
<TextView android:text=",00"
23:

```

```

android:id="@+id/textView2"

24: android:layout_width="wrap_content"

android:layout_height="wrap_content"></TextView>

</TableRow>

<TableRow android:layout_width="fill_parent"

android:layout_height="wrap_content"

android:id="@+id/tableRow1"

android:layout_weight="1">

<TextView android:id="@+id/textView1"

32: android:layout_width="fill_parent"

33:

android:layout_height="wrap_content"

34: android:text="Besar Diskon "></TextView>

35:

<EditText android:text=" "

36: android:layout_height="wrap_content"

37:

android:id="@+id/diskon"

38: android:inputType="numberDecimal"

39:

android:layout_width="50dip"></EditText>

<TextView android:text="%"

41:

```

```

android:id="@+id/textView2"
42: android:layout_width="wrap_content"
    android:layout_height="wrap_content"></TextView>
</TableRow>
<View android:background="#FF909090"
    android:layout_width="wrap_content"
    android:id="@+id/view1"
    android:layout_height="2dip"></View>
<TextView android:id="@+id/tekshitung"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:gravity="center_horizontal"
    android:text=".: Hitung .:"></TextView>
<View android:background="#FF909090"
    android:layout_width="wrap_content"
    android:id="@+id/view1"
    android:layout_height="2dip"></View>
<TableRow android:id="@+id/tableRow2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
    <ImageButton android:layout_width="wrap_content"
62: android:layout_height="wrap_content"
63: android:src="@drawable/diskon"

```

```

64: android:id="@+id/imageButton1"
65:
    android:background="@color/trans"
66: android:paddingTop="10dip"
67:
    android:paddingBottom="10dip"></ImageButton>
    <Button android:text="Reset"
69:
    android:id="@+id/reset"
70: android:layout_marginLeft="50dip"
71:
    android:layout_width="wrap_content"
72: android:layout_marginTop="7dip"
73:
    android:gravity="center_vertical/left"
74: android:tag="reset all"
75:
    android:layout_gravity="left/center_vertical"
    android:layout_height="wrap_content"></Button>
</TableRow>
<View android:background="#FF909090"
    android:layout_width="wrap_content"
    android:id="@+id/view1"

```

```

    android:layout_height="2dip"></View>
82:
    <TableRow android:id="@+id/tableRow2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
        <TextView android:id="@+id/textView1"
87: android:layout_width="fill_parent"
88:
        android:layout_height="wrap_content"
89: android:text="Harga sekarang : "
90:
        android:maxLines="2"
91: android:textStyle="bold"
92:
        android:textSize="15dip"></TextView>
        <TextView android:text=""
94:
        android:layout_height="wrap_content"
95: android:id="@+id/hasil"
96:
        android:inputType="numberDecimal"
97: android:textStyle="bold"
98:

```

```

android:textSize="20dip"

android:layout_width="wrap_content"></TextView>

</TableRow>

<TextView android:layout_width="fill_parent"
102:
android:layout_height="wrap_content"
103: android:maxLines="2"
104:
android:textStyle="bold"
105:
android:textSize="15dip"
106: android:id="@+id/hemat"
107:
android:gravity="center_horizontal"
108: android:paddingTop="10dip"></TextView>

</TableLayout>

</ScrollView>

```

Buka activity kemudian tulis kode berikut ini

```

1: package com.diskon;
2:
3: import java.text.DecimalFormat;
4:
5: import android.app.Activity;

```

```
6: import android.app.AlertDialog;

7:

import android.content.DialogInterface;

8: import android.os.Bundle;

9:

import android.text.Editable;

10: import android.view.Menu;

11: import android.view.MenuItem;

12: import android.view.View;

13: import android.view.View.OnClickListener;

14: import android.widget.Button;

15: import android.widget.EditText;

import android.widget.ImageButton;

import android.widget.TextView;

18:

public class hargaDiskon extends Activity {

    EditText harga, diskon;

    TextView hargabarur, hmt;

    Editable isiharga, isidiskon;

    Button reset;

    ImageButton htng;

    String sharga, sdiskon;
```

26:

```
/** Called when the activity is first created. */
```

```
@Override
```

```
public void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
```

```
setContentView(R.layout.main);
```

32:

```
hargabaru = (TextView) findViewById(R.id.hasil);
```

```
harga = (EditText) findViewById(R.id.harga);
```

```
diskon = (EditText) findViewById(R.id.diskon);
```

```
hmt = (TextView) findViewById(R.id.hemat);
```

```
reset = (Button) findViewById(R.id.reset);
```

```
reset.setOnClickListener(new reset());
```

```
htng = (ImageButton) findViewById(R.id.imageButton1);
```

```
htng.setOnClickListener(new itung());
```

41:

42: }

43:

```
private class itung implements OnClickListener {
```

```
public void onClick(View v) {
```

```
try {
```

```
Double h =
```

```
Double.parseDouble(harga.getText().toString());
```



```

Double d =
Double.parseDouble(diskon.getText().toString());
Double nd = (d / 100) * h;
double hsl = h - nd;
Number irit = h - hsl;
DecimalFormat df = new DecimalFormat("@ @##");
hargabaru.setText("Rp." + df.format(hsl) + ",00");
hmt.setText("Wow!! Kamu Hemat Rp."
57: + df.format(irit) + " !!");
} catch (Exception e) {
};
}
}
62:
public class reset implements OnClickListener {
public void onClick(View v) {
65: hargabaru.setText("");
66: harga.setText("");
67: diskon.setText("");
68: hmt.setText("");
}
}
71:

```

```

public boolean onCreateOptionsMenu(Menu menu) {

super.onCreateOptionsMenu(menu);

menu.add(0, 0, 0, "Tutorial");

menu.add(0, 1, 0, "Info");

menu.add(0, 2, 0, "Keluar");

return true;

}

79:

80: public boolean onOptionsItemSelected(MenuItem item) {

81:

54

if (item.getItemId() == 0) {

    new AlertDialog.Builder(this)

        .setTitle("Info Aplikasi")

        .setMessage("1. Masukkan harga asli "

            + "\n2. Masukkan besar diskon"

            + "\n3. Klik gambar HITUNG DISKON!"

            + "\n4. Klik tombol Reset untuk mengosongkan data")

        .setNeutralButton("Kembali",

            new DialogInterface.OnClickListener() {

                @Override

                public void onClick(DialogInterface arg0,

                    int arg1) { //

```

```

// TODO Auto-generated method stub

}

}).show();

} else if (item.getItemId() == 1) {

    new AlertDialog.Builder (this)

        .setTitle("Info Aplikasi")

        .setMessage(

            "Aplikasi DISKON! dibuat oleh OmAyib2011."

            + "Kunjungi www.omyib.com."

        )

        .setNeutralButton("Kembali",

            new DialogInterface.OnClickListener() {

                @Override

                public void onClick(DialogInterface arg0,

                    int arg1) { //

                    // TODO Auto-generated method stub

                    }).show();

                } else if (item.getItemId() == 2) {

                    111: hargaDiskon.this.finish();

                }

            return true;

        }

    }

```

DAFTAR PUSTAKA

1. Winarno, Edy. Ali Zaki, SmitDev Community. 2010. Membuat Sendiri Aplikasi Android Untuk Pemula. Elexmedia. Jakarta
2. Hidayatulloh, Priyanto. 2013. Membuat Mobile Game Edukatif Dengan Flash. Informatika. Bandung
3. Azhari, Mukti Adi. Agung Arnas Wibowo. 2012. Membuat Game Edukasi Dengan HTML 5 Dan Android Studio. Lokomedia. Jakarta
4. Pratama, Galih. 2012. Seminggu Menguasai Android Studio. BelajarKoding. Jakarta