

android dan anak tukang sayur

Buku Praktis Belajar Pemrograman Android



- ☐ Pembuatan Project
- ☐ Pengenalan User Interface
- ☐ Text View
- ☐ Button
- ☐ Check Box
- ☐ Radio Button
- ☐ Spinner
- ☐ Progress Bar
- ☐ Project Latihan
- ☐ dan sebagainya



Edisi 1.0
Bahasa Indonesia

Anna Arthdi Putra



ANDROID

Aplikasi Java Translate ,SMS Sain , Aplikasi Bean
Shell , Androbot

Anna Arthdi putra
@annaarthdiputra
wartech-id.com



Kata Pengantar

Assalamualaikum wr.wb

Kegiatan membaca dan menulis sangat sederhana untuk mendapatkan pengetahuan , karena tanpa kita membaca kita tidak akan dapat menulis . Buku yang berjudul Android dan Anak Sayur saya ambil tema dari kehidupan pada diri saya , saya merupakan anak dari kedua orang tua saya , ibu tercinta saya penjual Sayur di daerah Lubuk linggau dan ayah saya karyawan , Buku ini sangatlah sederhana , karena saya juga di ajarkan keserhanaan kepada kedua orang tua saya , hidup itu gak perlu gengsi , kalau sesuatu masih bisa di kerjakan kerjakan sendiri “Kerjakan sendiri ”dan kalau sesuatu masih layak untuk di gunakan , perjuangkan . Buku ini saya rangkum dari tahun 2011 pada saat saya masih semester 3 , allhamdulillah selesai dan sesuai yang saya harapkan , walaupun sangat banyak kekurangan dari buku ini.

Buku Ini merupakan Buku pertama saya , dan Insya Allah akan menyusul buku yang lainnya , **Buku ini akan bersifat Gratis dan layak untuk di kembangkan lagi atau pun di sebarakan secara luas** , Bila pembaca bermaksud untuk mengkomersialkan sebaiknya di bicara kan terlebih dahulu kepada saya , bisa langsung ke kediaman saya di Jl.cendrawasih blog 0 banguntapan bantul atau bisa twett saya di @annaarthdiputra , dan semua pemasukkan dari buku ini akan saya sumbangkan kepada Panti asuhan Hidayah , di Jl.Yossudarso No 144 Lubuklinggau Sumatera Selatan.

Saya sangat berterima kasih kepada StackOverflow , Blog blog teman teman yang memberi postingan yang dapat saya kembangkan lagi sehingga menjadi buku ini ,dan Saya berterimakasih kepada my brother Anna Erdiawan yang menyatuhkan buku dengan cover , bila teman teman ingin meminta bantuan jasa desain bisa hubungi beliau di twit @annaerdiawan , mungkin ada teman teman yang memiliki scrip yang sama saya mohon maaf , karena saya lupa sumber yang saya ambil, tetapi bila teman teman merasa ada yang sama bisa hubungi saya dan akan saya perbaharui, saya berterima kasih kepada orang tua yang turut mendukung hingga saat ini , kepada allah saya mohon ampun dan saya ucapkan wassalam mualaikum wr.wb . Selamat belajar.

Lubuklinggau , 30 Juli 2012



Anna arthi putra



Daftar ISI

| | |
|--|----|
| Kata pengantar | 3 |
| Daftar isi | 4 |
| Pengenalan Android | 6 |
| Kenapa Harus Android | 6 |
| Instalsi Android | 8 |
| • Instalasi menggunakan url | 8 |
| • Instalasi menggunakan ADT | 11 |
| Pengenalan layar kerja eclipse IDE | 12 |
| • activity | 13 |
| • xml layout | 14 |
| • pallate | 16 |
| • properties | 17 |
| • informasi | 18 |
| Pembuatan project | 19 |
| Pengenalan user interface | 21 |
| • Text View | 21 |
| • Button | 21 |
| • Toggel Button | 22 |
| • Check Box | 22 |
| • Radio Button | 23 |
| • Spinner | 23 |
| • Progress Bar | 24 |
| • Radio Button | 24 |
| • Ratting | 25 |
| • Plain Text | 25 |
| • Linear Layout | 26 |
| • Relative Layout | 26 |
| • Frame Layout | 26 |
| • Table Layout | 27 |
| • List View | 28 |
| • Spinner | 28 |
| • Grid View | 29 |
| • Sliding Drawer | 28 |
| • Images , Images Button | 30 |
| • Time and Date | 30 |
| Project Latihan..... | 30 |
| • Alerg Dialog | 31 |
| • Check Box | 32 |
| • Download File by Progress | 36 |
| • Dua Activity | 41 |
| • External Font | 45 |
| • Grid View | 48 |
| • List View | 50 |



| | |
|--|-----|
| • Menu | 52 |
| • Scrol View | 55 |
| • Set Date and Time | 58 |
| • Sound..... | 62 |
| • Spinner | 63 |
| • Tab Host | 65 |
| • Text Watcher | 69 |
| • Phone Gap | 70 |
| • Contact Manager | 72 |
| Data Base Internal pada Android (sqlite) | 88 |
| • SQLite dalam Android | 89 |
| • Struktur SQLite | 89 |
| HTTP Request | 96 |
| Facebook Desain | 98 |
| JSON Parsing | 98 |
| Google Map | 107 |
| Sensor | 110 |
| • Pengenalan Sensor | 110 |
| • Sensor Framework | 110 |
| • Sensor Manager | 110 |
| • Sensor Availability | 111 |
| • Monitoring Sensor Events | 113 |
| • Handling Different Sensor Configurations | 117 |
| • Sensor Coordinate System | 118 |
| • Best Practices for Accessing and Using Sensors | 119 |
| • Motion Sensor | 120 |
| • Gravity Sensor | 121 |
| • Gyroscope | 123 |
| • Linear Accelerometer | 125 |
| • Rotation Vector Sensor | 126 |
| • Sensor Suara | 130 |
| • Bluetooth Socket | 134 |
| Aplikasi SMSsain | 142 |
| Aplikasi BeanShellMini | 167 |
| Aplikasi Kurus Bonang | 171 |
| Aplikasi Mini java Translate | 173 |
| Game Androbot | 171 |



Pengenalan Android

Android banyak di katakan sebuah robot hijau , karena dalam logo yang di kenalkan oleh raksasa mesin pencari yaitu google.inc .Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilis 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).

Kenapa Harus Android

Android Merupakan sistem operasi yang berbasis linux , yang pastinya android bersifat terbuka oleh karena itu banyak pengembang yang ingin mengembangkan , sehingga dari versi pertama dan hingga terakhir 4.0 dengan cepat di rilis , dan juga android lebih cepat berkembang karena android di akusisi oleh google . Yang merupakan raksasa mesin pencari saat ini , jadi tidak ragu lagi kalau kita mendvelop android sendiri.

Sejarah

Pada Juli 2000, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc. bekerja pada Google, di antaranya Andy Rubin, 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. versi android terbaru yaitu versi 4.0. Android juga sudah bergabung dengan beberapa smart mobile seperti Nokia, Sony Ericsson, dan lainnya.

➤ 2007-2008: Produk awal

Sekitar September 2007 sebuah studi melaporkan bahwa Google mengajukan hak paten aplikasi telepon seluler (akhirnya Google mengenalkan Nexus One, salah satu jenis telepon pintar GSM yang menggunakan Android pada sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010).



- Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, Android, perangkat bergerak (mobile) yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru.
- Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android.
Android versi 1.1
- Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, voice search (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.
- Android versi 1.5 (Cupcake)
Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (Software Development Kit) dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.
- Android versi 1.6 (Donut)
Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang diintegrasikan; CDMA / EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine; kemampuan dial kontak; teknologi text to change speech (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.
- Android versi 2.0/2.1 (Eclair)
Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1. Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi mobile terbaik (killer apps - aplikasi unggulan). Kompetisi ini berhadiah \$25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik. Dengan semakin berkembangnya dan semakin bertambahnya jumlah handset Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi Android. Aplikasi terkenal yang diubah ke dalam sistem operasi Android adalah Shazam, Backgrounds, dan WeatherBug. Sistem operasi Android dalam situs Internet juga dianggap penting untuk menciptakan aplikasi Android asli, contohnya oleh MySpace dan Facebook.
- Android versi 2.2 (Froyo: Frozen Yoghurt)
Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya



terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market.

- **Android versi 2.3 (Gingerbread)**
Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (reverb, equalization, headphone virtualization, dan bass boost), dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.
- **Android versi 3.0/3.1 (Honeycomb)**
Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (hardware) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom. Perangkat tablet dengan platform Android 3.0 akan segera hadir di Indonesia. Perangkat tersebut bernama Eee Pad Transformer produksi dari Asus. Rencana masuk pasar Indonesia pada Mei 2011.
- **Android versi 4.0 (Ice Cream)**
Android versi 4.0 dirilis akhir tahun 2011. Platform yang dapat berjalan di table dan mobile sehingga tidak ada perbedaan android table dan android mobile.
- **Android versi 4.1 (Jelly Beans)**
Android 4.1 Jelly Beans akan di rilis akhir 2012.

Instalasi Android

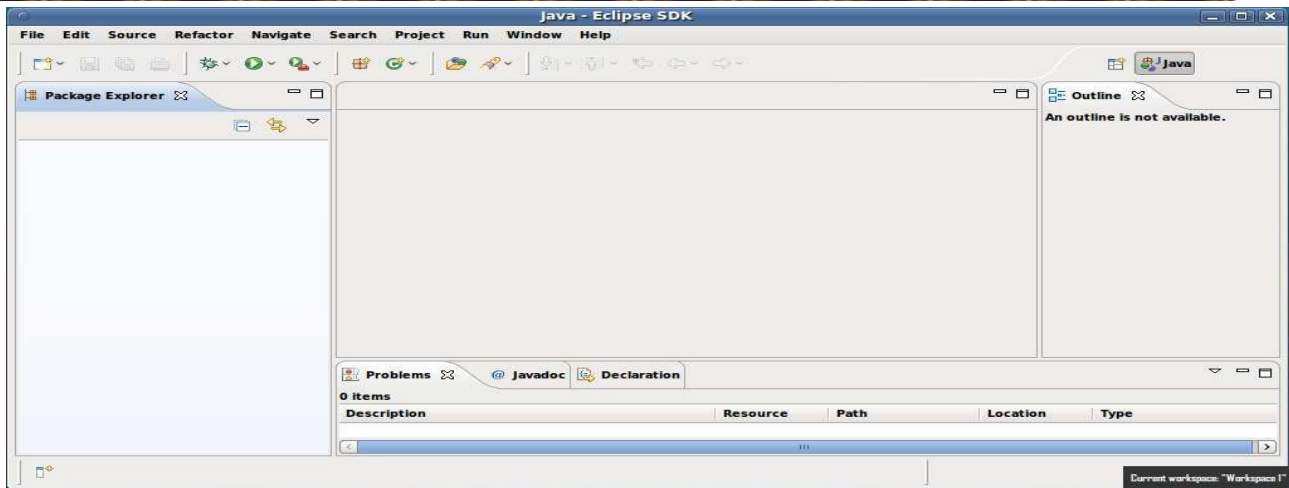
Instalasi android di eclipse di bilang awal dari segalanya , instalsi ini sering di sebut dengan ADT. Ada beberapa yang harus di butuhkan (wajib), karena dengan adanya syarat – syarat ini kita dapat membuat aplikasi android dengan sebaik mungki.

1. Eclipse IDE
2. ADT-Android
3. SDK-Android

ada 2 cara untuk menginstall adt android ,dengan cara mendownload adt- android atau dengan menginstall melalui path alamat URL android.

➤ Instalasi pertama Menggunakan URL Android

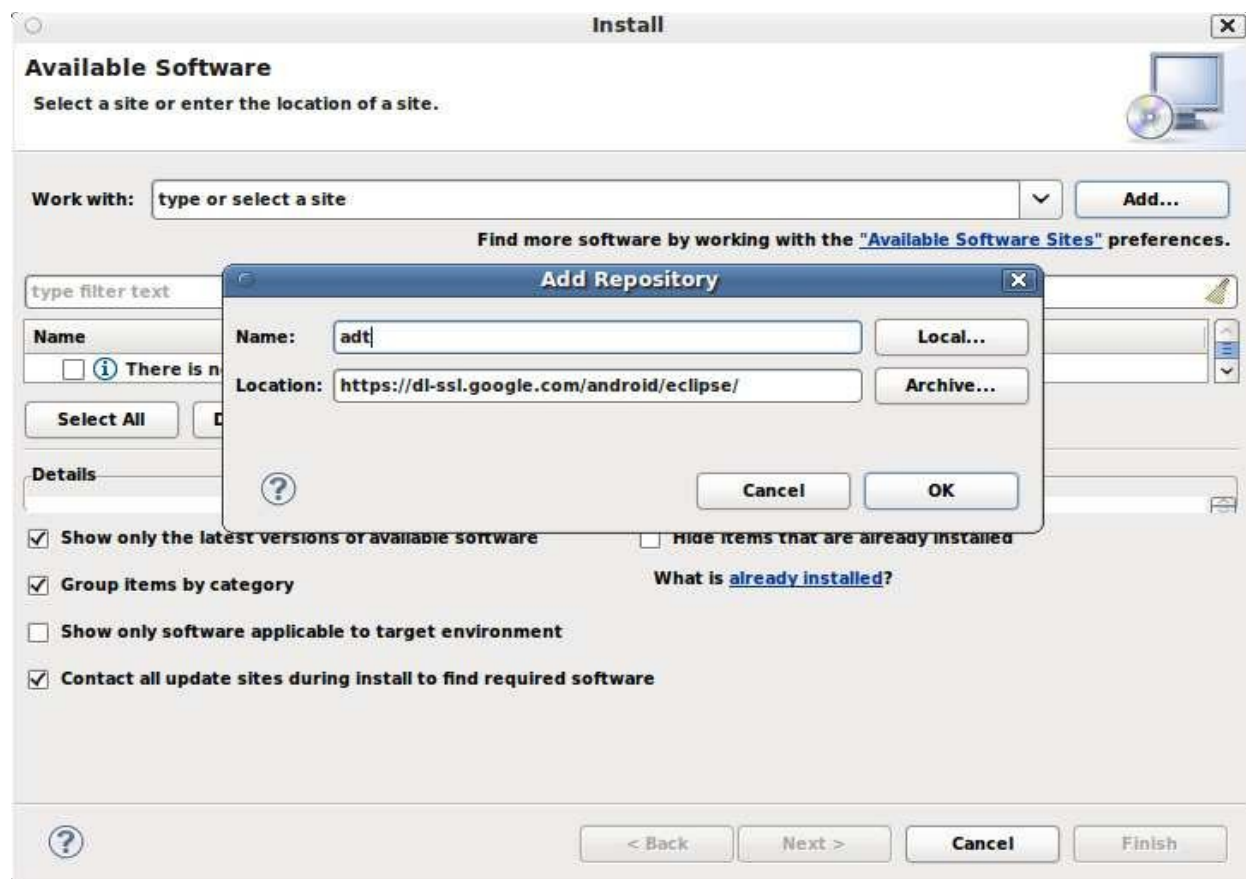
1. Buka Eclipse IDE



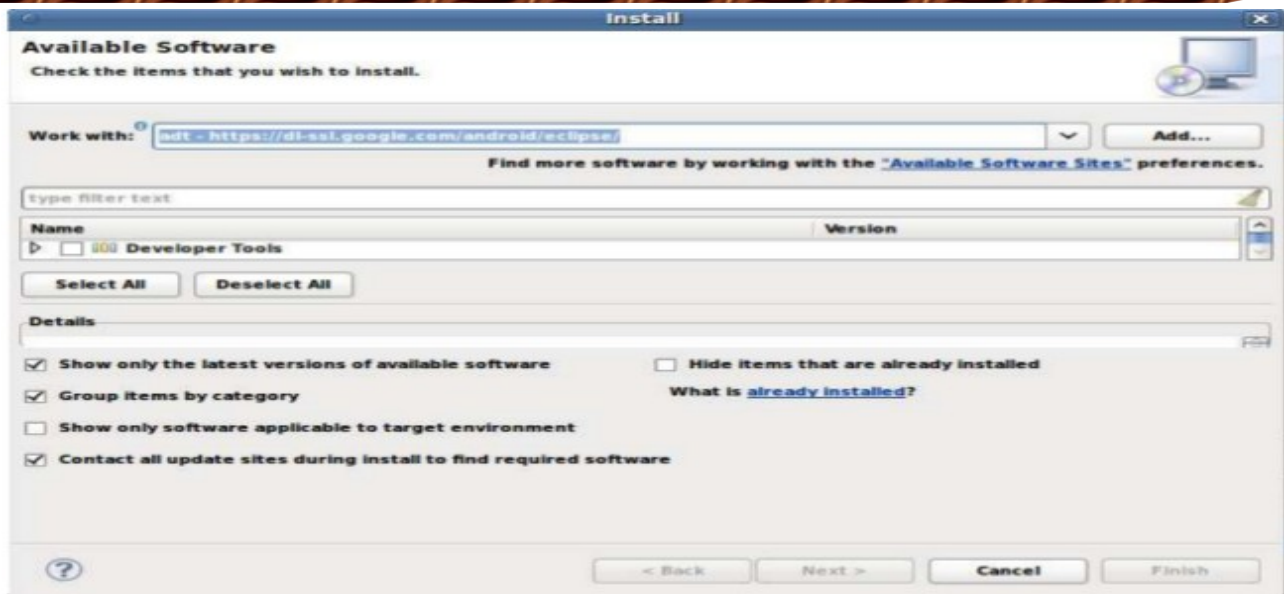
2. Klik Help – Install new software
3. add isi kan

Name : adt

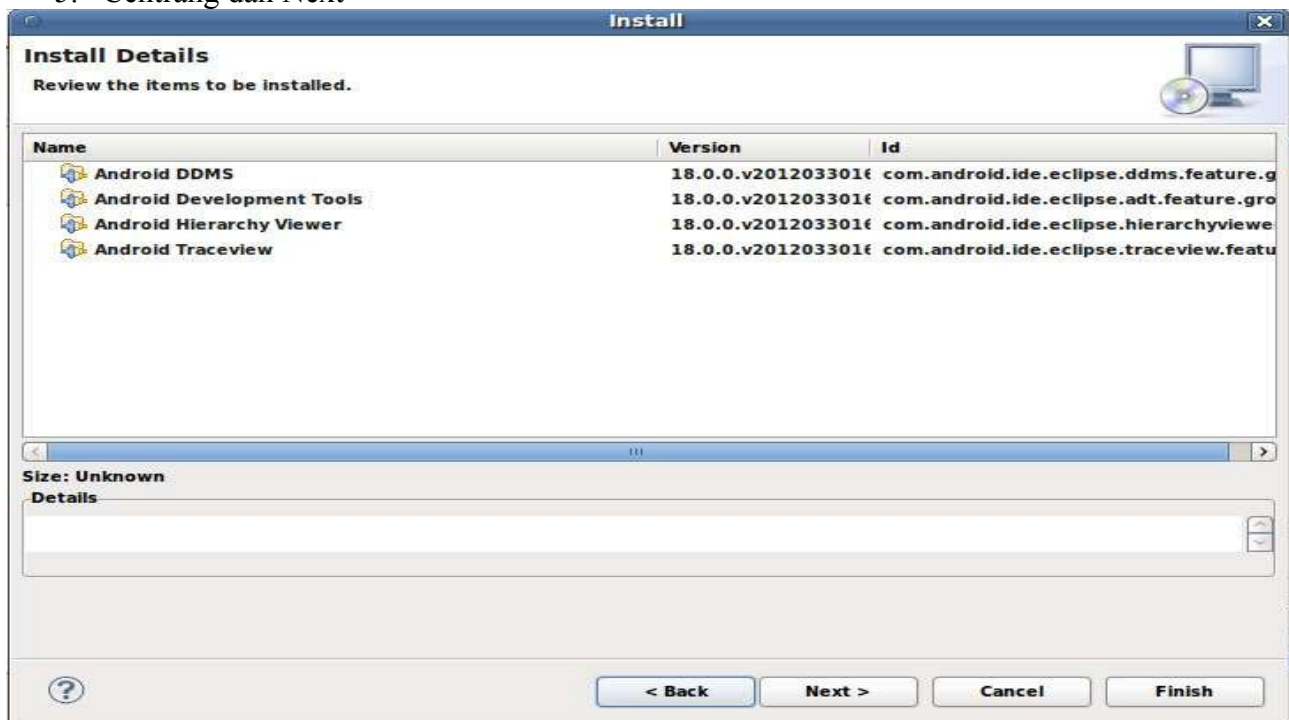
Location : <https://dl-ssl.google.com/android/eclipse/>



4. Tunggu beberapa saat saat Developer Tools ter lihat di bagian TextEdit.



5. Centrang dan Next





7. Next – dan tunggu proses Installing software selesai



➤ Menginstall android Menggunakan ADT Android

1. Download ADT Android pada android developer.
2. Buka eclipse IDE
3. Klik Help – New Software
4. Pilih Add

Name : adt



Location : [Alamat ADT](#) pada directory anda ex:/home/root/adt.zip

Pengenalan Layar Kerja Eclipse

Untuk bekerja dengan apa pun di Android, Anda perlu sebuah proyek. Dengan Java biasa, jika Anda ingin, Anda hanya bisa menulis program sebagai satu file, compile dengan javac, dan menjalankannya dengan java, tanpa setiap struktur pendukung lainnya. Android adalah lebih kompleks, tapi untuk membantu tetap dikelola Google telah memasok alat untuk membantu membuat proyek. -Jika Anda menggunakan Android diaktifkan IDE, seperti Eclipse dengan plugin Android (tersedia dalam SDK Android), Anda dapat membuat sebuah proyek dalam dari IDE (File pilih New - Project, kemudian pilih Android - Android Project). Jika Anda menggunakan alat yang tidak Android diaktifkan, Anda dapat menggunakan script activitycreator, ditemukan dalam alat / direktori instalasi SDK Anda. Hanya lulus activitycreator paket nama kegiatan yang akan dibuat dan - keluar saklar yang menunjukkan di mana file proyek harus dihasilkan. Berikut ini contohnya:

```
activitycreator - out / path / to / my/ project/ dir \
com.commonsware.android.Now
```

Anda akan berakhir dengan beberapa pra-generated file, seperti yang dijelaskan dalam Bab 2. Kami akan menggunakan file-file untuk sisa bab ini. Anda juga dapat men-download direktori proyek dari sampel ditunjukkan dalam buku ini dalam ZIP file pada CommonsWare Web site¹. Proyek-proyek tersebut siap untuk digunakan, Anda tidak perlu menjalankan activitycreator pada contoh tersebut dibongkar.

Activity



Src proyek Anda / direktori berisi Java-style tree standar direktori berdasarkan paket Java yang digunakan saat membuat proyek. Di dalam direktori tersebut dalam Anda harus mencari pra- dihasilkan sumber file bernama Now.java, yang mana kegiatan pertama Anda akan pergi. kegiatan ini akan berisi satu tombol yang menampilkan kali tombol terakhir ditekan (atau saat aplikasi dimulai jika tombol belum ditekan). Buka selamatdatangactivity.java scrip yang akan muncul seperti berikut:

```
package selamat.wartec;
import android.app.Activity;
import android.os.Bundle;
public class SelamtBelajarActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```




```
}
```

```
package selamat.wartec;
import android.app.Activity;
import android.os.Bundle;
```

Package selamat.wartec merupakan nama package yang di gunakan , Import android.app .Activity merupakan Library yang di gunakan biasanya kalau mengalami error import Library dengan mudah tinggal arahkan kursor ke arah tanda ! Dan klik dan pilih Import.

```
public class SelamtBelajarActivity extends Activity {
```

public class SelamtBelajarActivity merupakan nama class yang di gunakan bertujuan untuk menandakan alamat class yang di gunakan bersifat public , sedangkan extends Activity , merupakan turunan dari Activity yang di gunakan.

```
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
```

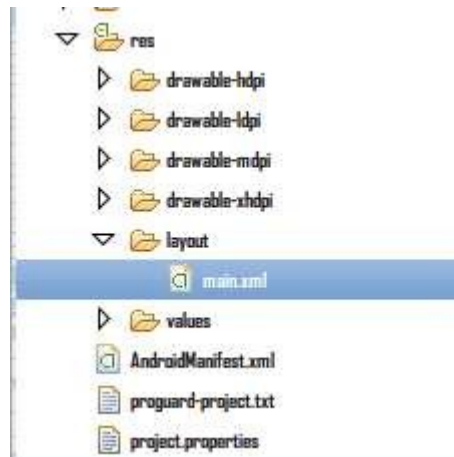
@Override merupakan bersifat abstrac , public void onCreate(Bundle savedInstanceState) merupakan method untuk memanggil aktivitas lain , setContentView(R.layout.main); merupakan method yang berisi alamat Id yang terdapat pada R.java , dengan adanya ini dapat terhubung dengan desain XML.

```
package selamat.wartec;
public final class R {
public static final class attr {
}
public static final class drawable {
public static final int ic_launcher=0x7f020000;
}
public static final class layout {
public static final int main=0x7f030000;
}
public static final class string {
public static final int app_name=0x7f040001;
public static final int hello=0x7f040000;
}
}
```

R.java merupakan alamat id yang di otomatisasi di isi , bila layout membuat suatu tool seperti `<Button android:id="@+id/idotomatis"/>` nah dengan android:id yang di buat pada Layout XML , dan di dalam R.java di deklarasi Drawable , string.xml dan yang lainnya . akan di deklarasi di R.java , jadi jangan khawatir dengan R.java . Bila terjadi Error dengan R.java Biasanya itu ada penulisan .



XML Layout



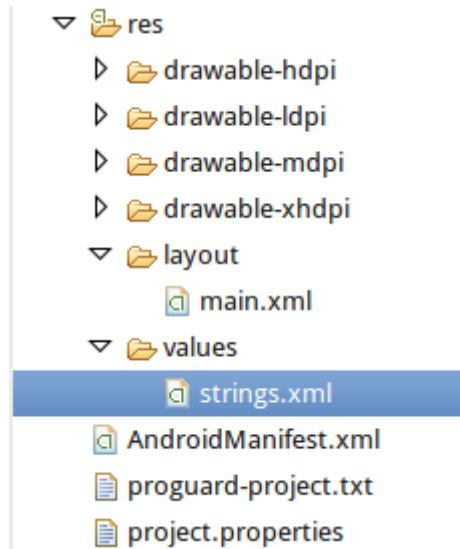
Seperti namanya, tata letak yang berbasis XML adalah spesifikasi hubungan widget 'satu sama lain' dan untuk kontainer mereka -dikodekan dalam format XML. Spesifik- Cally, Android menganggap berbasis XML layout untuk menjadi sumber daya, dan sebagai file tata letak tersebut disimpan di direktori res / t dalam proyek Android. Setiap file XML berisi pohon elemen menentukan tata letak widget dan kontainer mereka yang membentuk satu tampilan hirarki Atribut dari elemen XML adalah properti, yang menggambarkan bagaimana widget harus melihat atau bagaimana sebuah wadah harus bersikap. Misalnya, jika elemen Tombol memiliki nilai atribut android: textStyle = "berani", yang berarti bahwa teks muncul di tampilan tombol harus diberikan dengan gaya font tebal. Android SDK ships tools (AAPT) yang menggunakan layout. Alat ini harus auto- matically dipanggil oleh rantai alat Android Anda (misalnya Eclipse, build.xml Ant). Tentu tertentu penting untuk Anda sebagai pengembang adalah bahwa AAPT menghasilkan file sumber dalam R.java pada proyek, yang memungkinkan Anda untuk mengakses layout dan widget dalam layout yang langsung dari Anda Kode Java.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

</LinearLayout>
```

Pada folder res/values dan string.xml itu merupakan pendeklerasian id string yang digunakan untuk mempermudah tidak terjadi string yang panjang pada pembuatan di layoutt xml , biasanya dengan string yang panjang di masukkan di layout xml akan mengalami peringatan atau tanda ! (Perhatian) kalau string yang di gunakan cukup panjang , dengan adanya string.xml kita dapat membuat id string di dalamnya agar tidak terjadi perhatian karena tanda! Merupakan bug yang muncul pada aplikasi yang di buat.

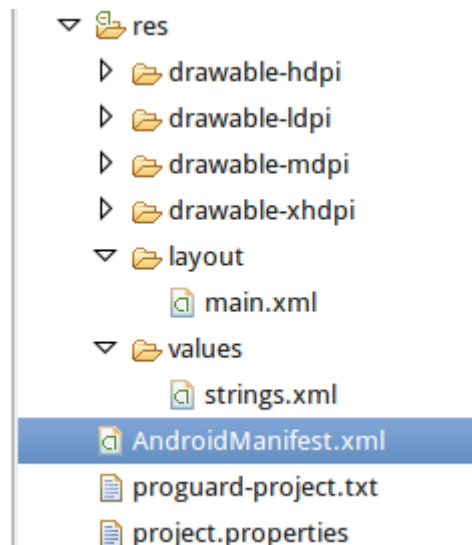


```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="hello">Hello World, BahanActivity!</string>
    <string name="app_name">Bahan</string>

</resources>
```

Pada folder AndroidManifest.xml merupakan tempat penghubung antara activity java dengan layout.xml .



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

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

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



```

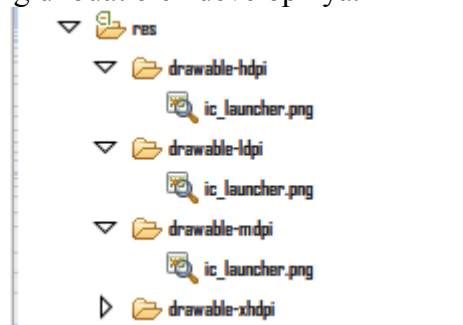
<activity
    android:name=".BahanActivity"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
</application>

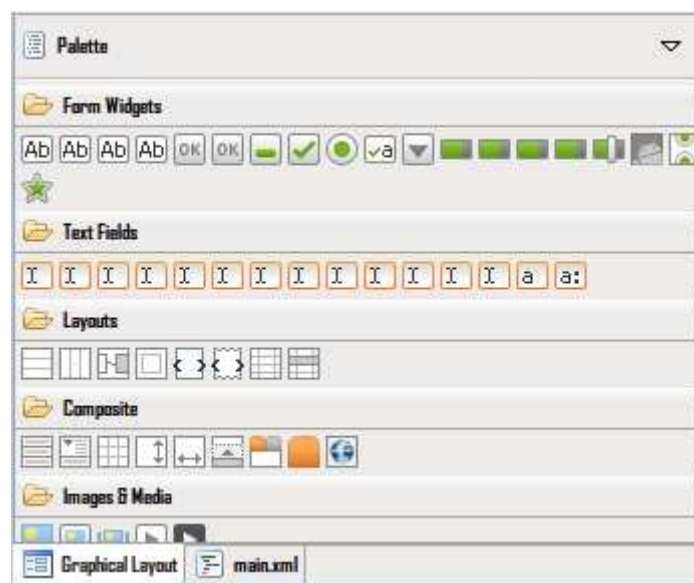
</manifest>

```

Sedangkan pada folder drawable yang terdapat pada res / drawable-hdpi ,res / drawable-ldpi , res / drawable-mdpi . Merupakan tempat kumpulan gambar – gambar untuk menambakan pada aplikasi yang akan kita buat , Biasa nya di sini akan terisi folder yang lain seperti menu , raw dan yang lainnya tergantung dengan apa yang di buat oleh develop nya.



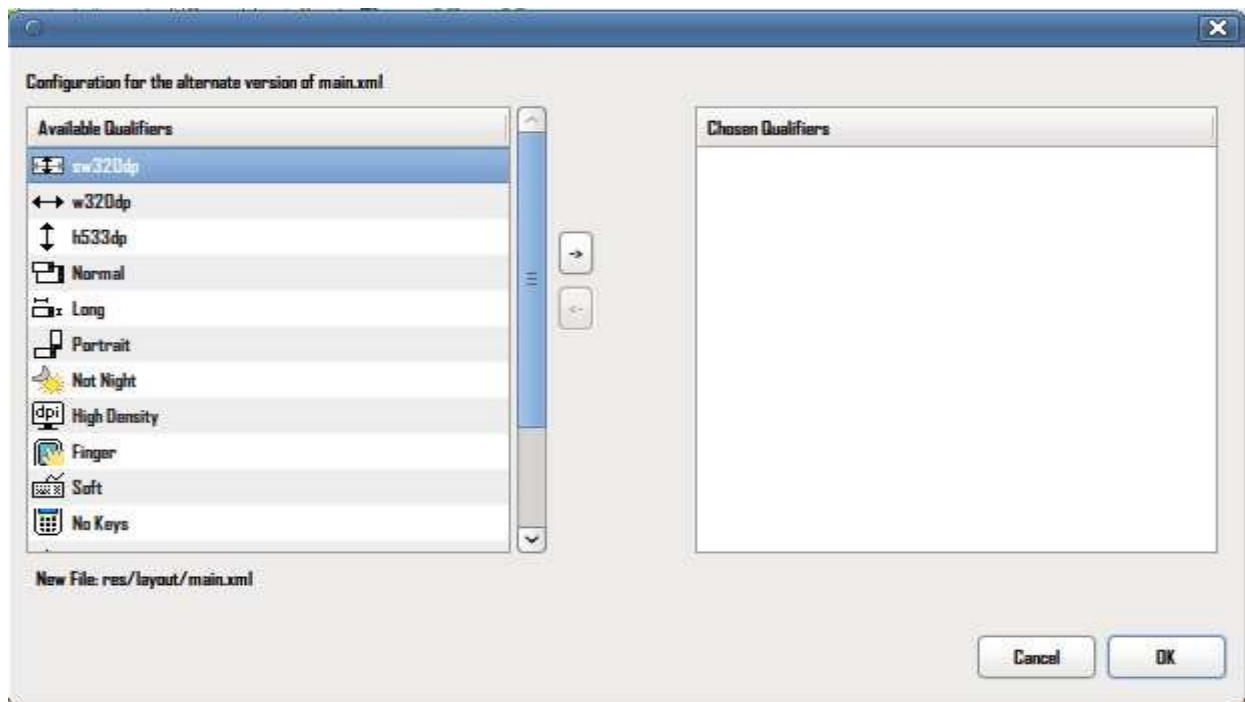
Pallate



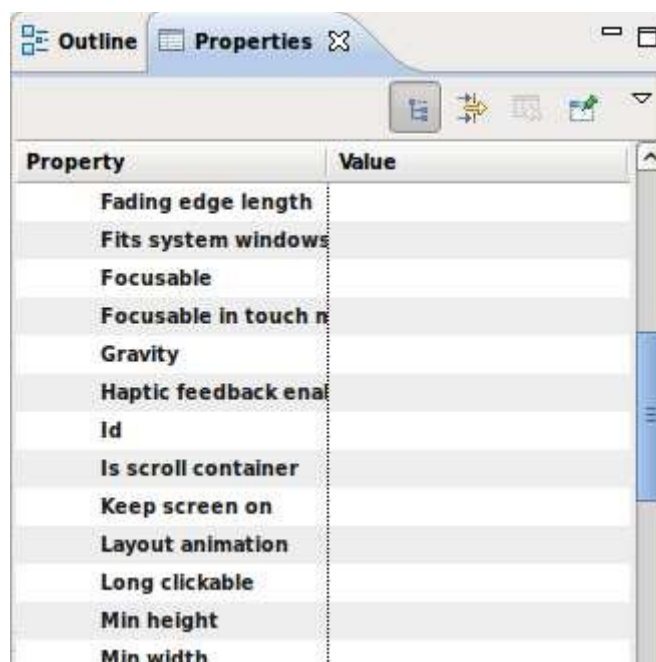
Pallate merupakan pallate bantuan yang bertujuan memudahkan develop untuk membuat suatu aplikasi , dengan adanya ini kita tinggal pilih dan drag ke lembar kerja main.xml tanpa menulis scrip nya secara langsung.



Pada gambar diatas merupakan Editing config default yang di buat dengan manula , di atas terdapat screen tampilan yang di gununakaan , layar kerja berbentuk Portrait atau Landscape , Day Time , Themes tampilan pada layar kerja , serta platform yang digunakan dan ada fungsi lain pada xml editing ini . Klik pada tombol create dan akan muncul lembar kerja seperti di bawah ini.



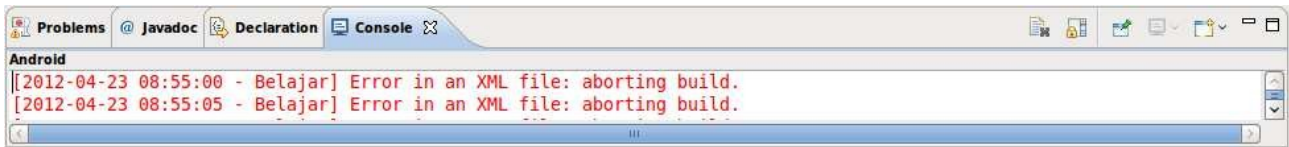
Properties





Dalam Layar kerja Properties kita dapat mengedit tampilan seperti dalam IDE yang lain , tanpa pengedita Scrip dapat di lakukan di sini , seperti penamaan ID , Mengganti gambar , dan sebagainya.

Informasi

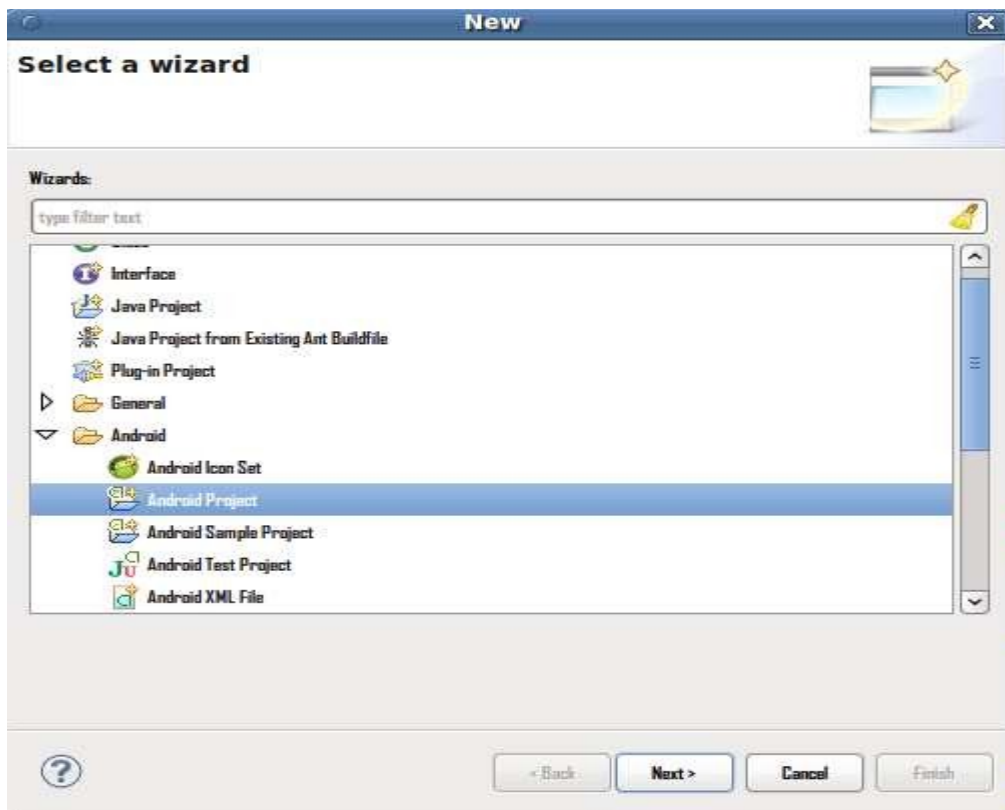


Dalam Informasi yang dapat di pesankan kepada Developer yang sedang aktif membuat akan di lakukan di Information , di sini akan mengeluarkan pesan bila ada Project atau pun scrip yang tidak sesuai.

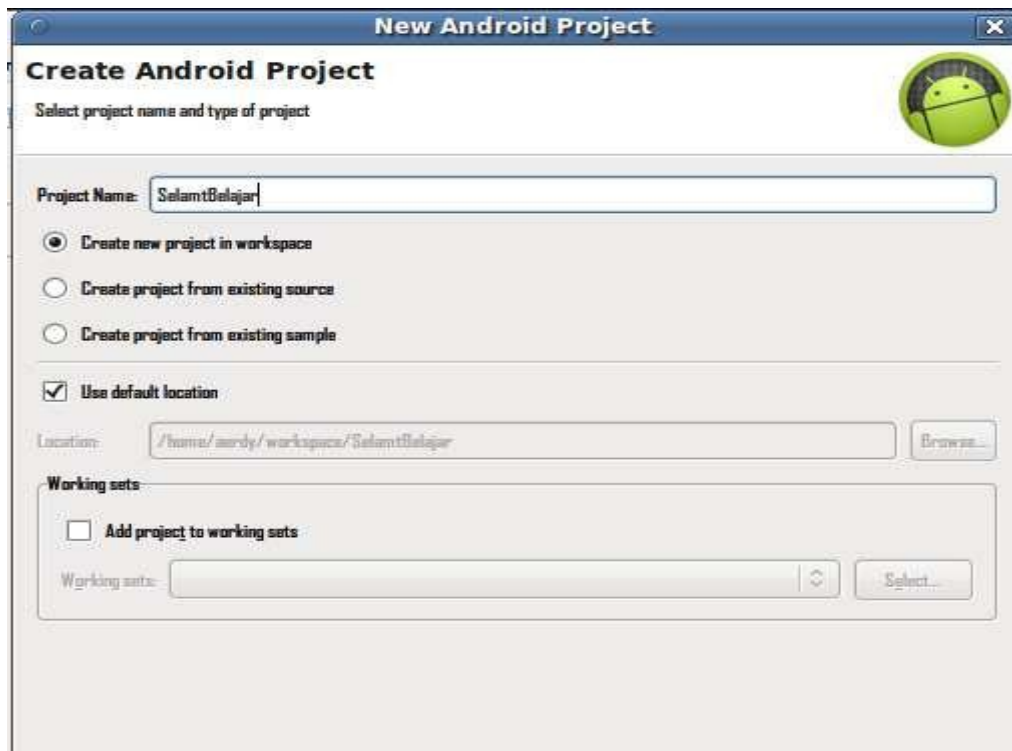
Pembuatan Project

Dengan mencapai beberapa tahapan , dari install plugin ADT hingga sdk – android ,serta dengan Plathform yang di keluarkan android dari 2,1 – 4.0.3 kita dapat menggunakan semua Plathform tersebut , Untuk pembuatan project sebagai berikut :

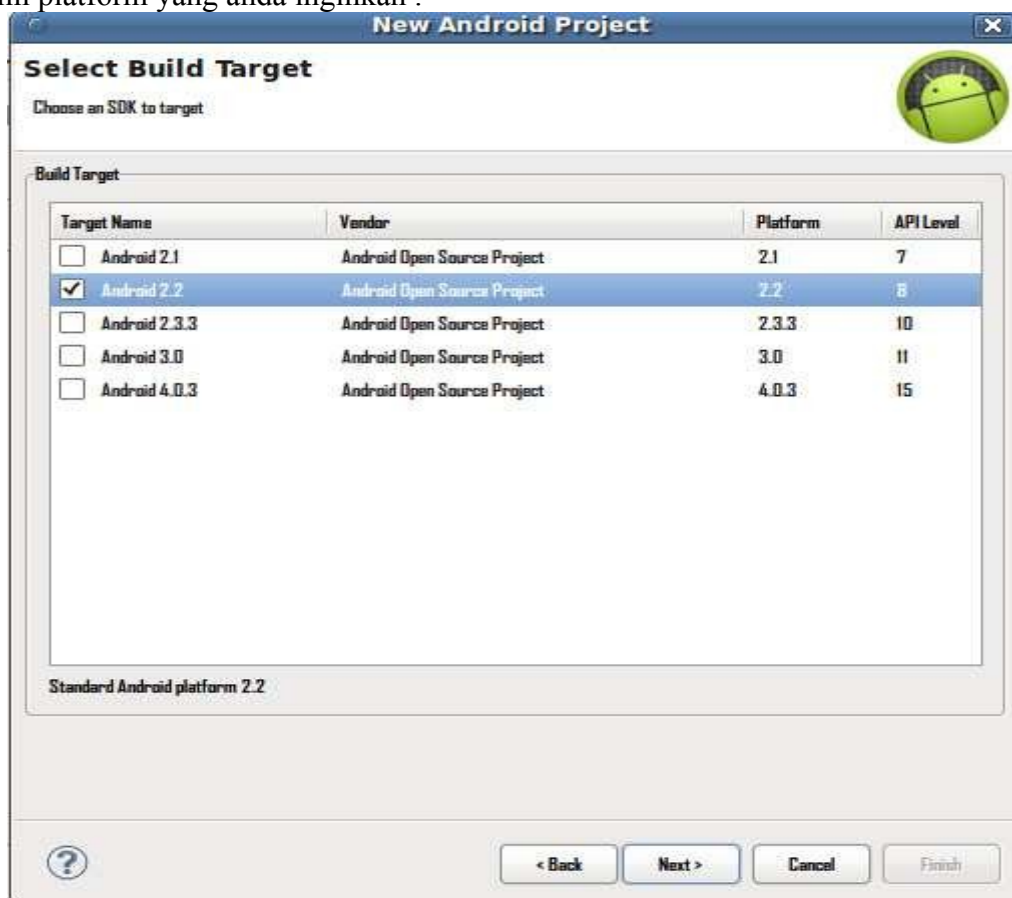
1. Buka Eclipse yang telah terinstall Plugin ADT – Hingga SDK android
2. Menu File – New – Android Project – Nex t



3. Isi Project Selamat datang dan next



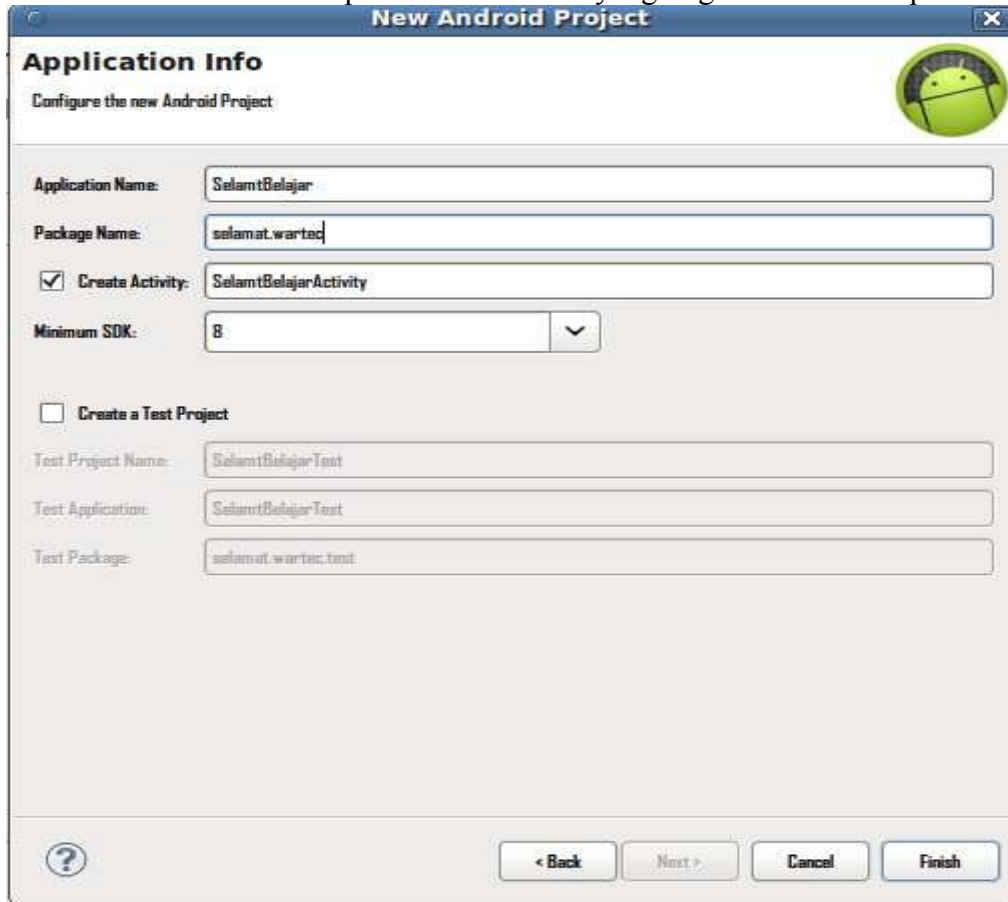
4. Pilih platform yang anda inginkan .



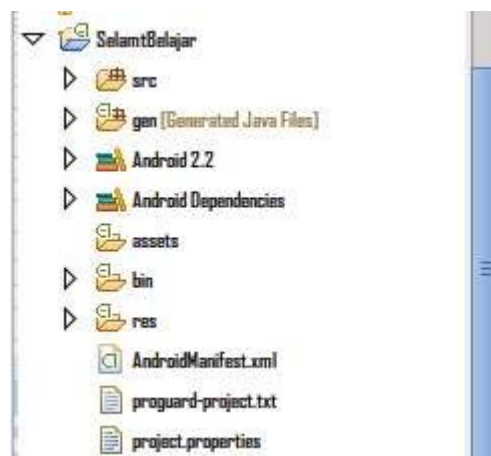
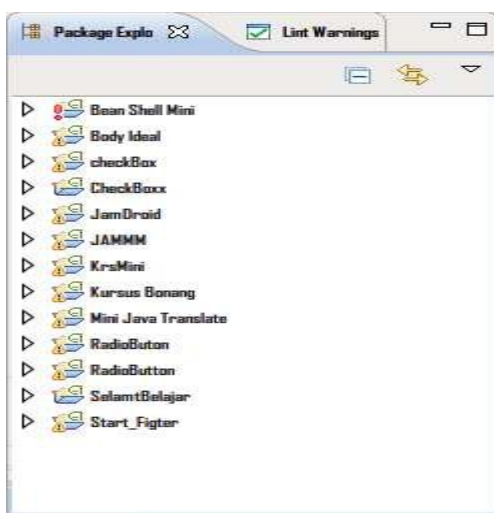
5. Pengisian data Aplikasi info , yang bertujuan untuk tidak ada duplikasi aplikasi yang di buat oleh orang lain (Gambar 1.4)



Application Name : Nama Project(Biarkan Default yang di berikan)
 Package Name : Nama Package yang digunakan (Contoh : SelamatBelajar.wartec)
 Create Activity : (Biarkan Default)
 Minimum SDK : Minimum Spesifikasi Platform yang di gunakan untuk aplikasi



6. Finish , semuanya akan terlihat pada window Project Explorer pada gambar di bawah merupakan isi dari beberapa package dan activity pada android.

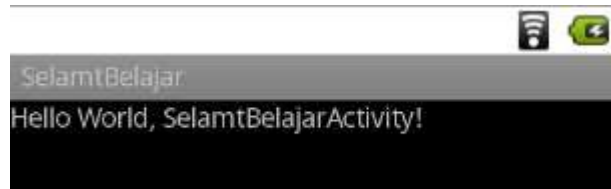




Pengenalan User Interface

Dalam ini kita akan mulai membuat user interface , dalam android kita dapat melakukan dua cara yaitu dengan menggunakan bantuan palette dan scrip di tulis secara langsung , sesuai dengan develop lakukan .

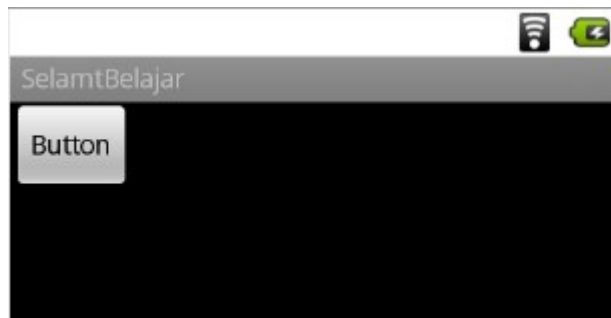
1. Text View



Pada tools Pallate , kita langsung dapat mendrag Text view yang di gunakan , ada beberapa pilihan pada Pallate , tetapi ini kita dapat lakukan dengan mengedit sctip yang ada pada XML .

```
<TextView
    android:id="@+id/textview"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:textColor="#ffffff"
    android:textSize="3in"
    android:textStyle="bold"
    android:text="@string/hello" />
```

2. Button

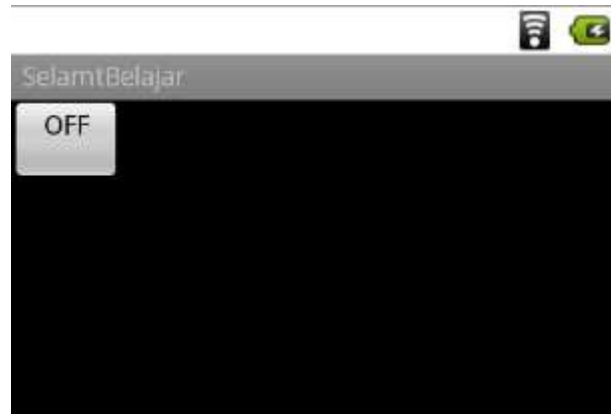


Pada Button terdapat beberapa pilihan yang terdapat pada tools pallate , sama seperti text view kita dapat meng edit scrip agar sesuia dengan apa yang di tampilkan.

```
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button" />
```



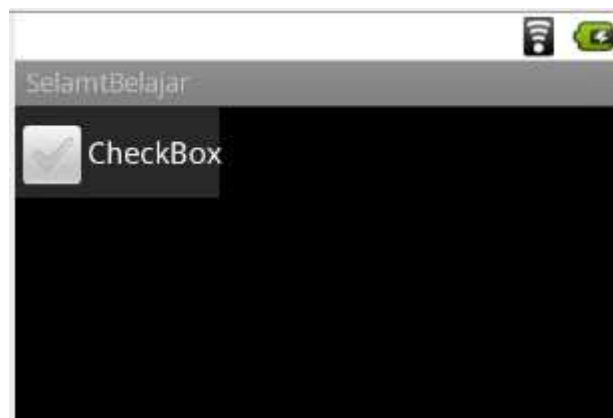
3. Toggel Button



Fungsi nya sama seperti button biasa .tetapi denga toggle button ini berisat boolean , dia bila di tekan berifat true , bila tidak di tekan bersifat false .

```
<ToggleButton  
    android:id="@+id/toggleButton1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="ToggleButton" />
```

4. Check Box



CheckBox Merupakan tools yang sering di gunakan karena simple penggunaan nya , CheckBox bersifat Boolean bila di check maka true bila tidak maka false.

```
<CheckBox  
    android:id="@+id/checkbox1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="CheckBox" />
```



5. Radio Button



Radio Button sangat sering di gunakan dengan , Radio Button bersifat True bila di tandai , dan Bila tidak bersifat false.

```
<RadioButton  
    android:id="@+id/radioButton1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="RadioButton" />
```

6. Spinner

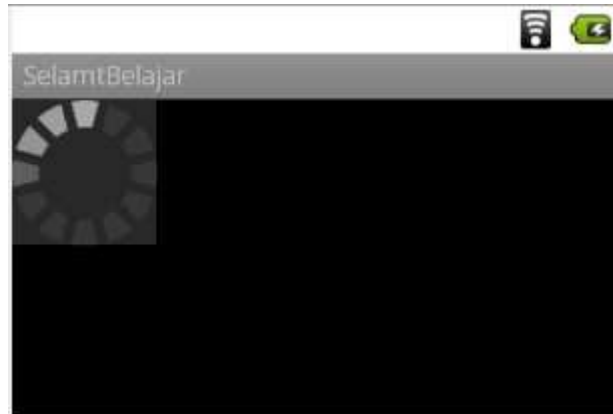


Spinner merupakan pilihan yang banyak di gunnakan karena simple dan praktis , tidak memakan banyak ruang , spinner sering di temukan pada layar desktop yaitu listbox , fugsinya sama , tinggal berikan perintah dan berapa list yang akan di tampilkan.

```
<Spinner  
    android:id="@+id/spinner1"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" />
```



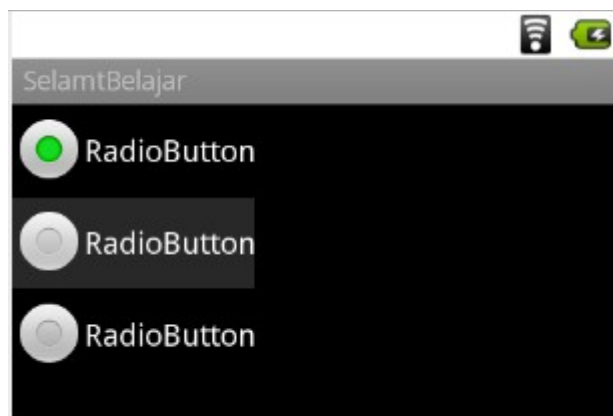
7. Progress Bar



Progress bar sering di temukan di aplikasi desktop dan game – game ini nama lainnya progress loading , dengan adanya ini kita dapat memberi timer pada saat aplikasi ingin berjalan atau melakukan proses.

```
<ProgressBar
    android:id="@+id/progressBar1"
    style="?android:attr/progressBarStyleLarge"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

8. Radio Button



Radio Group sebenarnya sama dengan Radio Button , tetapi di Radio Group terdiri dari lebih dari satu , dengan adanya radio group dengan mudah mengatur letak Radio Button tersebut .

```
<RadioGroup
    android:id="@+id/radioGroup1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >

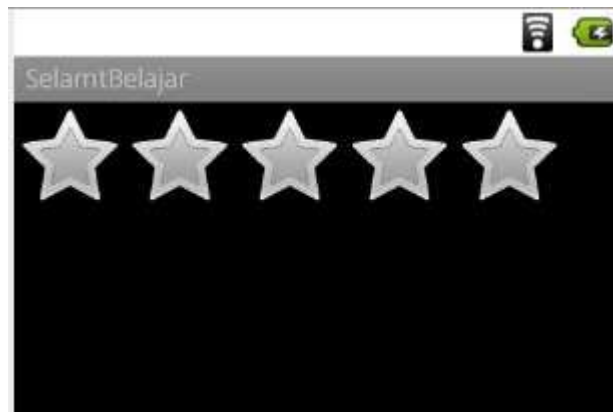
    <RadioButton
        android:id="@+id/radio0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checked="true"
        android:text="RadioButton" />
```




```
<RadioButton
    android:id="@+id/radio1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="RadioButton" />

<RadioButton
    android:id="@+id/radio2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="RadioButton" />
</RadioGroup>
```

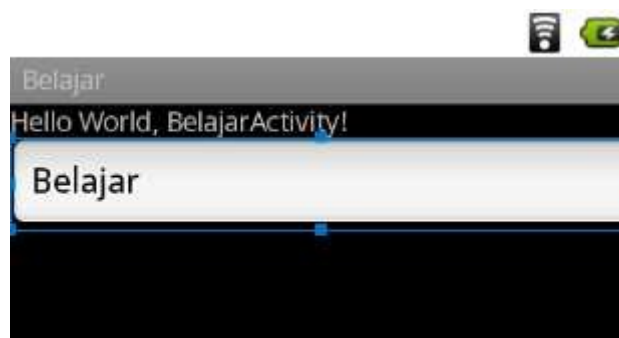
9. Rating



Rating Bar , merupakan tools yang digunakan untuk menilai suatu proses , seperti menilai proses memasukkan . Rating nya mau di beri berapa bintang.

```
<RatingBar
    android:id="@+id/ratingBar1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

10. Plain Text



Plain Text Biasa di gunakan untuk Meng inputkan String atau pun Number . Dan akan di proses .

```
<EditText
    android:id="@+id/editText1"
```



```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10" >

        <requestFocus />
    </EditText>
```

11. Linear Layout

XML yang digunakan untuk mengatur tata letak dari semua tools yang di masukkan. Biasa nya Orientation Horizontal dan Vertical.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

</LinearLayout>
```

12. Relative Layout



Relative Layout adalah ViewGroup yang menampilkan elemen View anak dalam posisi relatif. Posisi View dapat ditetapkan sebagai relatif terhadap elemen saudara (seperti ke-kiri atau di bawah elemen tertentu) atau dalam posisi relatif ke daerah RelativeLayout (seperti selaras ke bawah, kiri pusat) .

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

</RelativeLayout>
```

13. Frame Layout





FrameLayout dirancang untuk menampilkan satu item pada suatu waktu. Anda dapat memiliki beberapa elemen dalam FrameLayout tetapi setiap elemen akan diposisikan berdasarkan kiri atas layar. Elemen yang tumpang tindih akan ditampilkan tumpang tindih. Saya telah membuat tata letak XML sederhana menggunakan FrameLayout yang menunjukkan bagaimana ini bekerja.

```
<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
</FrameLayout>
```

14. Table Layout

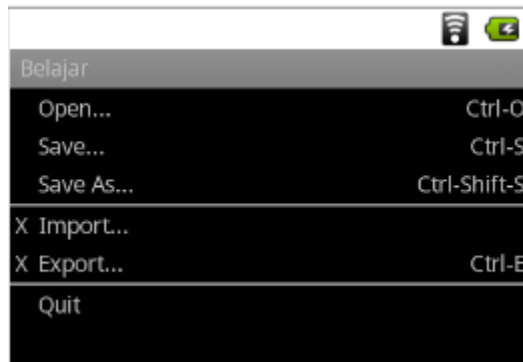


Table Layout di gunakan untuk memberi Layout dalam bentuk Table , Table layout sering di gunakan untuk mengatur Layar Layout yang akan di gunakan.

```
<TableLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

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

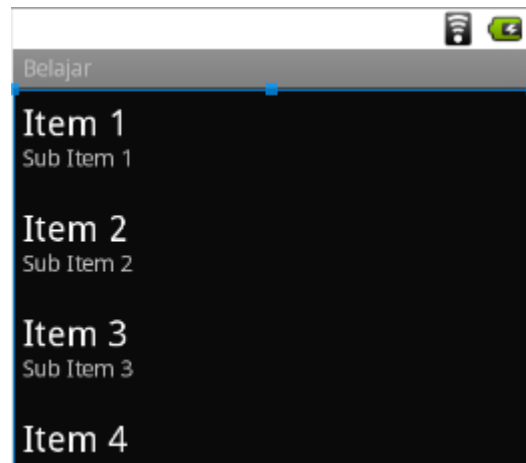
    <TableRow
        android:id="@+id/tableRow2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >
    </TableRow>

    <TableRow
        android:id="@+id/tableRow3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >
    </TableRow>

    <TableRow
        android:id="@+id/tableRow4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >
    </TableRow>
</TableLayout>
```



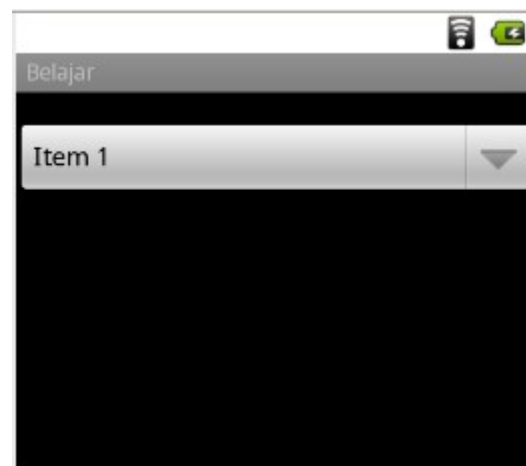
15. List View



List View Merupakan layout yang sering di gunakan , karena simple dan fungsi yang selalu di gunakan , dengan tampilan sejajar dan terurut.

```
<ListView
    android:id="@+id/listView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
</ListView>
```

16. Spinner

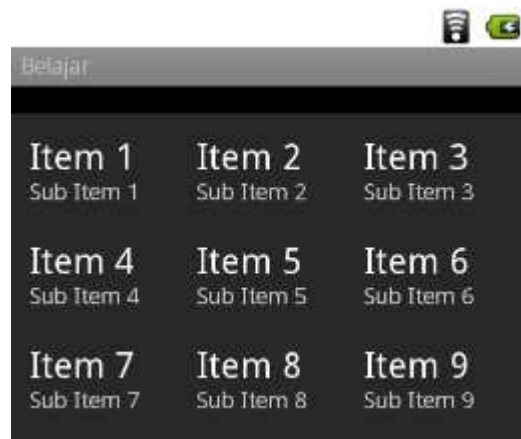


Spinner adalah sebuah widget mirip dengan daftar drop-down untuk memilih item.

```
<Spinner
    android:id="@+id/spinner1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```




17. Grid View



```
<GridView
    android:id="@+id/gridView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:numColumns="3" >
</GridView>
```

18. Sliding Drawer



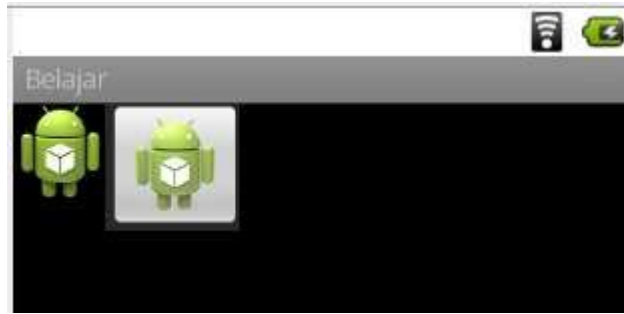
```
<SlidingDrawer
    android:id="@+id/slidingDrawer1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:content="@+id/content"
    android:handle="@+id/handle" >

    <Button
        android:id="@+id/handle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Handle" />

    <LinearLayout
        android:id="@+id/content"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >
    </LinearLayout>
</SlidingDrawer>
```



19. Images , Images Button



```
<ImageButton
    android:id="@+id/imageButton1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/ic_launcher" />

<ImageView
    android:id="@+id/imageView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/ic_launcher" />
```

20. Time And Date



```
<AnalogClock
    android:id="@+id/analogClock1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<Chronometer
    android:id="@+id/chronometer1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Chronometer" />

<DatePicker
    android:id="@+id/datePicker1"
```



```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

<DigitalClock
    android:id="@+id/digitalClock1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="DigitalClock" />

<TimePicker
    android:id="@+id/timePicker1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

```

Project Latihan

Project latihan ini berisi bab yang akan membuat suatu project untuk menghasilkan user interface yang kita inginkan , karena dalam sebuah tampilan pada aplikasi itu utama , oleh karena itu sebaiknya latihan ini dilakukan langsung ke lembar kerja dan dilakukan dengan menulis scrip tanpa copy paste , agar ini terlatih dan mengingat hasil pembelajaran latihan ini . Dari beberapa tampilan ini kita akan membuat se sederhana mungkin agar mudah untuk di pahami

1. Alerg Dialog (Perhatian / Warning)

Alerg Dialog merupakan suatu pesan yang di tampilkan pada layar kerja , tujuan adanya alerg dialog yaitu agar pengguna dapat mengetahui letak kesalahan , dalam alerg dialog ini banyak bentuk yang ada , kita akan membahas satu persatu.

1. Create Project dengan nama Alerg_Dialog
2. Buka File pada res – layout.xml
3. Tuliskan scrip di bawah ini.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Pesan" />

    <Button

```

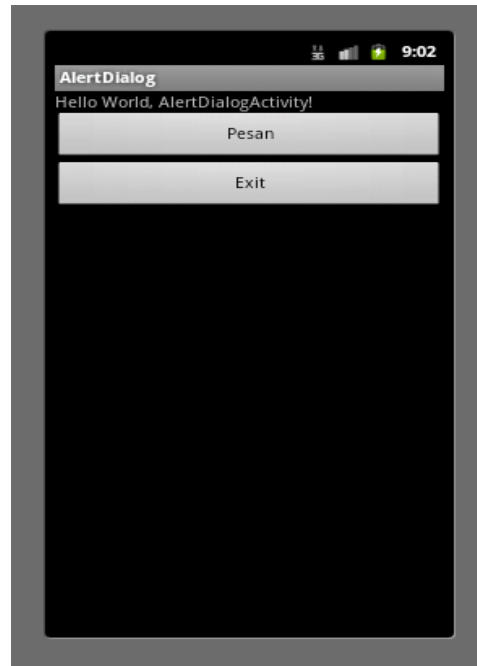


```

        android:id="@+id/btnExit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Exit" />

```

```
</LinearLayout>
```



4. Buka File pada src – package – Alerg_dialog.java dan tuliskan scrip di bawah ini.

```

package alertdialog.wartec;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.AlertDialog.Builder;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class AlertDialogActivity extends Activity implements
OnClickListener {
    /** Called when the activity is first created. */

    Button btn;
    Button exit;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        btn = (Button)findViewById(R.id.button1);
        btn.setOnClickListener(new OnClickListener() {

```




```

        public void onClick(View arg0) {
            // TODO Auto-generated method stub
            AlertDialog.Builder build = new
AlertDialog.Builder(AlertDialogActivity.this);
            build.setTitle("Message")
                .setIcon(R.drawable.ic_launcher)
                .setMessage("Hai")
                .setPositiveButton("OK",null)
                .show();
        }
    });

    exit = (Button)findViewById(R.id.btnExit);
    exit.setOnClickListener(new OnClickListener() {

        public void onClick(View arg0) {
            // TODO Auto-generated method stub
            AlertDialog.Builder build = new
AlertDialog.Builder(AlertDialogActivity.this);
            build
                .setMessage("apakah benar anda akan keluar")
                .setCancelable(false)
                .setPositiveButton("yes",new
DialogInterface.OnClickListener() {

                    public void onClick(DialogInterface arg0, int
arg1) {

                        // TODO Auto-generated method stub
                        System.exit(RESULT_OK);

                    }

                })
                .setNegativeButton("tidak",new
DialogInterface.OnClickListener() {

                    public void onClick(DialogInterface arg0, int
arg1) {

                        // TODO Auto-generated method stub
                        arg0.cancel();

                    }

                })
                .show();
        }

    });

}

//kita akan membuat method untuk menampilkan message alerdialog banya
bentuk yang di gunakan.

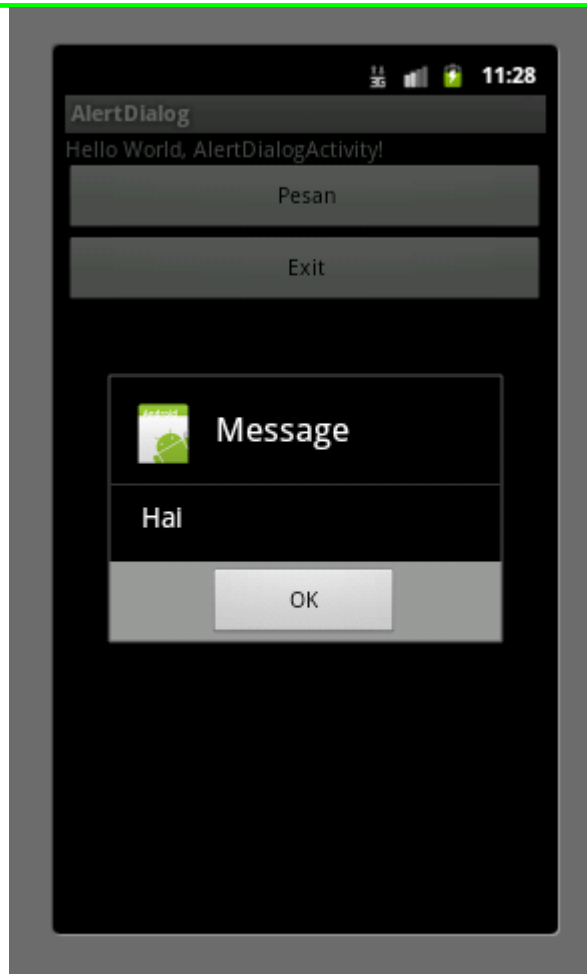
public void message(){
    AlertDialog.Builder build = new
AlertDialog.Builder(AlertDialogActivity.this);
    build.setTitle("Message")
        .setIcon(R.drawable.ic_launcher)
        .setMessage("Hai")
        .setPositiveButton("OK",null)
        .show();
}

```



```
public void onClick(View arg0) {
    // TODO Auto-generated method stub

}
```



2. Check Box

Check Box di bersifat true atau false , dan pada dasarnya check box sama seperti radio Button , bila media di centang maka bersifat true , dan bila media tidak di centrang maka bersifat false , biasanya di di lakukan pada aplikasi membutuhkan pilihan lebih dari satu.

1. Create Project dengan nama Check Box
2. Buka File res – layout – main.xml
3. Tuliskan Scrip di bawah ini.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

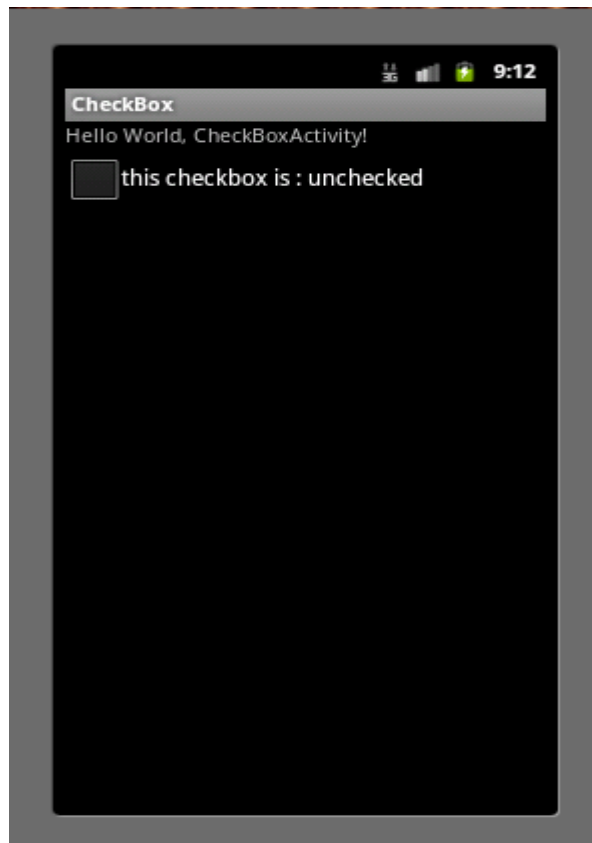
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
```



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

<CheckBox
    android:id="@+id/check"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="This checkbox is : unchecked" />

</LinearLayout>
```



4. Buka File pada src – package – Check_box.java

```
package checkbox.wartec;

import android.app.Activity;
import android.os.Bundle;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.CompoundButton.OnCheckedChangeListener;
//implement class is onchecngcheck

public class CheckBoxActivity extends Activity implements OnCheckedChangeListener{

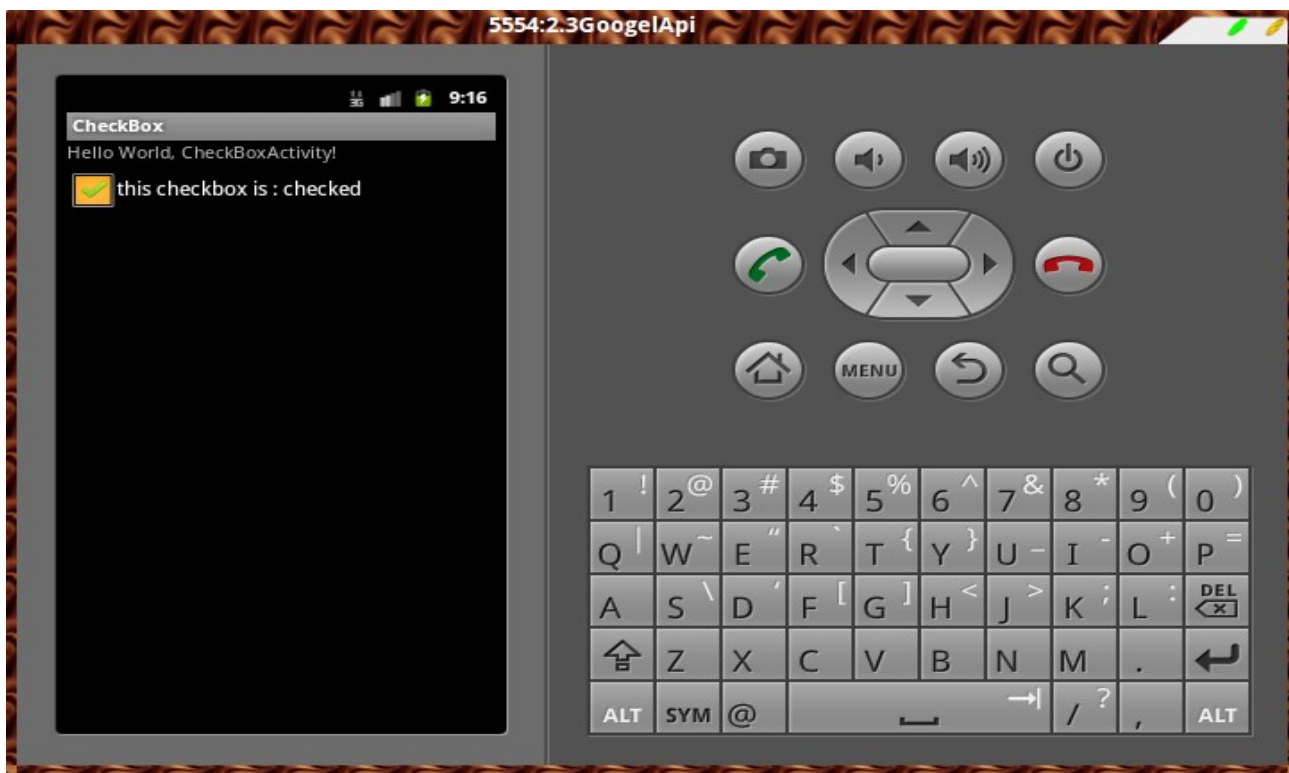
    /** Called when the activity is first created. */
```



```
//Decklare checkbox
CheckBox cb;
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    cb = (CheckBox)findViewById(R.id.check);
    cb.setOnCheckedChangeListener(this);
}

    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        // TODO Auto-generated method stub
        // Memberikan Opsion percabangan.
        if(isChecked){
            cb.setText("this checkbox is : checked");
        }else {
            cb.setText("this checkbox is : unchecked");
        }
    }
}
```



3. Download File By Progress

Progress sering kita temukan setiap kali kita akan memulai suatu permainan atau pun aplikasi , seprti pada game playstation yang sering kita mainkan , dalam tulisan itu sering bertulisan Loading , Tunggu sebentar , tunggu proses selesai , Downloading dan lain lain , itu juga ada pada android ,



tujuan dari progress adalah untuk membuat jeda aplikasi yang sedang berjalan dan mempersiapkan sistem yang akan bekerja selanjutnya.

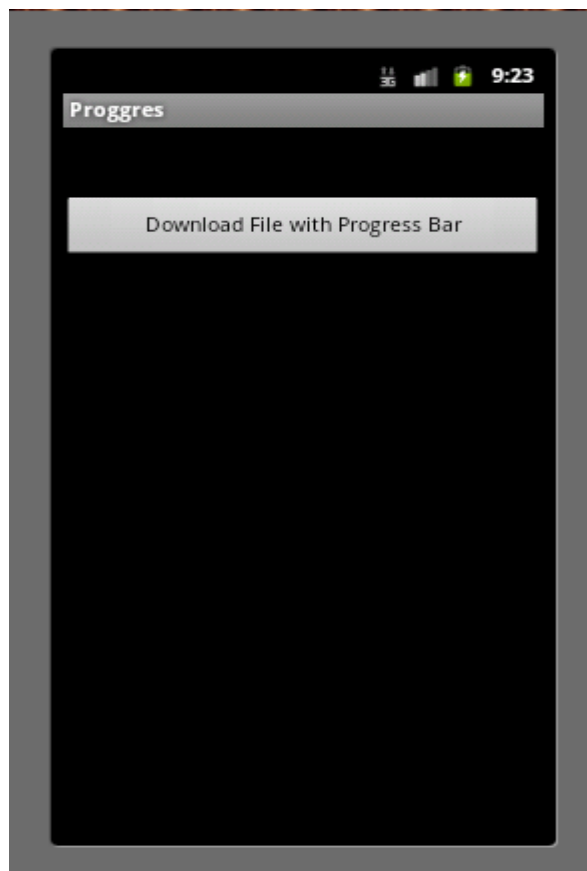
1. Create project dengan nama Progress
2. Buka file pada res – layout – main.xml
3. Tuliskan scrip di bawah ini

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <!-- Download Button -->
    <Button android:id="@+id/btnProgressBar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Download File with Progress Bar"
        android:layout_marginTop="50dip"/>

    <!-- Image view to show image after downloading -->
    <ImageView android:id="@+id/my_image"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"/>

</LinearLayout>
```



4. Buka file pada src – package – Progress.java



```
package progress.wartec;

import java.io.BufferedReader;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.URL;
import java.net.URLConnection;

import android.app.Activity;
import android.app.Dialog;
import android.app.ProgressDialog;
import android.graphics.drawable.Drawable;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Environment;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;

public class ProgresActivity extends Activity {
    // button to show progress dialog
    Button btnShowProgress;

    // Progress Dialog
    private ProgressDialog pDialog;
    ImageView my_image;
    // Progress dialog type (0 - for Horizontal progress bar)
    public static final int progress_bar_type = 0;

    // File url to download
    private static String file_url = "http://api.androidhive.info/progressdialog/hive.jpg";

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        // show progress bar button
        btnShowProgress = (Button) findViewById(R.id.btnProgressBar);
        // Image view to show image after downloading
        my_image = (ImageView) findViewById(R.id.my_image);
        /**
         * Show Progress bar click event
         */
        btnShowProgress.setOnClickListener(new View.OnClickListener() {

            public void onClick(View v) {
                // starting new Async Task
            }
        });
    }
}
```



```

        new DownloadFileFromURL().execute(file_url);
    }
});
}
/**
 * Showing Dialog
 * */
@Override
protected Dialog onCreateDialog(int id) {
    switch (id) {
        case progress_bar_type:
            pDialog = new ProgressDialog(this);
            pDialog.setMessage("Downloading file. Please wait...");
            pDialog.setIndeterminate(false);
            pDialog.setMax(100);
            pDialog.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
            pDialog.setCancelable(true);
            pDialog.show();
            return pDialog;
        default:
            return null;
    }
}

/**
 * Background Async Task to download file
 * */
class DownloadFileFromURL extends AsyncTask<String, String, String> {

    /**
     * Before starting background thread
     * Show Progress Bar Dialog
     * */
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        showDialog(progress_bar_type);
    }

    /**
     * Downloading file in background thread
     * */
    @Override
    protected String doInBackground(String... f_url) {
        int count;
        try {
            URL url = new URL(f_url[0]);
            URLConnection conection = url.openConnection();
            conection.connect();

```



```
// getting file length
int lenghtOfFile = conection.getContentLength();

// input stream to read file - with 8k buffer
InputStream input = new BufferedInputStream(url.openStream(), 8192);

// Output stream to write file
OutputStream output = new FileOutputStream("/sdcard/downloadedfile.jpg");

byte data[] = new byte[1024];

long total = 0;

while ((count = input.read(data)) != -1) {
    total += count;
    // publishing the progress....
    // After this onProgressUpdate will be called
    publishProgress(""+(int)((total*100)/lenghtOfFile));

    // writing data to file
    output.write(data, 0, count);
}

// flushing output
output.flush();

// closing streams
output.close();
input.close();

} catch (Exception e) {
    Log.e("Error: ", e.getMessage());
}

return null;
}

/**
 * Updating progress bar
 * */
protected void onProgressUpdate(String... progress) {
    // setting progress percentage
    pDialog.setProgress(Integer.parseInt(progress[0]));
}

/**
 * After completing background task
 * Dismiss the progress dialog
 * */
```

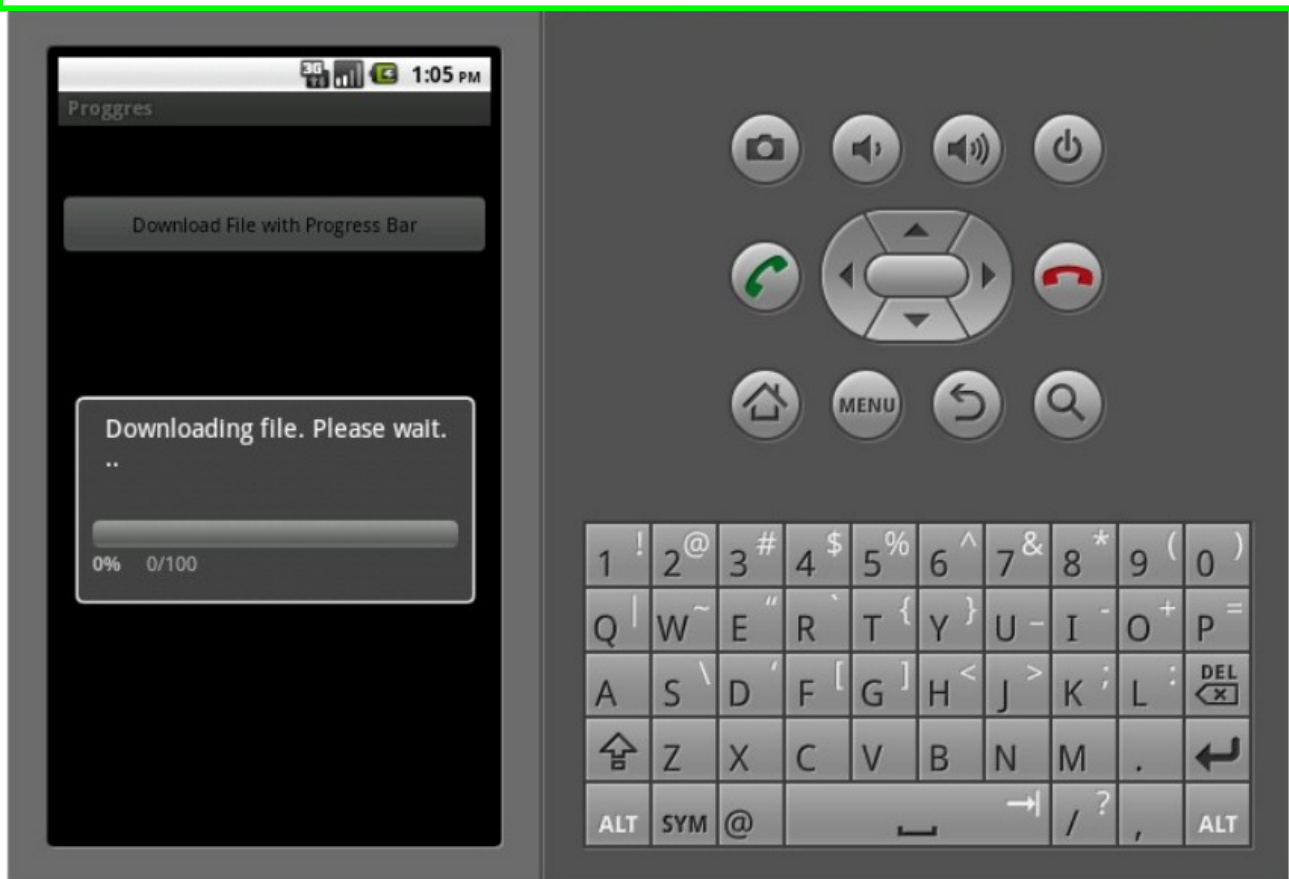



```

@Override
protected void onPostExecute(String file_url) {
    // dismiss the dialog after the file was downloaded
    dismissDialog(progress_bar_type);

    // Displaying downloaded image into image view
    // Reading image path from sdcard
    String imagePath = Environment.getExternalStorageDirectory().toString() +
"/downloadedfile.jpg";
    // setting downloaded into image view
    my_image.setImageDrawable(Drawable.createFromPath(imagePath));
}
}
}

```



4. Dua Activity

Pada Materi Dua Activity ini merupakan dua Aktivitas yang lain , bisa di bayangkan seperti dua Ruang yang isi berbeda , yang satu ruang Tamu dan yang satu ruang tidur , Prinsip pada dua activity ini merupakan pemanggilan dua class , yang satu dapat memanggil class yang satu nya dan juga sebalik nya . Logika dalam dua activity ini ada dua tampilan dan ada dua class main . Sehingga dua dua nya dapat berhubungan dengan di satukan dalam Scrip xml androidmanifest.

1. Create Project dengan nama Dua_Aktivitiy



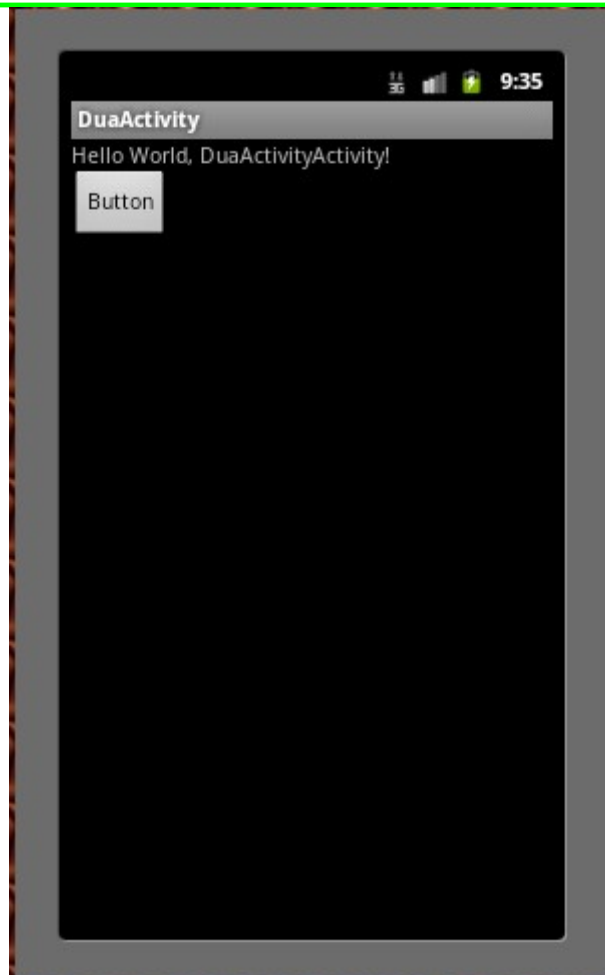
2. Buka file pada res – layout – main.xml
3. tuliskan scrip di bawah ini

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button" />

</LinearLayout>
```



4. Kita akan membuat Tampilan Kedua . File – new file – android xml dan bernama child dan tuliskan scrip di bawah ini.

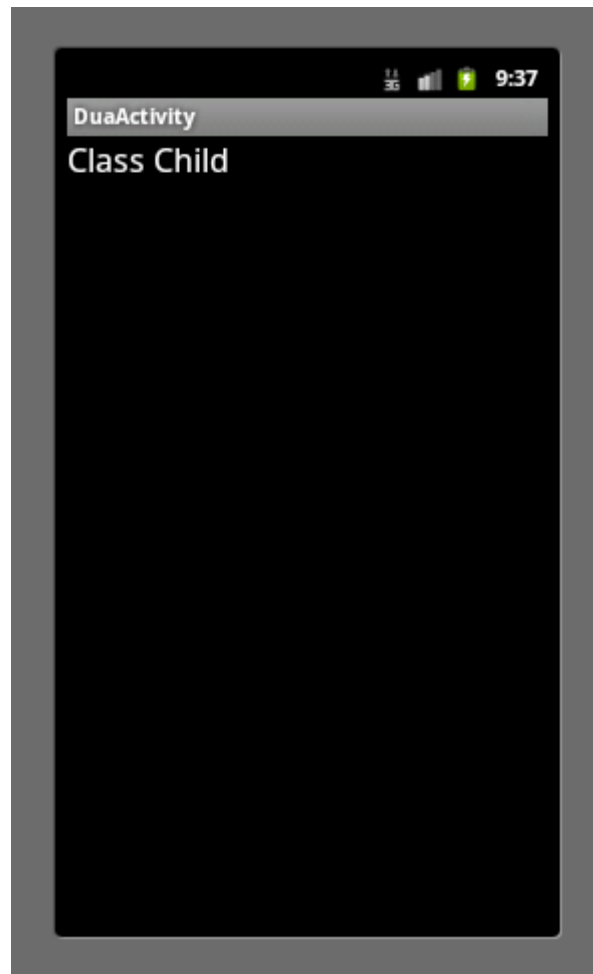
```
<?xml version="1.0" encoding="utf-8"?>
```



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

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Class Child"
        android:textAppearance="?android:attr/textAppearanceLarge" />

</LinearLayout>
```



5. buka file pada src – package – Dua_Activity.java dan tuliskan scrip di bawah ini

```
package duaactivity.wartec;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
```



```
import android.widget.Button;

public class DuaActivityActivity extends Activity implements OnClickListener{
    /** Called when the activity is first created. */

    Button btn;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        btn = (Button)findViewById(R.id.button1);
        btn.setOnClickListener(this);
    }

    //method untuk menampilkan aktivitas kedua yaitu child.class
    public void onClick(View arg0) {
        // TODO Auto-generated method stub
        Intent i = new Intent(DuaActivityActivity.this , Child.class);

        startActivity(i);
    }
}
```

6. Kita akan membuat class kembali dengan tujuan dari tampilan Child.xml File – new file – class dan tuliskan scrip di bawah ini.

```
package duaactivity.wartec;

import android.app.Activity;
import android.os.Bundle;

public class Child extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.child);
    }
}

//setelah membuat semua class dan masing - masing layout kita akan mendeklarasikan class Child
dalam androidmanifest
```




7. Buka file androidmanifest.xml , tujuan ini kita akan menghubungkan class java tersebut dengan device android . Atau sering di sebut juga pendeklerasian class. Dan tuliskan scrip di bawah ini.

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

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

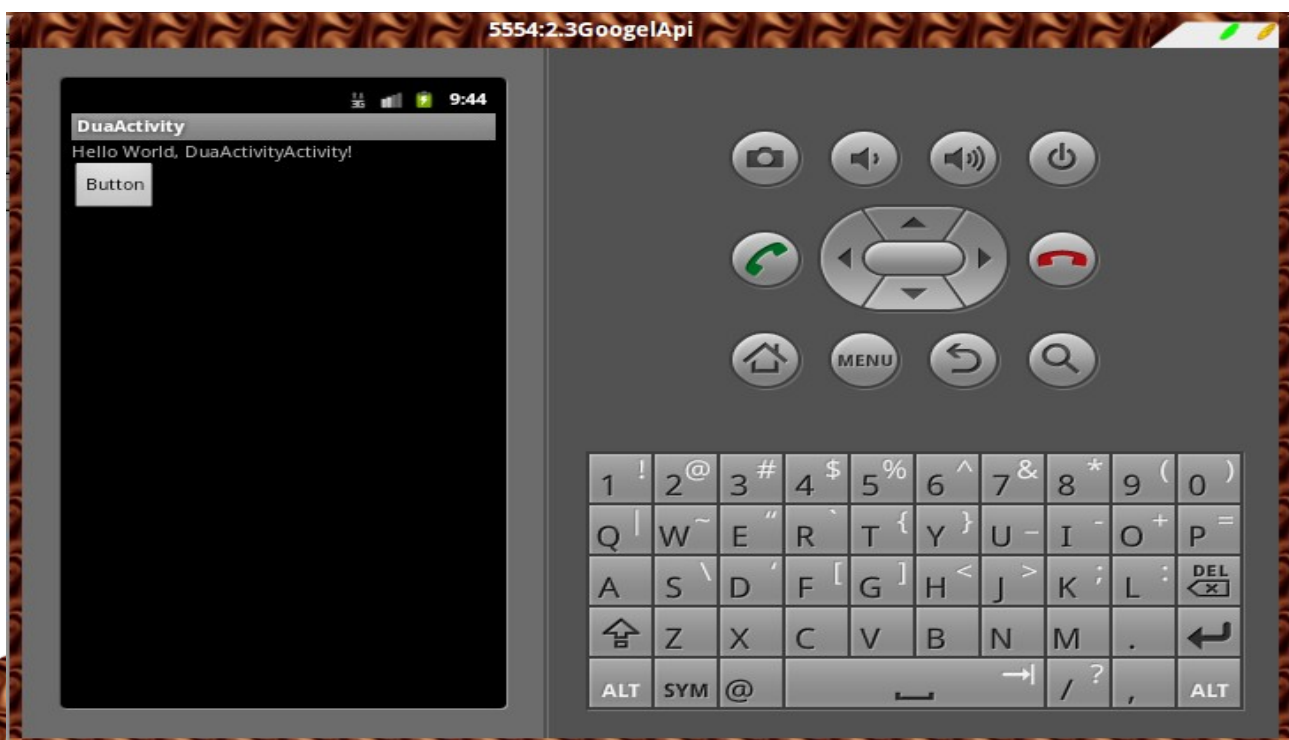
    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".DuaActivityActivity" <!--Nama class pertama-->
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

        <activity
            android:name=".Child" <!--Nama class kedua-->
            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>

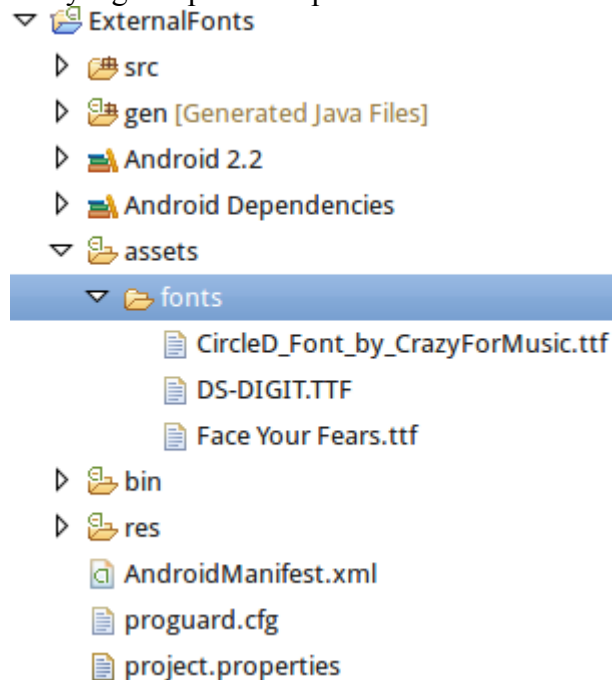
</manifest>
```





Bab kali ini kita akan membahas external font , external font merupakan font yang tidak ada pada font yang di berikan platform android , kita di sini memasukkan font pada suatu project dan memanggil nya , pada dasar nya font ini dapat di gunakan pada font font biasanya , karena font ini juga font yang sering kita jumpai di tampilan tampilan font yang di buat,

1. Create Project dan beri nama Exernal_Font
2. Create Folder dan beri nama folder Fonts
3. Masukkan Copy Font yang ada pada computer .



4. Buka File pada
res – layout –

main.xml dan tulis kan scrip di bawah ini :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/wartec"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:textSize="70dp"
        android:layout_marginTop="50dp"
        android:textColor="#ef0000"
        android:text="wartec"
        android:textAppearance="?android:attr/textAppearanceLarge" />

</LinearLayout>
```



5. Buka file pada src – package – External_Font.java dan tuliskan scrip di bawah ini

```
package ExternalFonts.wartec;

import android.app.Activity;
import android.graphics.Typeface;
import android.os.Bundle;
import android.widget.TextView;

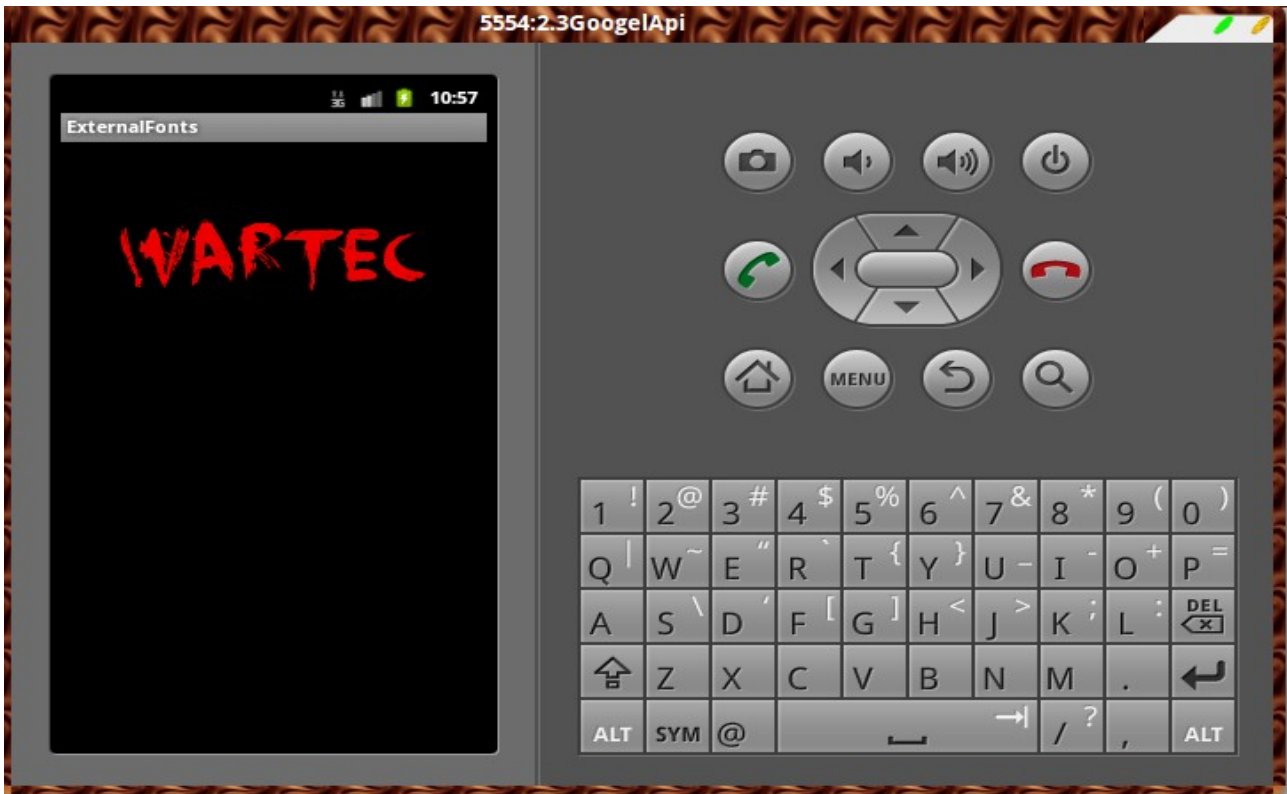
public class ExternalFontsActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        String fontpath = "fonts/Face Your Fears.ttf"; // pendeklarasian alamat font

        TextView txtWartec= (TextView)findViewById(R.id.wartec);

        Typeface tf = Typeface.createFromAsset(getAssets(),fontpath); // memanggil font pada
acktifitas typeface

        txtWartec.setTypeface(tf);
    }
}
```



6. Grid View

Grid View . Tampilan dalam bentuk grid , grid view memiliki tempat yang baik dalam user interface , tampilan grid ini seperti matrik tetapi kita dapat membuat bentuk yang sesuai dengan yang kita masukkan. Secara otomatis akan membuat bentuk sendiri.

1. Create Project dengan nama Grid_View
2. Buka file pada res – layout – main.xml dan tuliskan scrip di bawah ini :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/selection"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"/>

    <GridView
        android:id="@+id/gridView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:numColumns="3" >
    </GridView>

</LinearLayout>
```




3. Buka File pada src – project – Grid_view.java dan tuliskan scrip di bawah ini :

```
package gridview.wartec;

import android.app.Activity;
import android.content.Context;
import android.os.Bundle;
import android.view.View;
import android.view.animation.GridLayoutAnimationController;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.GridView;
import android.widget.TextView;

//sebelumnya kita memasukkan implement kepada class activity
public class GridViewActivity extends Activity implements OnItemClickListener {
    /** Called when the activity is first created. */

    TextView selection;
    GridView g;

    //membuat array untuk memasukkan nilai pada layout

    String [] items = {"dede","mamas","bapak","ibu","ryan","pakde","bude"};

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        selection = (TextView)findViewById(R.id.selection);
        g = (GridView)findViewById(R.id.gridView1);
```



```
g.setAdapter(new FunnyAdapter(this , android.R.layout.simple_list_item_1,items));
g.setOnItemClickListener(this);
}

//Masih terjadi error pada FunnyAdapter

// kita akan membuat class Abstrac dan di implement kan ArrayAdapter
public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
    long arg3) {
    // TODO Auto-generated method stub
    selection.setText(items[arg2]);
}

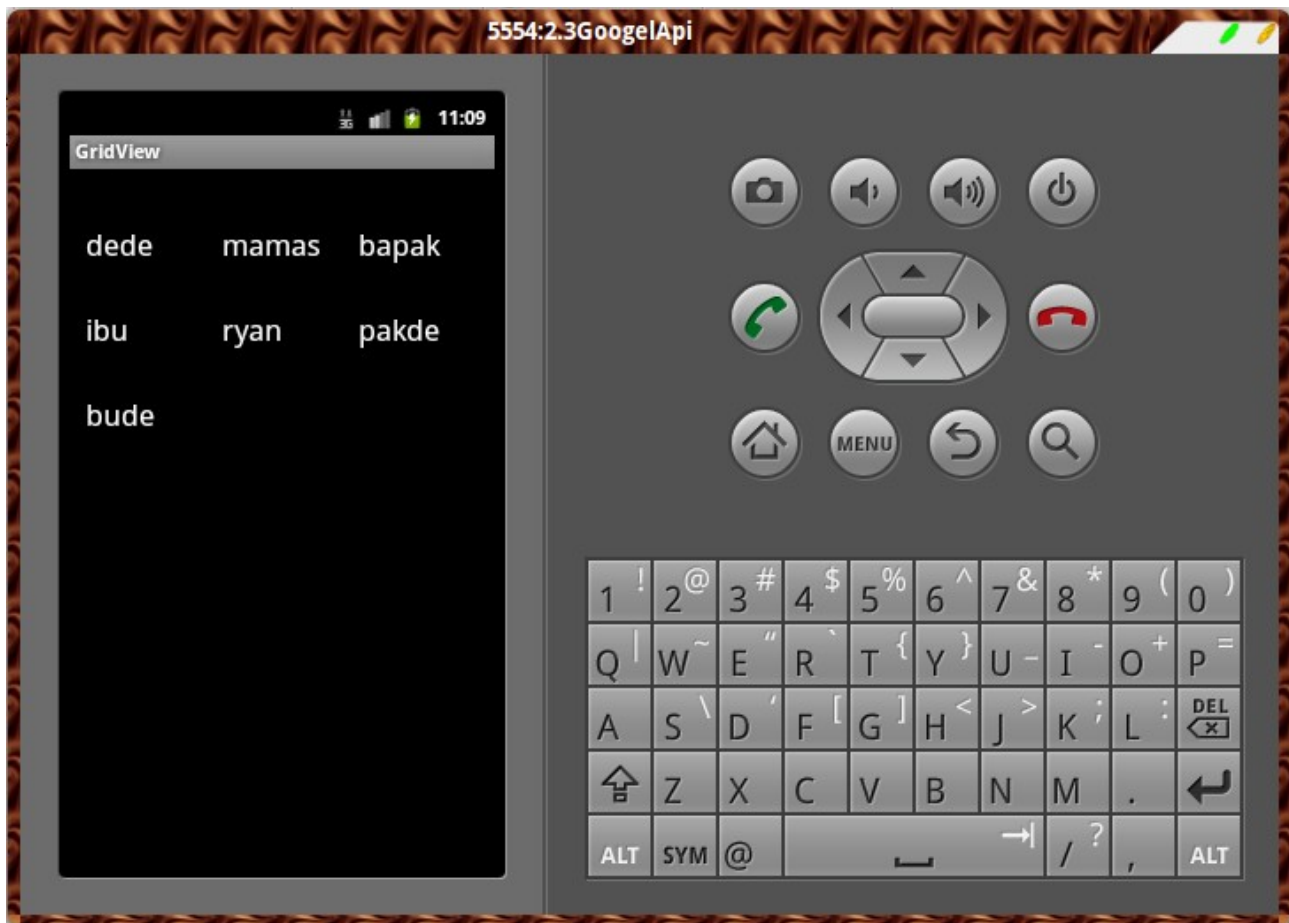
public void onNothingSelected(AdapterView<?> arg0) {
    // TODO Auto-generated method stub
    selection.setText("");
}

public class FunnyAdapter extends ArrayAdapter{

    private Context context;

    public FunnyAdapter(Context context, int textViewResourceId , String [] items) {
        super(context, textViewResourceId,items);
        // TODO Auto-generated constructor stub

        this.context = context;
    }
}
```



7. List View

List View merupakan tampilan berbentuk list list secara horizontal , dalam list view ini kita dapat menampilkan dan memberikan eksekusi pada list view tersebut , sama seperti aktifitas lainnya , bila di klik akan bersifat true dan bila tanpa di klik bersifat false.

1. Create project dengan nama List_View
2. Buka File pada res – layout – main.xml
3. Tulis scrip di bawah ini

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/selection"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"/>

    <ListView
        android:id="@android:id/list"
```



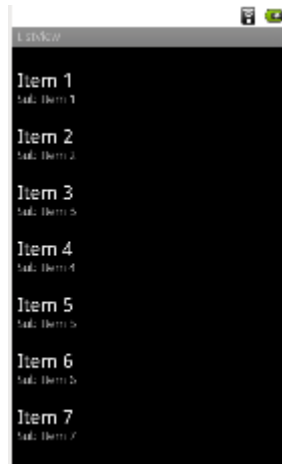
```

        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:drawSelectorOnTop="false"></ListView>

</LinearLayout>

```

Sehinga tampilan dari scrip di atas seperti ini :



4. Buka file pada src – package – ListViewActivity.java dan tuliskan scrip di bawah ini :

```

package listview.wartec;

import android.app.Activity;
import android.app.ListActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;

public class ListViewActivity extends ListActivity {
    /** Called when the activity is first created. */
    TextView selection;

    String[]
    items={"dede","wawan","ryan","bapak","ibu","Keluarga","Bude","pakde","Mbah","Mbok"};
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        setListAdapter(new ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_1,
            items));
        selection=(TextView)findViewById(R.id.selection);
    }
    public void onItemClick(ListView parent, View v, int position,

```

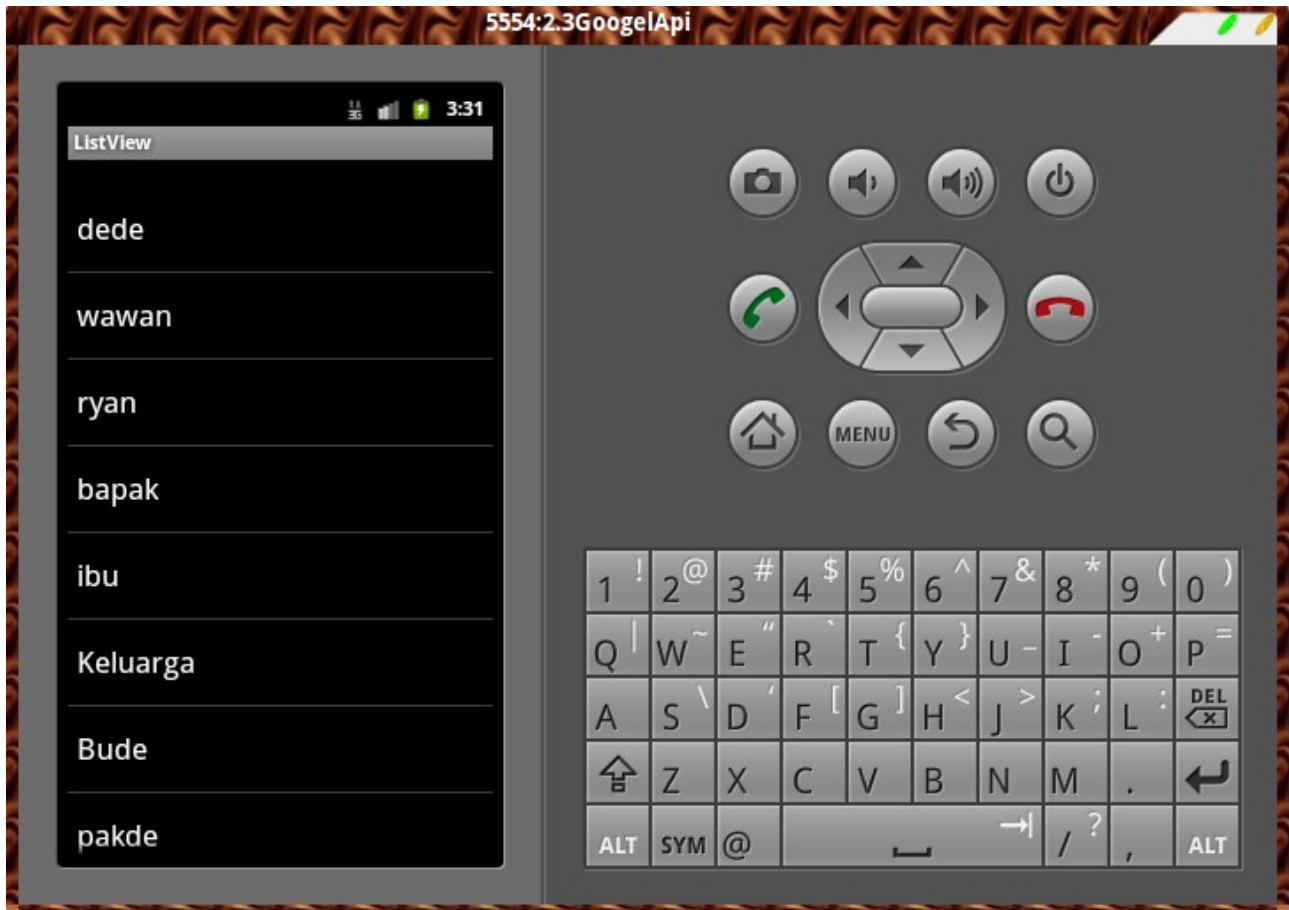



```

        long id) {
            selection.setText(items[position]);
        }
    }

```

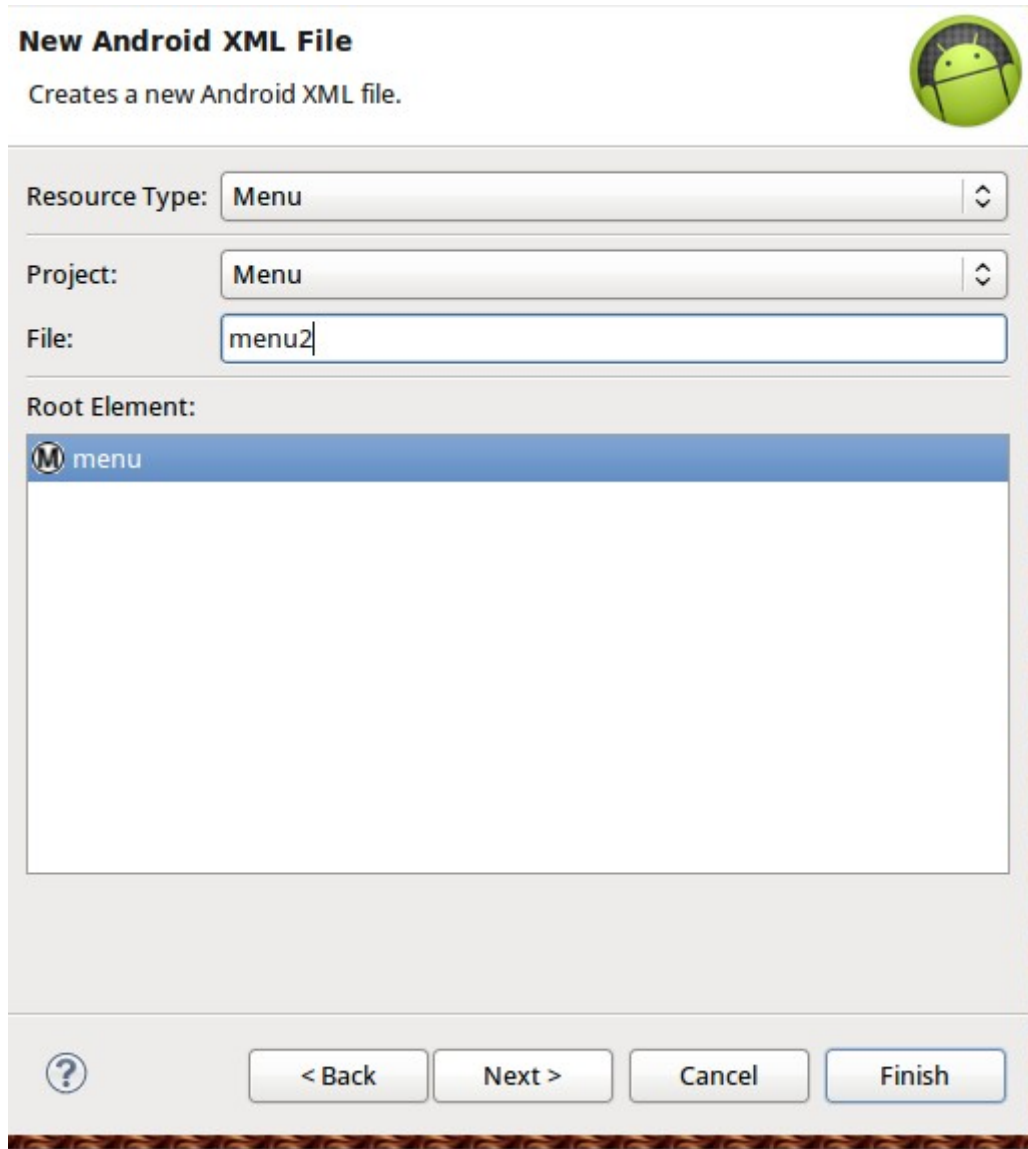
Jalankan project dan hasil nya seperti di bawah ini :



8. Menu

Sama seperti menu dalam sebuah aplikasi mobile lainnya , menu ini menunjukkan pada beberapa pilihan , menu dalam sebuah aplikasi sangat di butuhkan , dengan beberapa tema pasti memiliki menu tersendiri , biasa nya dalam menu sebuah aplikasi berupa pilihan yang tidak terlalu di lihat kan , seperti about , atau pun help , tetapi ini merupakan user interface yang di butuhkan .

1. Create Project dengan nama Menu
2. Klik kanan pada sebuah folder res dan pilih new – android xml



3. Kita akan memasukkan perintah menu menu apa saja yang akan di tampilkan: tulis scrip di bawah ini pada xml menu :

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
    <item
        android:id="@+id/about"
        android:title="About"
    />
    <item
        android:id="@+id/exit"
        android:title="Exit"/>
</menu>
```

4. Buka file pada src – package – MenuActivity.java dan tuliskan scrip di bawah ini :

```
package menu.wartec;

import android.app.Activity;
```



```
import android.media.ExifInterface;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;

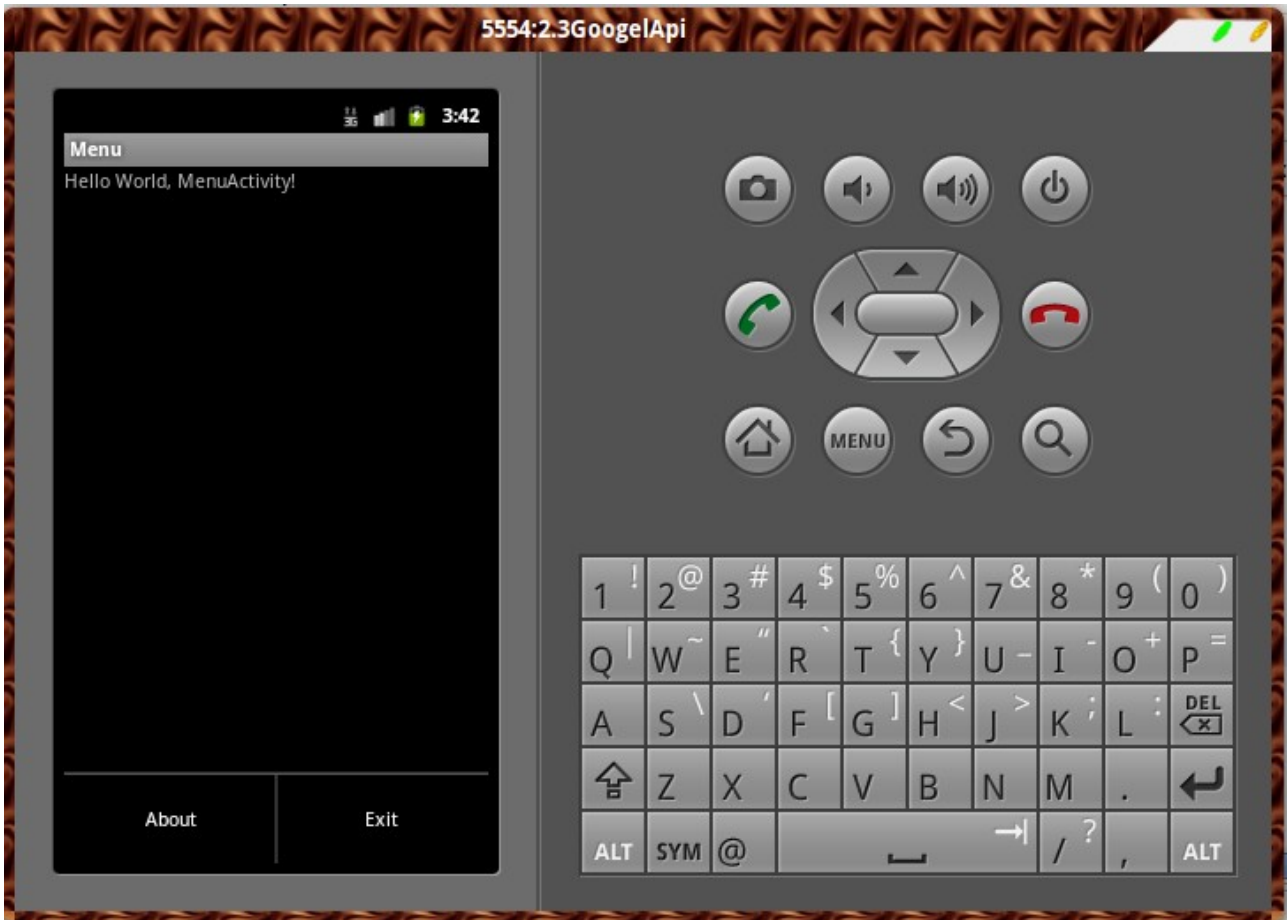
public class MenuActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

    }
    //pembuatan method inflater menu.
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.menu, menu);
        return true;
    }
    //pembuatan method pemanggilan menu
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle item selection
        switch (item.getItemId()) {
            case R.id.about:
                about();
                return true;
            case R.id.exit:
                exit();
                return true;

            default:
                return super.onOptionsItemSelected(item);
        }
    }
    //Method exit
    private void exit() {
        // TODO Auto-generated method stub
    }
    //method about
    private void about() {
        // TODO Auto-generated method stub
    }
}
```



Dan jalankan project hasil nya akan seperti di bawah ini :



9. Scrol View

Scrol View secara otomatis akan membuat pada device bila sebuah object yang dimasukkan terlalu banyak atau melebihi batas , scrol view sama seperti Linear layout , yang membedakan kalau linear layout hanya layout default tanpa ada embel embel lainnya , seperti ruang yang telah ditentukan , bila scrol view ini , sebuah ruang tetapi bila ruang itu terisi dengan banyak tempat atau penuh maka dia akan otomatis menambah ruang tersebut dengan sendirinya . Sebagai contoh lihat di bawah ini :

1. Create Project dengan nam ScrolView
2. Buka file pada res – layout dan main.xml dan tuliskan scrip di bawah ini :

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TableLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:stretchColumns="0">
```




```
<TableRow >
    <View
        android:layout_height="80px"
        android:background="#52e236"/>

    <EditText
        android:text="#52e236"
        android:paddingLeft="4px"
        android:layout_gravity="center_vertical"/>
</TableRow>

<TableRow >
    <View
        android:layout_height="80px"
        android:background="#08dfff"/>

    <EditText
        android:text="#08dfff"
        android:paddingLeft="4px"
        android:layout_gravity="center_vertical"/>
</TableRow>

<TableRow >
    <View
        android:layout_height="80px"
        android:background="#fcca00"/>

    <EditText
        android:text="#fcca00"
        android:paddingLeft="4px"
        android:layout_gravity="center_vertical"/>
</TableRow>

<TableRow >
    <View
        android:layout_height="80px"
        android:background="#646464"/>

    <EditText
        android:text="#646464"
        android:paddingLeft="4px"
        android:layout_gravity="center_vertical"/>
</TableRow>

<TableRow >
    <View
        android:layout_height="80px"
        android:background="#ce0ef7"/>

    <EditText
```



```

                                android:text="#ce0ef7"
                                android:paddingLeft="4px"
android:layout_gravity="center_vertical"/>

                                </TableRow>

                                <TableRow >
                                    <View
                                        android:layout_height="80px"
                                        android:background="#fd2603"/>

                                    <EditText
                                        android:text="#fd2603"
                                        android:paddingLeft="4px"

                                </TableRow>

                                <TableRow >
                                    <View
                                        android:layout_height="80px"
                                        android:background="#fcff08"/>

                                    <EditText
                                        android:text="#fcff08"
                                        android:paddingLeft="4px"

                                </TableRow>

                                </TableLayout>
</ScrollView>

```

3. dan tampilan akan seperti ini :





10.Set Date And Time

Tanggal dan waktu pada sebuah layar desktop sangat kita kenal , apalagi dengan orang yang sibuk pasti sangat membutuhkan jam dan tanggal pada sebuah layar desktop pada mobile atau pun yang alinnya kali ini kita akan membuat suatu aplikasi yang berhubungan dengan tanggal dan jam : langsung ke lembar kerja :

1. Create Project dengan nama DateAndTime
2. Buka file pada res – layout – main.xml untuk membuat tampilan
3. tuliskan scrip di bawah ini :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/dateAndTime"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"/>

    <Button
        android:id="@+id/btnDate"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Set Date" />

    <Button
        android:id="@+id/btnTime"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="set Time" />

</LinearLayout>
```

Hasil dari scrip di atas akan menampilkan serti di bawah ini :



4. Buka file pada src – package – DateAndTimeActivity.java kita disini akan membuat suatu logika untuk mengatur jam dan tanggal pada layar desktop mobile kita :

```
package setDateAndTime.wartec;

import java.util.Calendar;
import android.app.Activity;
import android.app.DatePickerDialog;
import android.app.TimePickerDialog;
import android.os.Bundle;
import android.text.format.DateFormat;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.TextView;
import android.widget.TimePicker;

public class SetDateAndTimeeActivity extends Activity {
    //deklarasi date and time serta button yang akan di gunakan

    DateFormat formatDateAndTime;
    TextView dateAndTimeLabel;
```




```
Calendar dateAndTime=Calendar.getInstance();

Button btnDate;
Button btnTime;

//merupakan method date yang berdasarkan tahun tanggal dan bulan
DatePickerDialog.OnDateSetListener d = new DatePickerDialog.OnDateSetListener() {

    public void onDateSet(DatePicker arg0, int year, int month, int day) {
        // TODO Auto-generated method stub
        dateAndTime.set(Calendar.YEAR,year);
        dateAndTime.set(Calendar.MONTH,month);
        dateAndTime.set(Calendar.DAY_OF_MONTH,day);

        //kita masukkan pada method date and time
        updateLabel();
    }
};

//kali ini kita akan membuat method untuk time

TimePickerDialog.OnTimeSetListener t = new TimePickerDialog.OnTimeSetListener() {

    public void onTimeSet(TimePicker arg0, int hour, int minute) {
        // TODO Auto-generated method stub

        dateAndTime.set(Calendar.HOUR_OF_DAY,hour);
        dateAndTime.set(Calendar.MINUTE,minute);
        updateLabel();
    }
};

//sebelumnya kita terlebih dahulu membuat UpdateLabel

private void updateLabel(){
    dateAndTimeLabel.setText((CharSequence) formatDateAndTime);
}

/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    btnDate = (Button)findViewById(R.id.btnDate);
    btnDate.setOnClickListener(new View.OnClickListener() {

        public void onClick(View arg0) {
            // TODO Auto-generated method stub
```



```

        new DatePickerDialog(SetDateAndTimeActivity.this,d,
            dateAndTime.get(Calendar.YEAR),
            dateAndTime.get(Calendar.MONTH),

            dateAndTime.get(Calendar.DAY_OF_MONTH)).show();
        }
    });

    btnTime = (Button)findViewById(R.id.btnTime);
    btnTime.setOnClickListener(new View.OnClickListener() {

        public void onClick(View arg0) {
            // TODO Auto-generated method stub
            new TimePickerDialog(SetDateAndTimeActivity.this,t,
                dateAndTime.get(Calendar.HOUR_OF_DAY),
                dateAndTime.get(Calendar.MINUTE),true).show();

        }

    });

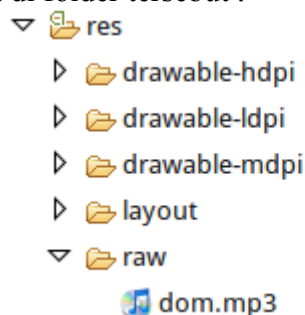
    dateAndTimeLabel = (TextView)findViewById(R.id.dateAndTime);
    updateLabel();
}
}

```

11.Sound

Suara a , I , u , e , o sangat lah sering kita dengar pada kehidupan sehari hari , karena pada setiap device biasa nya selalu ada spiker , pasti ada sound yang dapat kita masukan , itu benar . Dalam sebuah aplikasi sound sangat dibutuhkan untuk membuat suatu aplikasi itu lebih menarik , jadi sound pada sebuah platform android berupa format mp3,wav,ogg ,midi dan yang lainnya . Sekarang kita akan mempraktekan bagaimana membuat sound pada android :

1. Create project dengan nama sound
2. Buat Folder pada res dengan nama raw
3. masukkan lagu anda di folder tersebut .





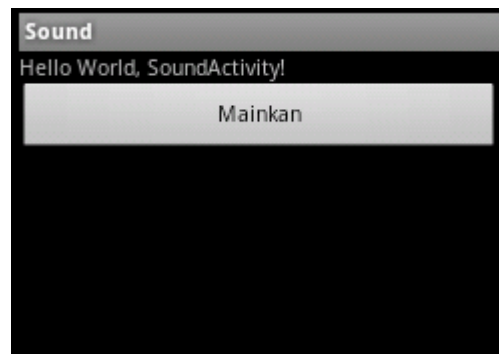
4. Sekarang kita akan membuat tampilan untuk mengeksekusi , buka file pada res – layout – main.xml dan tuliskan scrip di bawah ini :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Mainkan" />

</LinearLayout>
```



5. Sekarang kita akan membuat penghubung antara file lagu dengan button yang telah kita buat .buka file pada src – package – SoundActivity.java

```
package sound.wartec;

import android.app.Activity;
import android.media.MediaPlayer;
import android.media.MediaPlayer.OnCompletionListener;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class SoundActivity extends Activity implements OnClickListener {
    /** Called when the activity is first created. */
    Button btn;

    @Override
```



```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    btn = (Button)findViewById(R.id.button1);
    btn.setOnClickListener(new OnClickListener() {

        public void onClick(View arg0) {
            // TODO Auto-generated method stub
            btn.setEnabled(false);
            main(); // kita akan membuat method main untuk mendeklarasikan lagu pada folder
raw yang sebelumnya sudah di bat
        }

    });
}

public void main(){
    MediaPlayer main = MediaPlayer.create(this,R.raw.dom);
    try {
        main.prepare();
    } catch (Exception e) {
        // TODO: handle exception
    }

    main.start();
    main.setOnCompletionListener(new OnCompletionListener() {

        public void onCompletion(MediaPlayer arg0) {
            // TODO Auto-generated method stub
            btn.setEnabled(true);
        }

    });
}

public void onClick(View arg0) {
    // TODO Auto-generated method stub

}
}

```

12.Spinner

Spinner sama seperti list view tapi ini dalam bentuk pesan , seperti alerdialog yang sebelumnya kita buat , spinner juga sangat di butuh kan karena pesan itu berbentuk list dan bisa di masukkan banyak pilihan , oke kita langsung ke layar kerja :

1. Create Project dengan nama Spinner



2. Sebelumnya kita akan membuat tampilan terlebih dahulu , buka file pada res – layout – main.xml

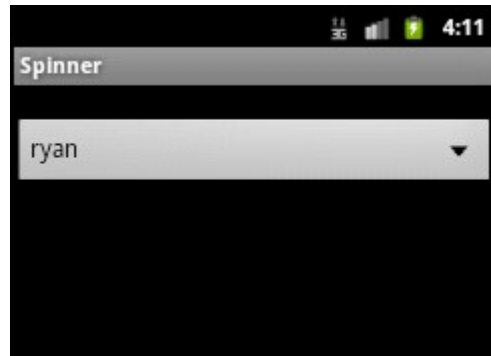
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/selection"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"/>

    <Spinner
        android:id="@+id/spinner"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"/>

</LinearLayout>
```

Dan tampilan nya akan seperti ini :



3. Buka file pada src – package – Spinner.java

```
package spinner.wartec;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.TextView;

public class SpinnerActivity extends Activity implements OnItemSelectedListener {
    /** Called when the activity is first created. */
    TextView selection;
    Spinner spin;

    String [] items = {"dede","mamas","ryan","ibu","bapak"};

    @Override
```



```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    //Membuat hubungan antara Layout dengan Activity java
    selection = (TextView)findViewById(R.id.selection);
    spin = (Spinner)findViewById(R.id.spinner);

    ArrayAdapter<String> aa = new ArrayAdapter<String>(this,
    android.R.layout.simple_spinner_item,items);

    aa.setDropDownViewResource(android.R.layout.simple_spinner_item);

    spin.setAdapter(aa);
}

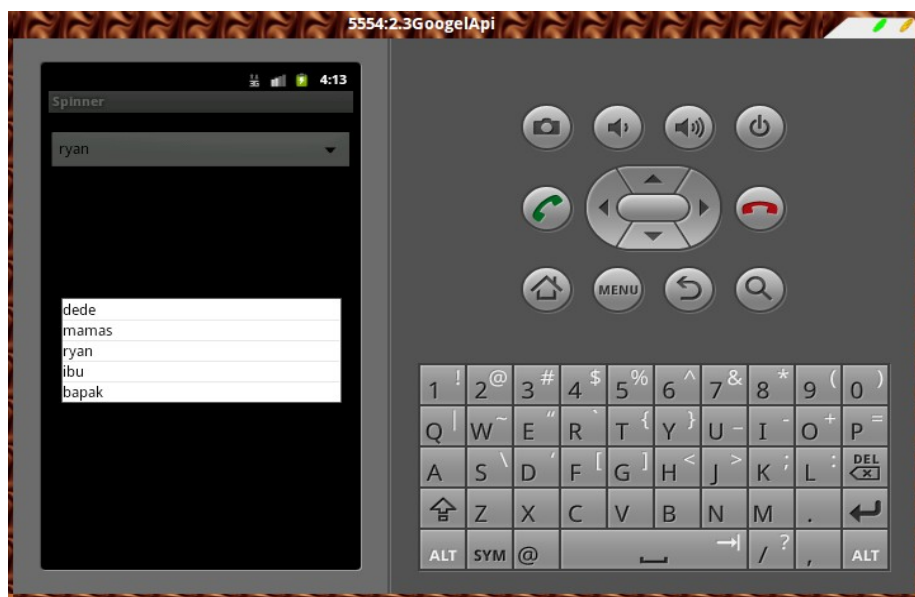
public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
    long arg3) {
    // TODO Auto-generated method stub

    selection.setText(items[arg2]);
}

public void onNothingSelected(AdapterView<?> arg0) {
    // TODO Auto-generated method stub
    selection.setText("");
}
}

```

Dan run project hasilnya akan seperti ini :





13. Tab Host

TabHost merupakan tampilan berupa tab tab pilihan , basic nya sama seperti menu tetapi ini langsung tampil pada layar kerja , Kita langsung ke tkp aja :

1. Create Project dengan nama TabHost
2. Buka file pada res – layout – main.xml (Untuk membuat tampilan)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

    <TabHost
        android:id="@+id/tabhost"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <TabWidget
            android:id="@android:id/tabs"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >
        </TabWidget>

        <FrameLayout
            android:id="@android:id/tabcontent"
            android:layout_width="match_parent"
            android:layout_height="match_parent" >

            <LinearLayout
                android:id="@+id/tab1"
                android:layout_width="match_parent"
                android:layout_height="match_parent" >
                <AnalogClock
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                />
            </LinearLayout>

            <LinearLayout
                android:id="@+id/tab2"
                android:layout_width="match_parent"
                android:layout_height="match_parent" >
                <TextView
                    android:text="Tab_2"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"/>
            </LinearLayout>

            <LinearLayout
                android:id="@+id/tab3"
```



```

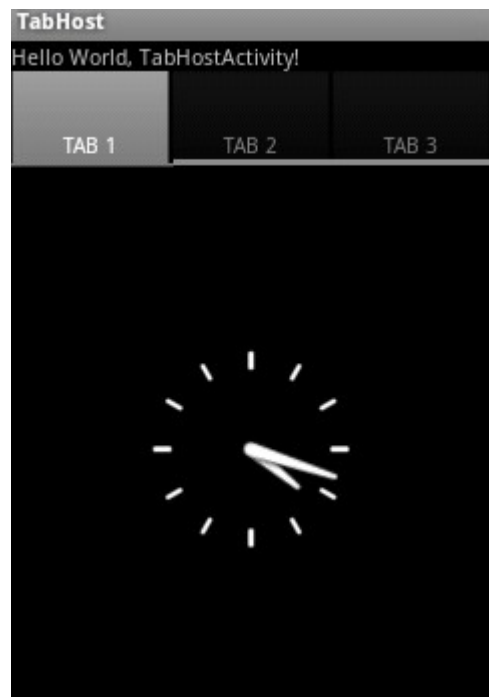
        android:layout_width="match_parent"
        android:layout_height="match_parent" >
            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"/>

            </LinearLayout>
        </FrameLayout>

    </TabHost>
</LinearLayout>

```

Dan tampilan nya akan seperti ini :



4. Buka file pada src – package – TabHostActivity.java

```

package tabHost.wartec;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TabHost;

public class TabHostActivity extends Activity {
    /** Called when the activity is first created. */
    TabHost tabs;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}

```




```
tabs = (TabHost)findViewById(R.id.tabhost);
tabs.setup();

TabHost.TabSpec spec = tabs.newTabSpec("Tag1");

//Kita akan mendeklarasikan tab 1

spec.setContent(R.id.tab1);
spec.setIndicator("TAB 1");
tabs.addTab(spec);

//tab 2

spec = tabs.newTabSpec("Tag 2");
spec.setContent(R.id.tab2);
spec.setIndicator("TAB 2");
tabs.addTab(spec);
tabs.setCurrentTab(0);

//tab 3
spec = tabs.newTabSpec("Tag 3");
spec.setContent(R.id.tab3);
spec.setIndicator("TAB 3");
tabs.addTab(spec);
tabs.setCurrentTab(0);

    }
}
```

Dan hasil nya akan seperti ini :





14. Text Watcher

Text Watcher sangat di butuhkan karena ini merupakan text pembantu , sama dengan text yang lainnya tetapi kita memakai class Watcher pada android , jadi tidak bingung bingung lagi karena sudah di sediakan , tinggal Menghubungkan langsung dengan device dan tampilan deh. Langsung ke layar kerja

1. Create Project dengan nama TextWatcher
2. Buka pada res – layout - main .xml untuk membuat tampilan

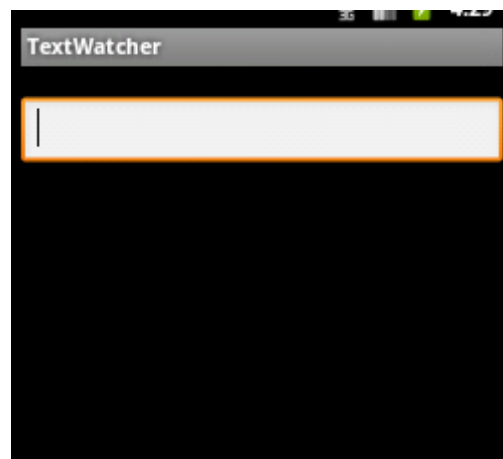
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/selection"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"/>

    <AutoCompleteTextView
        android:id="@+id/edit"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"/>

</LinearLayout>
```

AutoCompleteView lah yang akan kita gunakan , dan tampilannya akan seperti ini :



3. Buka File pada src – package – TextWatcherActivity.java

```
package textwatcher.wartec;

import android.app.Activity;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.widget.ArrayAdapter;
```



```
import android.widget.AutoCompleteTextView;
import android.widget.TextView;

//implement TextWatcher terlebih dahulu
public class TextWatcherActivity extends Activity implements TextWatcher {
    /** Called when the activity is first created. */
    TextView selection;
    AutoCompleteTextView edit;

    // Membuat Array Untuk mengisi items pada layout

    String [] items = {"dede","mamas","ryan","bapak","ibu","pakde","bude"};

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        selection = (TextView)findViewById(R.id.selection);
        edit = (AutoCompleteTextView)findViewById(R.id.edit);
        edit.setAdapter(new ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,items));
        edit.addTextChangedListener(this);
    }

    public void afterTextChanged(Editable arg0) {
        // TODO Auto-generated method stub

    }

    public void beforeTextChanged(CharSequence arg0, int arg1, int arg2,
        int arg3) {
        // TODO Auto-generated method stub

    }

    public void onTextChanged(CharSequence arg0, int arg1, int arg2, int arg3) {
        // TODO Auto-generated method stub
        edit.setText(edit.getText());
    }
}
```

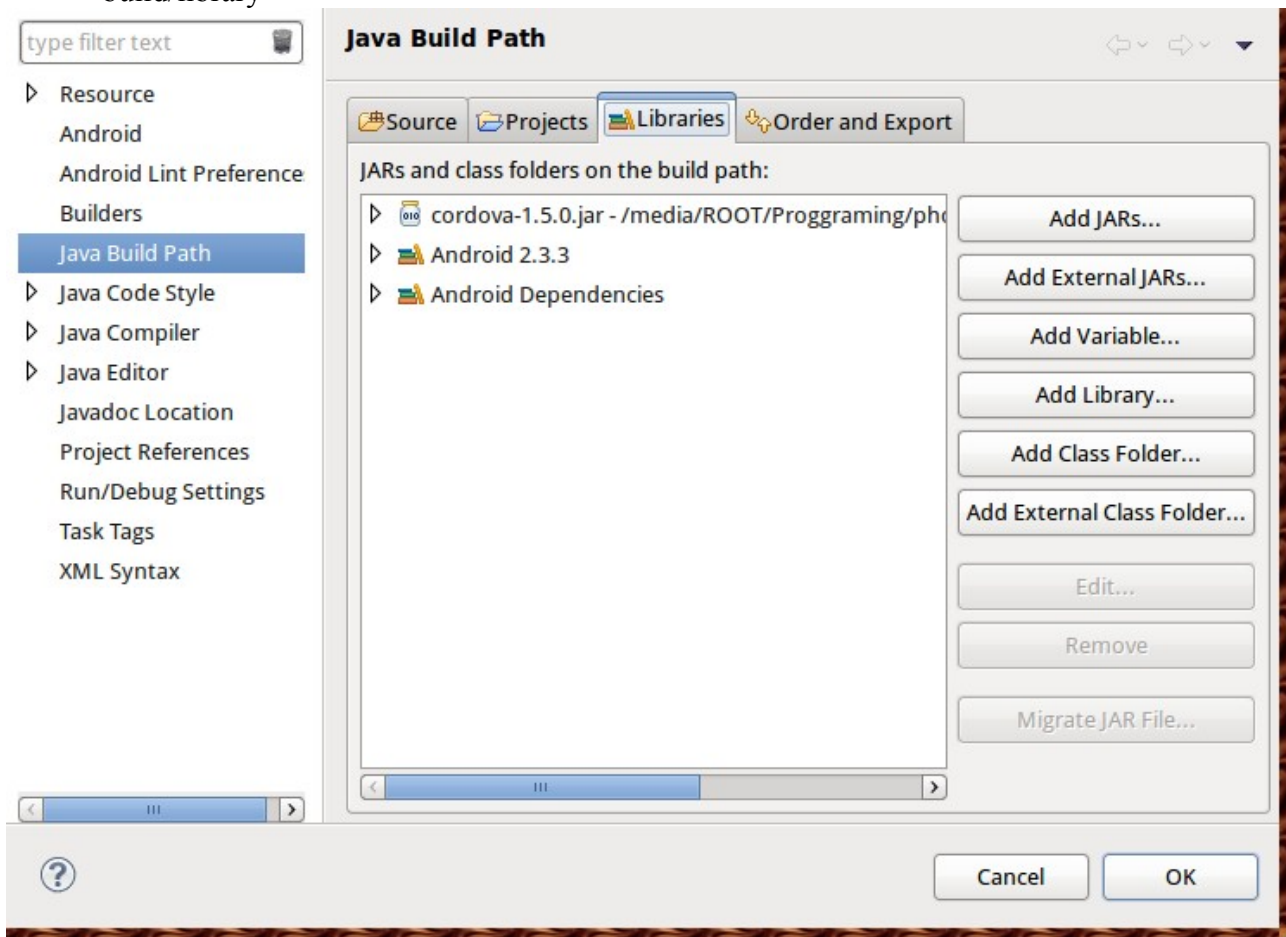
Phone Gap

Phone gap Merupakan suatu platform web yang berjalan di atas device mobile dengan menggunakan JavaScript , phone gap banyak di gunakan di plathform mobile lainnya seperti bada,blackberry,ios,symbian,webos,windows phone dan yang pastinya android, phone gap

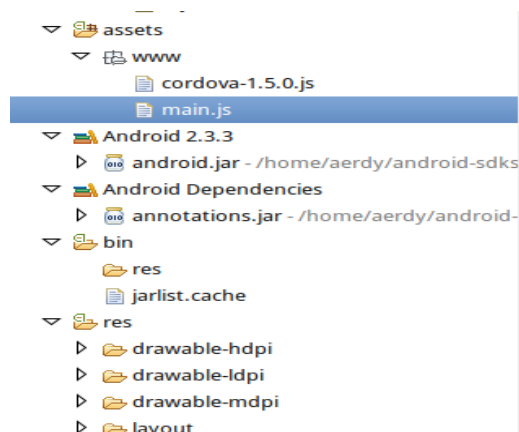


berlisensi apache Cardova , dalam membuat aplikasi berbasis web dengan phonegap kisa sudah di permudah oleh beberapa kelebihan , karena dalam phonegap ini sendiri menyediakan class class yang di butuhkan untuk mendvelop aplikasi , kita tinggal datang ke situs <http://phonegap.com> dan download package phonegap cardova . Langkah langkah menjalankan phone gap .

1. download package jar phone gap cardova di sini <http://phonegap.com>
2. file tersebut berbentuk rar jadi di butuhkan ekstrak dan didalamnya terdapat cardova.jar
3. create project dengan nama phonegapexample
4. Add package cardova.jar dengan cara klik kanan pada project build/configuration build/library



5. create folder dengan nama assets/www/
6. copy file cardova.js pada folder yang di ekstrak ke dalam folder assets/www/





6. Buka file PhoneGapActivity pada src/package

```
package phonegap.wartec;

import org.apache.cordova.DroidGap;

import android.app.Activity;
import android.os.Bundle;

public class Phone_gapActivity extends DroidGap {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        super.loadUrl("file:///android_asset/www/index.html");
    }
}
```

Kontak Manager

Kontak Manager pada kontak telephone ataupun memori eksternak dapat kita import , kontak manager biasanya digunakan pada saat kita kan membuat aplikasi yang membutuhkan kontak manager , contoh sms dan chat . Dibawah ini contoh bagaimana mengambil data dari kontak manager.

1. Create Project dengan nama ContacManger
2. Buka file main.xml untuk membuat tampilan

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

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

</LinearLayout>
```

3. Create Android.xml daan beri nama contac_manager

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <ListView android:layout_width="match_parent"
        android:id="@+id/contactList"
        android:layout_height="wrap_content"
```



```

        android:layout_weight="1"/>
<CheckBox android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/showInvisible"
        android:text="@string/showInvisible"/>
<Button android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/coba"
        android:text="coba"/>
<EditText
        android:id="@+id/isi"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
<Button android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/addContactButton"
        android:text="@string/addContactButtonLabel"/>
</LinearLayout>

```

4. Create Android.xml dan beri nama contac_entry

```

<?xml version="1.0" encoding="utf-8"?>
<!-- Copyright (C) 2009 The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    <TextView android:text="@+id/contactEntryText"
        android:id="@+id/contactEntryText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
</LinearLayout>

```

5. Create Android.xml dan beri nama contact_adder

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    <TableLayout android:layout_width="match_parent"
        android:layout_height="match_parent">
        <TableRow>
            <TextView android:layout_width="wrap_content"
                android:layout_height="wrap_content"

```



```

        android:text="@string/targetAccountLabel"/>
    </TableRow>
    <TableRow>
        <Spinner android:layout_height="wrap_content"
            android:layout_width="match_parent"
            android:layout_weight="1"
            android:id="@+id/accountSpinner"/>

    </TableRow>
    <TableRow>
        <TextView android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/contactNameLabel"/>

    </TableRow>
    <TableRow>
        <EditText android:id="@+id/contactNameEditText"
            android:layout_height="wrap_content"
            android:layout_width="wrap_content"
            android:layout_weight="1"/>

    </TableRow>
    <TableRow>
        <TextView android:text="@string/contactPhoneLabel"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>

    </TableRow>
    <TableRow>
        <EditText android:id="@+id/contactPhoneEditText"
            android:layout_height="wrap_content"
            android:layout_width="wrap_content"
            android:layout_weight="1"/>

        <Spinner android:id="@+id/contactPhoneTypeSpinner"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>

    </TableRow>
    <TableRow>
        <TextView android:text="@string/contactEmailLabel"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>

    </TableRow>
    <TableRow>
        <EditText android:id="@+id/contactEmailEditText"
            android:layout_height="wrap_content"
            android:layout_width="wrap_content"
            android:layout_weight="1"/>

        <Spinner android:id="@+id/contactEmailTypeSpinner"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>

    </TableRow>
    <TableRow>
        <Button android:layout_height="wrap_content"
            android:text="@string/save"
            android:id="@+id/contactSaveButton"
            android:layout_width="match_parent"
            android:layout_weight="1"/>

    </TableRow>
</TableLayout>
</ScrollView>

```

6. Create android.xml dan beri nama account_entry



```
<?xml version="1.0" encoding="utf-8"?>
<!-- Copyright (C) 2009 The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="?android:attr/listPreferredItemHeight"
    android:padding="6dip">
    <ImageView
        android:id="@+id/accountIcon"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_alignParentTop="true"
        android:layout_alignParentBottom="true"
        android:layout_marginRight="6dip" />
    <TextView
        android:id="@+id/secondAccountLine"
        android:layout_width="match_parent"
        android:layout_height="26dip"
        android:layout_toRightOf="@id/accountIcon"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:singleLine="true"
        android:ellipsize="marquee"
        android:textColor="@android:color/secondary_text_light" />
    <TextView
        android:id="@+id/firstAccountLine"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_toRightOf="@id/accountIcon"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:layout_above="@id/secondAccountLine"
        android:layout_alignWithParentIfMissing="true"
        android:gravity="center_vertical"
        android:textColor="@android:color/primary_text_light"/>
</RelativeLayout>
```

7. Create Class Dengan nama ContactAdder

```
package contact.wartech;
import android.accounts.Account;
import android.accounts.AccountManager;
import android.accounts.AuthenticatorDescription;
import android.accounts.OnAccountsUpdateListener;
```




```
import android.app.Activity;
import android.content.ContentProviderOperation;
import android.content.Context;
import android.content.pm.PackageManager;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.AdapterView.OnItemClickListener;

import java.util.ArrayList;
import java.util.Iterator;

public class ContactAdder extends Activity implements OnAccountsUpdateListener {
    public static final String TAG = "ContactAdder";
    public static final String ACCOUNT_NAME =
        "com.example.android.contactmanager.ContactAdder.ACCOUNT_NAME";
    public static final String ACCOUNT_TYPE =
        "com.example.android.contactmanager.ContactAdder.ACCOUNT_TYPE";

    private ArrayList<AccountData> mAccounts;
    private AccountAdapter mAccountAdapter;
    private Spinner mAccountSpinner;
    private EditText mContactEmailEditText;
    private ArrayList<Integer> mContactEmailTypes;
    private Spinner mContactEmailTypeSpinner;
    private EditText mContactNameEditText;
    private EditText mContactPhoneEditText;
    private ArrayList<Integer> mContactPhoneTypes;
    private Spinner mContactPhoneTypeSpinner;
    private Button mContactSaveButton;
    private AccountData mSelectedAccount;

    /**
     * Called when the activity is first created. Responsible for initializing the UI.
     */
    @Override
    public void onCreate(Bundle savedInstanceState)
```




```
{
    Log.v(TAG, "Activity State: onCreate()");
    super.onCreate(savedInstanceState);
    setContentView(R.layout.contact_adder);

    // Obtain handles to UI objects
    mAccountSpinner = (Spinner) findViewById(R.id.accountSpinner);
    mContactNameEditText = (EditText) findViewById(R.id.contactNameEditText);
    mContactPhoneEditText = (EditText) findViewById(R.id.contactPhoneEditText);
    mContactEmailEditText = (EditText) findViewById(R.id.contactEmailEditText);
    mContactPhoneTypeSpinner = (Spinner)
        findViewById(R.id.contactPhoneTypeSpinner);
    mContactEmailTypeSpinner = (Spinner)
        findViewById(R.id.contactEmailTypeSpinner);
    mContactSaveButton = (Button) findViewById(R.id.contactSaveButton);

    // Prepare list of supported account types
    // Note: Other types are available in ContactsContract.CommonDataKinds
    // Also, be aware that type IDs differ between Phone and Email, and MUST be
    computed
    // separately.
    mContactPhoneTypes = new ArrayList<Integer>();

    mContactPhoneTypes.add(ContactsContract.CommonDataKinds.Phone.TYPE_HOME);
    mContactPhoneTypes.add(ContactsContract.CommonDataKinds.Phone.TYPE_WORK);
    mContactPhoneTypes.add(ContactsContract.CommonDataKinds.Phone.TYPE_MOBILE);
    mContactPhoneTypes.add(ContactsContract.CommonDataKinds.Phone.TYPE_OTHER);
    mContactEmailTypes = new ArrayList<Integer>();

    mContactEmailTypes.add(ContactsContract.CommonDataKinds.Email.TYPE_HOME);
    mContactEmailTypes.add(ContactsContract.CommonDataKinds.Email.TYPE_WORK);
    mContactEmailTypes.add(ContactsContract.CommonDataKinds.Email.TYPE_MOBILE);
    mContactEmailTypes.add(ContactsContract.CommonDataKinds.Email.TYPE_OTHER);

    // Prepare model for account spinner
    mAccounts = new ArrayList<AccountData>();
    mAccountAdapter = new AccountAdapter(this, mAccounts);
    mAccountSpinner.setAdapter(mAccountAdapter);

    // Populate list of account types for phone
    ArrayAdapter<String> adapter;
    adapter = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item);
```



```

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    Iterator<Integer> iter;
    iter = mContactPhoneTypes.iterator();
    while (iter.hasNext()) {
        adapter.add(ContactsContract.CommonDataKinds.Phone.getTypeLabel(
            this.getResources(),
            iter.next(),
            getString(R.string.undefinedTypeLabel)).toString());
    }
    mContactPhoneTypeSpinner.setAdapter(adapter);
    mContactPhoneTypeSpinner.setPrompt(getString(R.string.selectLabel));

    // Populate list of account types for email
    adapter = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item);

    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    iter = mContactEmailTypes.iterator();
    while (iter.hasNext()) {
        adapter.add(ContactsContract.CommonDataKinds.Email.getTypeLabel(
            this.getResources(),
            iter.next(),
            getString(R.string.undefinedTypeLabel)).toString());
    }
    mContactEmailTypeSpinner.setAdapter(adapter);
    mContactEmailTypeSpinner.setPrompt(getString(R.string.selectLabel));

    // Prepare the system account manager. On registering the listener below, we also ask
for
    // an initial callback to pre-populate the account list.
    AccountManager.get(this).addOnAccountsUpdatedListener(this, null, true);

    // Register handlers for UI elements
    mAccountSpinner.setOnItemSelectedListener(new OnItemSelectedListener() {
        public void onItemSelected(AdapterView<?> parent, View view, int position, long
i) {
            updateAccountSelection();
        }

        public void onNothingSelected(AdapterView<?> parent) {
            // We don't need to worry about nothing being selected, since Spinners don't
allow
            // this.
        }
    });
    mContactSaveButton.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            onSaveButtonClicked();
        }
    });

```



```

    }

    /**
     * Actions for when the Save button is clicked. Creates a contact entry and terminates the
     * activity.
     */
    private void onSaveButtonClicked() {
        Log.v(TAG, "Save button clicked");
        createContactEntry();
        finish();
    }

    /**
     * Creates a contact entry from the current UI values in the account named by
     mSelectedAccount.
     */
    protected void createContactEntry() {
        // Get values from UI
        String name = mContactNameEditText.getText().toString();
        String phone = mContactPhoneEditText.getText().toString();
        String email = mContactEmailEditText.getText().toString();
        int phoneType = mContactPhoneTypes.get(
            mContactPhoneTypeSpinner.getSelectedItemPosition());
        int emailType = mContactEmailTypes.get(
            mContactEmailTypeSpinner.getSelectedItemPosition());

        // Prepare contact creation request
        //
        // Note: We use RawContacts because this data must be associated with a particular
        account.
        // The system will aggregate this with any other data for this contact and create a
        // corresponding entry in the ContactsContract.Contacts provider for us.
        ArrayList<ContentProviderOperation> ops = new
        ArrayList<ContentProviderOperation>();

        ops.add(ContentProviderOperation.newInsert(ContactsContract.RawContacts.CONTENT_URI)
            .withValue(ContactsContract.RawContacts.ACCOUNT_TYPE,
            mSelectedAccount.getType())
            .withValue(ContactsContract.RawContacts.ACCOUNT_NAME,
            mSelectedAccount.getName())
            .build());

        ops.add(ContentProviderOperation.newInsert(ContactsContract.Data.CONTENT_URI)
            .withValueBackReference(ContactsContract.Data.RAW_CONTACT_ID, 0)
            .withValue(ContactsContract.Data.MIMETYPE,
            ContactsContract.CommonDataKinds.StructuredName.CONTENT_ITEM_TYPE)
            .withValue(ContactsContract.CommonDataKinds.StructuredName.DISPLAY_NAME, name)

```



```

        .build());

ops.add(ContentProviderOperation.newInsert(ContactsContract.Data.CONTENT_URI)
        .withValueBackReference(ContactsContract.Data.RAW_CONTACT_ID, 0)
        .withValue(ContactsContract.Data.MIMETYPE,
            ContactsContract.CommonDataKinds.Phone.CONTENT_ITEM_TYPE)
        .withValue(ContactsContract.CommonDataKinds.Phone.NUMBER, phone)
        .withValue(ContactsContract.CommonDataKinds.Phone.TYPE, phoneType)
        .build());

ops.add(ContentProviderOperation.newInsert(ContactsContract.Data.CONTENT_URI)
        .withValueBackReference(ContactsContract.Data.RAW_CONTACT_ID, 0)
        .withValue(ContactsContract.Data.MIMETYPE,
            ContactsContract.CommonDataKinds.Email.CONTENT_ITEM_TYPE)
        .withValue(ContactsContract.CommonDataKinds.Email.DATA, email)
        .withValue(ContactsContract.CommonDataKinds.Email.TYPE, emailType)
        .build());

// Ask the Contact provider to create a new contact
Log.i(TAG, "Selected account: " + mSelectedAccount.getName() + " (" +
    mSelectedAccount.getType() + ")");
Log.i(TAG, "Creating contact: " + name);
try {
    getContentResolver().applyBatch(ContactsContract.AUTHORITY, ops);
} catch (Exception e) {
    // Display warning
    Context ctx = getApplicationContext();
    CharSequence txt = getString(R.string.contactCreationFailure);
    int duration = Toast.LENGTH_SHORT;
    Toast toast = Toast.makeText(ctx, txt, duration);
    toast.show();

    // Log exception
    Log.e(TAG, "Exception encountered while inserting contact: " + e);
}
}

/**
 * Called when this activity is about to be destroyed by the system.
 */
@Override
public void onDestroy() {
    // Remove AccountManager callback
    AccountManager.get(this).removeOnAccountsUpdatedListener(this);
    super.onDestroy();
}

/**
 * Updates account list spinner when the list of Accounts on the system changes.

```




Satisfies

```

* OnAccountsUpdateListener implementation.
*/
public void onAccountsUpdated(Account[] a) {
    Log.i(TAG, "Account list update detected");
    // Clear out any old data to prevent duplicates
    mAccounts.clear();

    // Get account data from system
    AuthenticatorDescription[] accountTypes =
AccountManager.get(this).getAuthenticatorTypes();

    // Populate tables
    for (int i = 0; i < a.length; i++) {
        // The user may have multiple accounts with the same name, so we need to
construct a
        // meaningful display name for each.
        String systemAccountType = a[i].type;
        AuthenticatorDescription ad = getAuthenticatorDescription(systemAccountType,
            accountTypes);
        AccountData data = new AccountData(a[i].name, ad);
        mAccounts.add(data);
    }

    // Update the account spinner
    mAccountAdapter.notifyDataSetChanged();
}

/**
 * Obtain the AuthenticatorDescription for a given account type.
 * @param type The account type to locate.
 * @param dictionary An array of AuthenticatorDescriptions, as returned by
AccountManager.
 * @return The description for the specified account type.
 */
private static AuthenticatorDescription getAuthenticatorDescription(String type,
    AuthenticatorDescription[] dictionary) {
    for (int i = 0; i < dictionary.length; i++) {
        if (dictionary[i].type.equals(type)) {
            return dictionary[i];
        }
    }
    // No match found
    throw new RuntimeException("Unable to find matching authenticator");
}

/**
 * Update account selection. If NO_ACCOUNT is selected, then we prohibit inserting
new contacts.

```




```

    */
    private void updateAccountSelection() {
        // Read current account selection
        mSelectedAccount = (AccountData) mAccountSpinner.getSelectedItem();
    }

    /**
     * A container class used to represent all known information about an account.
     */
    private class AccountData {
        private String mName;
        private String mType;
        private CharSequence mTypeLabel;
        private Drawable mIcon;

        /**
         * @param name The name of the account. This is usually the user's email address or
         *             username.
         * @param description The description for this account. This will be dictated by the
         *                    type of account returned, and can be obtained from the system
AccountManager.
         */
        public AccountData(String name, AuthenticatorDescription description) {
            mName = name;
            if (description != null) {
                mType = description.type;

                // The type string is stored in a resource, so we need to convert it into something
                // human readable.
                String packageName = description.packageName;
                PackageManager pm = getPackageManager();

                if (description.labelId != 0) {
                    mTypeLabel = pm.getText(packageName, description.labelId, null);
                    if (mTypeLabel == null) {
                        throw new IllegalArgumentException("LabelID provided, but label not
found");
                    }
                } else {
                    mTypeLabel = "";
                }

                if (description.iconId != 0) {
                    mIcon = pm.getDrawable(packageName, description.iconId, null);
                    if (mIcon == null) {
                        throw new IllegalArgumentException("IconID provided, but drawable not "
+
                        "found");
                    }
                }
            }
        }
    }

```



```

        } else {
            mIcon = getResources().getDrawable(android.R.drawable.sym_def_app_icon);
        }
    }
}

public String getName() {
    return mName;
}

public String getType() {
    return mType;
}

public CharSequence getTypeLabel() {
    return mTypeLabel;
}

public Drawable getIcon() {
    return mIcon;
}

public String toString() {
    return mName;
}
}

/**
 * Custom adapter used to display account icons and descriptions in the account spinner.
 */
private class AccountAdapter extends ArrayAdapter<AccountData> {
    public AccountAdapter(Context context, ArrayList<AccountData> accountData) {
        super(context, android.R.layout.simple_spinner_item, accountData);
        setDropDownViewResource(R.layout.account_entry);
    }

    public View getDropDownView(int position, View convertView, ViewGroup parent) {
        // Inflate a view template
        if (convertView == null) {
            LayoutInflater inflater = getLayoutInflater();
            convertView = inflater.inflate(R.layout.account_entry, parent, false);
        }

        TextView firstAccountLine = (TextView)
convertView.findViewById(R.id.firstAccountLine);
        TextView secondAccountLine = (TextView)
convertView.findViewById(R.id.secondAccountLine);
        ImageView accountIcon = (ImageView)
convertView.findViewById(R.id.accountIcon);

```



```

        // Populate template
        AccountData data = getItem(position);
        firstAccountLine.setText(data.getName());
        secondAccountLine.setText(data.getTypeLabel());
        Drawable icon = data.getIcon();
        if (icon == null) {
            icon = getResources().getDrawable(android.R.drawable.ic_menu_search);
        }
        accountIcon.setImageDrawable(icon);
        return convertView;
    }
}

```

8. Buka file ContactManagerActivity.java

```

package contact.wartech;

import android.app.Activity;
import android.content.Intent;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.CompoundButton.OnCheckedChangeListener;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.SimpleCursorAdapter;
import android.os.Bundle;

public class ContactManagerActivity extends Activity {
    /** Called when the activity is first created. */
    public static final String TAG = "ContactManager";

    private Button mAddAccountButton,coba;
    private ListView mContactList;
    private boolean mShowInvisible;
    private CheckBox mShowInvisibleControl;
    EditText isi;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.contact_manager);
    }
}

```



```
// Obtain handles to UI objects
mAddAccountButton = (Button) findViewById(R.id.addContactButton);
mContactList = (ListView) findViewById(R.id.contactList);
mShowInvisibleControl = (CheckBox) findViewById(R.id.showInvisible);
isi = (EditText) findViewById(R.id.isi);
// Initialize class properties
mShowInvisible = false;
mShowInvisibleControl.setChecked(mShowInvisible);

// Register handler for UI elements
mAddAccountButton.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Log.d(TAG, "mAddAccountButton clicked");
        launchContactAdder();
    }
});
mShowInvisibleControl.setOnCheckedChangeListener(new OnCheckedChangeListener() {
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        Log.d(TAG, "mShowInvisibleControl changed: " + isChecked);
        mShowInvisible = isChecked;
        populateContactList();
    }
});

// Populate the contact list
populateContactList();

coba = (Button) findViewById(R.id.coba);
coba.setOnClickListener(new View.OnClickListener() {

    public void onClick(View arg0) {
        // TODO Auto-generated method stub
        Intent intent = new Intent(Intent.ACTION_PICK,
ContactsContract.Contacts.CONTENT_URI);
        startActivityForResult(intent, 1);
    }
});
}
private void populateContactList() {
    // Build adapter with contact entries
    Cursor cursor = getContacts();
    String[] fields = new String[] {
        ContactsContract.Data.DISPLAY_NAME
    };
    SimpleCursorAdapter adapter = new SimpleCursorAdapter(this, R.layout.contact_entry,
cursor,
        fields, new int[] {R.id.contactEntryText});
    mContactList.setAdapter(adapter);
}
```



```

/**
 * Obtains the contact list for the currently selected account.
 *
 * @return A cursor for for accessing the contact list.
 */
private Cursor getContacts()
{
    // Run query
    Uri uri = ContactsContract.Contacts.CONTENT_URI;
    String[] projection = new String[] {
        ContactsContract.Contacts._ID,
        ContactsContract.Contacts.DISPLAY_NAME
    };
    String selection = ContactsContract.Contacts.IN_VISIBLE_GROUP + " = " +
        (mShowInvisible ? "0" : "1") + """";
    String[] selectionArgs = null;
    String sortOrder = ContactsContract.Contacts.DISPLAY_NAME + " COLLATE
LOCALIZED ASC";

    return managedQuery(uri, projection, selection, selectionArgs, sortOrder);
}

/**
 * Launches the ContactAdder activity to add a new contact to the selected account.
 */
protected void launchContactAdder() {
    Intent i = new Intent(this, ContactAdder.class);
    startActivity(i);
}

public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode == Activity.RESULT_OK) {
        Uri contactData = data.getData();
        Cursor c = managedQuery(contactData, null, null, null, null);
        if (c.moveToFirst()) {
            String name =
c.getString(c.getColumnIndexOrThrow(ContactsContract.Contacts.CONTENT_VCARD_TYPE))
;
            isi.setText(name);
        }
    }
}
}

```

9. Buka file AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="contact.wartech"

```




```

android:versionCode="1"
android:versionName="1.0" >

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

<application
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name" >
    <activity
        android:name=".ContactManagerActivity"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

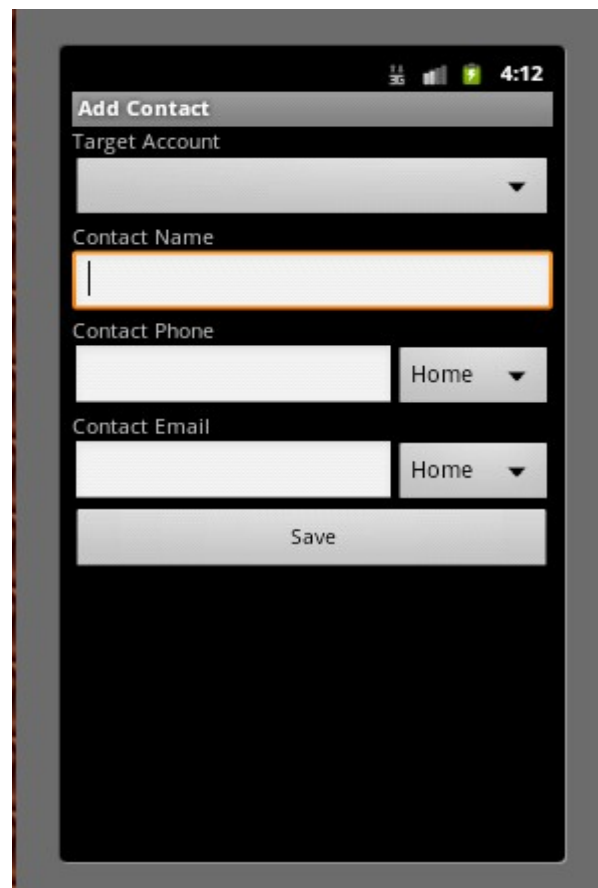
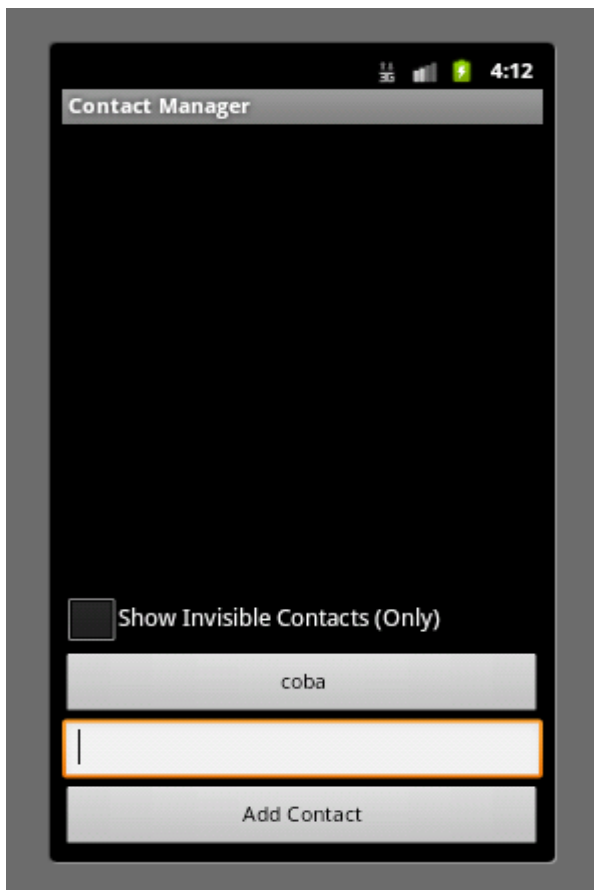
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>

        </activity>
        <activity android:name="ContactAdder"
android:label="@string/addContactTitle">
        </activity>
    </application>

<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.WRITE_CONTACTS"/>
</manifest>

```

10. Run Project dan hasilnya seperti di bawah ini :





Data Base Internal Pada Android (SQLite)

Pengenalan SQLite

SQLite adalah database Open Source yang tertanam ke Android. SQLite mendukung fitur database relasional standar seperti sintaks SQL, transaksi dan laporan disusun. Selain itu hanya memerlukan sedikit memori pada saat runtime (sekitar 250 Kbyte). SQLite mendukung TEKS tipe data (mirip dengan String di Java), INTEGER (mirip dengan yang lama di Java) dan REAL (mirip dengan dua kali lipat di Java). Semua jenis lain harus dikonversi ke dalam salah satu bidang sebelum menyimpannya dalam database. SQLite sendiri tidak memvalidasi jika jenis ditulis untuk kolom sebenarnya dari jenis yang didefinisikan, misalnya Anda dapat menulis sebuah integer ke dalam kolom string dan sebaliknya. Informasi lebih lanjut tentang SQLite dapat ditemukan di situs web SQLite: <http://www.sqlite.org>.

SQLite Dalam Android..?

SQLite adalah tersedia pada setiap perangkat Android. Menggunakan database SQLite di Android tidak memerlukan setup database atau administrasi. Anda hanya perlu menentukan pernyataan SQL untuk membuat dan memperbarui database. Setelah database secara otomatis dikelola untuk Anda dengan platform Android. Akses ke database SQLite melibatkan pengaksesan sistem berkas. Hal ini bisa lambat. Oleh karena itu dianjurkan untuk melakukan operasi database asynchronous, misalnya melalui kelas AsyncTask. . Jika aplikasi Anda menciptakan database, database ini disimpan di direktori DATA / data / APP_NAME / database / FILENAME. Bagian-bagian dari direktori di atas dibangun berdasarkan aturan berikut. DATA adalah jalan yang Environment.getDataDirectory () kembali metode. APP_NAME adalah nama aplikasi Anda. FILENAME adalah nama yang Anda tentukan dalam kode aplikasi Anda untuk database.

Struktur SQLite

1. Package

package android.database paket berisi semua kelas umum untuk bekerja dengan database. android.database.sqlite berisi kelas SQLite tertentu.

2. SQLiteOpenHelper

Untuk membuat dan meng-upgrade database dalam aplikasi Android Anda Anda biasanya subclass SQLiteOpenHelper. Dalam konstruktor dari subclass Anda Anda memanggil super () metode SQLiteOpenHelper, menentukan nama database dan versi database saat ini. Di kelas ini, Anda perlu untuk menimpa onCreate () dan onUpgrade () metode. onCreate () dipanggil oleh framework, jika database tidak ada. onUpgrade () dipanggil, jika versi database meningkat pada kode aplikasi anda. Metode ini memungkinkan Anda untuk memperbarui skema database. Kedua metode tersebut menerima objek SQLiteDatabase sebagai parameter yang mewakili database. SQLiteOpenHelper menyediakan getReadableDatabase metode () dan getWritableDatabase () untuk mendapatkan akses ke objek SQLiteDatabase; baik dalam membaca atau menulis mode. Tabel database harus menggunakan _id pengidentifikasi untuk primary key dari tabel. Beberapa fungsi Android bergantung pada standar ini. Ini adalah praktek terbaik untuk membuat kelas terpisah per meja. Kelas ini mendefinisikan statis onCreate () dan onUpdate () metode. Metode ini disebut dalam metode yang sesuai dari SQLiteOpenHelper. Dengan cara ini pelaksanaan Anda dari SQLiteOpenHelper akan tetap dibaca, bahkan jika Anda memiliki beberapa tabel.



3. SQLiteDatabase

SQLiteDatabase adalah kelas dasar untuk bekerja dengan database SQLite di Android dan menyediakan metode untuk membuka, query, update dan menutup database. Lebih khusus SQLiteDatabase menyediakan insert (), update () dan menghapus () metode. Selain itu ia menyediakan execSQL () metode, yang memungkinkan untuk mengeksekusi SQL langsung. Para ContentValues objek memungkinkan untuk mendefinisikan kunci / nilai. "Kunci" merupakan identifier kolom tabel dan "nilai" merupakan konten untuk catatan tabel dalam kolom ini. ContentValues dapat digunakan untuk menyisipkan dan update entri database. Pertanyaan dapat diciptakan melalui rawQuery () dan query () metode atau melalui kelas SQLiteQueryBuilder. rawQuery () langsung menerima pernyataan SQL sebagai masukan. query () menyediakan antarmuka terstruktur untuk menentukan query SQL. SQLiteQueryBuilder adalah kelas kenyamanan yang membantu untuk membangun query SQL.

4. Cursor

Sebuah query mengembalikan sebuah objek kursor. Kursor Sebuah merupakan hasil dari query dan pada dasarnya menunjuk pada satu baris dari hasil query. Dengan cara ini Android bisa buffer hasil query efisien; karena tidak harus memuat semua data ke memori. Untuk mendapatkan jumlah elemen dari query yang dihasilkan menggunakan getCount () method. Untuk beralih di antara baris data individu, Anda dapat menggunakan moveToFirst () dan moveToNext () metode. Para isAfterLast () metode memungkinkan untuk memeriksa apakah akhir hasil query telah tercapai. Kursor menyediakan diketik mendapatkan metode * (), misalnya getLong (columnIndex), GetString (columnIndex) untuk mengakses data kolom untuk posisi saat ini hasilnya. The "columnIndex" adalah jumlah dari .kolom Anda mengakses. Kursor juga menyediakan getColumnIndexOrThrow (String) metode yang memungkinkan untuk mendapatkan indeks kolom untuk nama kolom dari tabel.

5. ListViews, ListActivities and SimpleCursorAdapter

ListViews adalah Tampilan yang memungkinkan untuk menampilkan daftar elemen. ListActivities adalah Kegiatan khusus yang membuat penggunaan ListViews lebih mudah. Untuk bekerja dengan database dan ListViews Anda dapat menggunakan SimpleCursorAdapter. Para SimpleCursorAdapter memungkinkan untuk mengatur tata letak untuk setiap baris dari ListViews. Anda juga mendefinisikan sebuah array yang berisi nama kolom dan lain array yang berisi ID Views yang harus diisi dengan data. Kelas SimpleCursorAdapter akan memetakan kolom ke Tampilan berdasarkan kursor dikirimkan ke sana. Untuk mendapatkan kursor, Anda harus menggunakan kelas Loader. Tutorial ini akan menggunakan ListActivities tetapi tidak melihat secara rinci dari mereka.

Sekarang kita langsung ke lembar kerja project :

1. Create Project dengan nama DataBaseSQLite
2. Kita akan membuat tampilan terlebih dahulu , Buka file res – layout – main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <LinearLayout
```



```

        android:id="@+id/group"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >

        <Button
            android:id="@+id/add"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Add New"
            android:onClick="onClick"/>

        <Button
            android:id="@+id/delete"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Delete First"
            android:onClick="onClick"/>

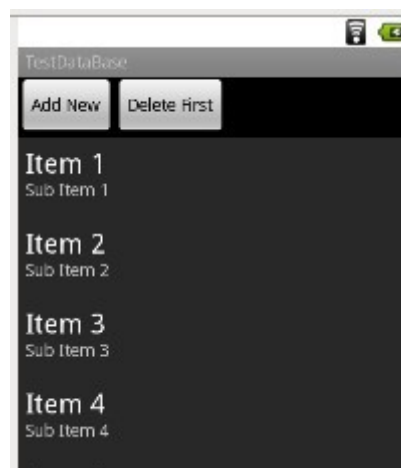
    </LinearLayout>

    <ListView
        android:id="@android:id/list"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

</LinearLayout>

```

Dan tampilan nya akan seperti ini :



3. Create Class pada folder src – package dan beri nama class dengan MySQLiteHelper untuk membuat class sebagai control pembuat data base dengan sendiri nya seperti crud pada java dan Tuliskan Scrip di bawah ini:

```

package testdatabase.wartec;

import android.content.Context;

```




```
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

public class MySQLiteHelper extends SQLiteOpenHelper{
    public static final String TABLE_COMMENTS = "comments";
    public static final String COLUMN_ID = "_id";
    public static final String COLUMN_COMMENT = "comment";

    private static final String DATABASE_NAME = "commments.db";
    private static final int DATABASE_VERSION = 1;

    // Statement Membuat create data base
    private static final String DATABASE_CREATE = "create table "
        + TABLE_COMMENTS + "( " + COLUMN_ID
        + " integer primary key autoincrement, " + COLUMN_COMMENT
        + " text not null);";

    public MySQLiteHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase database) {
        database.execSQL(DATABASE_CREATE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        Log.w(MySQLiteHelper.class.getName(),
            "Upgrading database from version " + oldVersion + " to "
            + newVersion + ", which will destroy all old data");
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_COMMENTS);
        onCreate(db);
    }
}
```

4. Create Class pada folder src – package dan beri nama class Comment bertujuan untuk membuat model entitas . Karena dalam konsep sebelumnya kita menggunakan konsep MVC.

```
package testdatabase.wartec;

//Class Entitas
public class Comment {
    private long id;
    private String comment;

    public long getId() {
        return id;
    }
}
```




```

    }

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

    public String getComment() {
        return comment;
    }

    public void setComment(String comment) {
        this.comment = comment;
    }

    // Menggunakan oleh ArrayAdapter di ListView
    @Override
    public String toString() {
        return comment;
    }
}

```

5. Create class dalam folder src - package dan beri nama class dengan CommentsDataSource yang bertujuan membuat kontrol hapus data base .

```

package testdatabase.wartec;
import java.util.ArrayList;
import java.util.List;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
public class CommentsDataSource {
    // Database fields
    private SQLiteDatabase database;
    private MySQLiteHelper dbHelper;
    private String[] allColumns = { MySQLiteHelper.COLUMN_ID,
                                    MySQLiteHelper.COLUMN_COMMENT };

    public CommentsDataSource(Context context) {
        dbHelper = new MySQLiteHelper(context);
    }

    public void open() throws SQLException {
        database = dbHelper.getWritableDatabase();
    }

    public void close() {
        dbHelper.close();
    }
}

```



```

        public Comment createComment(String comment) {
            ContentValues values = new ContentValues();
            values.put(MySQLiteHelper.COLUMN_COMMENT, comment);
            long insertId = database.insert(MySQLiteHelper.TABLE_COMMENTS,
null,
                values);
            Cursor cursor = database.query(MySQLiteHelper.TABLE_COMMENTS,
                allColumns, MySQLiteHelper.COLUMN_ID + " = " +
insertId, null,
                null, null, null);
            cursor.moveToFirst();
            Comment newComment = cursorToComment(cursor);
            cursor.close();
            return newComment;
        }

        public void deleteComment(Comment comment) {
            long id = comment.getId();
            System.out.println("Comment deleted with id: " + id);
            database.delete(MySQLiteHelper.TABLE_COMMENTS,
MySQLiteHelper.COLUMN_ID
                + " = " + id, null);
        }

        public List<Comment> getAllComments() {
            List<Comment> comments = new ArrayList<Comment>();

            Cursor cursor = database.query(MySQLiteHelper.TABLE_COMMENTS,
                allColumns, null, null, null, null, null);

            cursor.moveToFirst();
            while (!cursor.isAfterLast()) {
                Comment comment = cursorToComment(cursor);
                comments.add(comment);
                cursor.moveToNext();
            }
            //untuk menutup kursor
            cursor.close();
            return comments;
        }

        private Comment cursorToComment(Cursor cursor) {
            Comment comment = new Comment();
            comment.setId(cursor.getLong(0));
            comment.setComment(cursor.getString(1));
            return comment;
        }
    }

```



6. Buka file TestDataBaseActivity pada folder src – package – TestDataBaseActivity.java dan isikan scrip di bawah ini : yang merupakan sebuah main dari package

```
package testdatabase.wartec;

import android.app.Activity;
import java.util.List;
import java.util.Random;

import android.app.ListActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.os.Bundle;

public class TestDataBaseActivity extends ListActivity{
    /** Called when the activity is first created. */
    private CommentsDataSource datasource;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        datasource = new CommentsDataSource(this);
        datasource.open();

        List<Comment> values = datasource.getAllComments();

        // Menggunakan SimpleCursorAdapter untuk menunjukkan
        //Elemen dalam ListView
        ArrayAdapter<Comment> adapter = new ArrayAdapter<Comment>(this,
            android.R.layout.simple_list_item_1, values);
        setListAdapter(adapter);
    }

    public void onClick(View view) {
        @SuppressWarnings("unchecked")
        ArrayAdapter<Comment> adapter = (ArrayAdapter<Comment>) getListAdapter();
        Comment comment = null;
        switch (view.getId()) {
            case R.id.add:
                String[] comments = new String[] { "Dia", "Aku", "Mereka" };
                int nextInt = new Random().nextInt(3);
                // Menyimpan Command Array ke dalam data base
                comment = datasource.createComment(comments[nextInt]);
                adapter.add(comment);
                break;
            case R.id.delete:
                if (getListAdapter().getCount() > 0) {
                    comment = (Comment) getListAdapter().getItem(0);
                }
            }
        }
    }
}
```



```

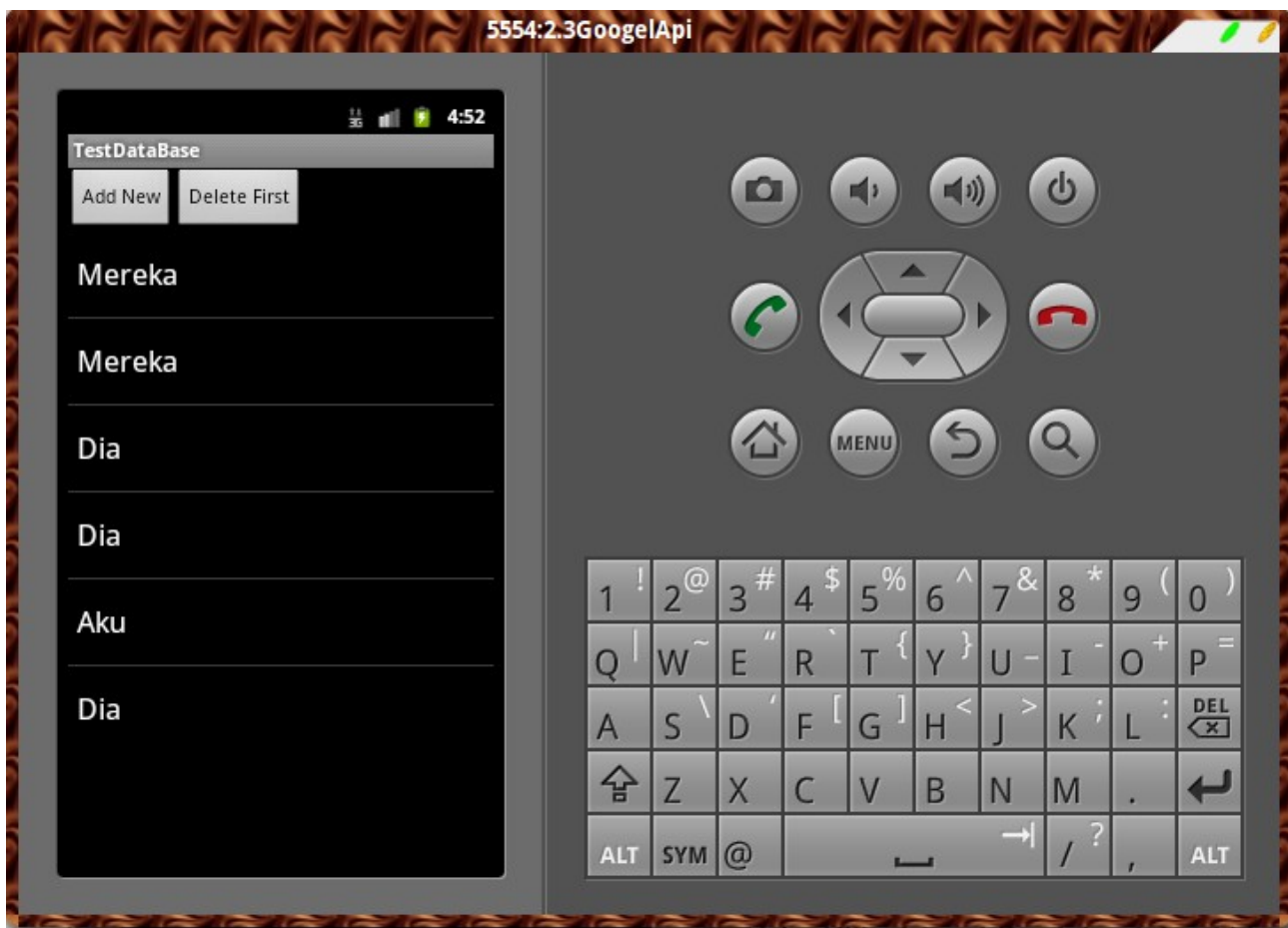
        datasource.deleteComment(comment);
        adapter.remove(comment);
    }
    break;
}
adapter.notifyDataSetChanged();
}

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

@Override
protected void onPause() {
    datasource.close();
    super.onPause();
}
}

```

Run Project dan hasil nya akan seperti ini :





HTTP Request

HTTP pada dasarnya merupakan suatu port http

sebagai contoh seperti ini :

```
package httpreques.wartec;

import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.List;

import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;
import android.os.Bundle;

public class HTTPPresquesActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        // Creating HTTP client
        HttpClient httpClient = new DefaultHttpClient();
        // Creating HTTP Post
        HttpPost httpPost = new HttpPost(
            "http://www.example.com/login");

        // Building post parameters
        // key and value pair
        List<NameValuePair> nameValuePair = new ArrayList<NameValuePair>(2);
        nameValuePair.add(new BasicNameValuePair("email", "user@gmail.com"));
        nameValuePair.add(new BasicNameValuePair("message",
            "Hi, trying Android HTTP post!"));

        // Url Encoding the POST parameters
        try {
            httpPost.setEntity(new UrlEncodedFormEntity(nameValuePair));
```




```

    } catch (UnsupportedEncodingException e) {
        // writing error to Log
        e.printStackTrace();
    }

    // Making HTTP Request
    try {
        HttpResponse response = httpClient.execute(httpPost);

        // writing response to log
        Log.d("Http Response:", response.toString());
    } catch (ClientProtocolException e) {
        // writing exception to log
        e.printStackTrace();
    } catch (IOException e) {
        // writing exception to log
        e.printStackTrace();
    }
}
}

```

Facebook Desain

Kita belajar mendesain sebuah tampilan yang di lakukan oleh perusahaan mark z . yaitu facebook . Karena dengan kita memperlajari desain ini kita dapat melihat trik trik yang digunakan dalam mendesain sebuah aplikasi , ini juga merupakan desain yang di berikana oleh platform android sendiri , kalau sudah mendapatkan platform 2.2 maka pad project android dalam folder samples , disini saya hanya menjelaskan bagaimana bisa seperti itu :





JSON Parsing

JSON (dilafalkan "Jason"), singkatan dari JavaScript Object Notation (bahasa Indonesia: notasi objek JavaScript), adalah suatu format ringkas pertukaran data komputer. Formatnya berbasis teks dan terbaca-manusia serta digunakan untuk merepresentasikan struktur data sederhana dan larik asosiatif (disebut objek). Format JSON sering digunakan untuk mentransmisikan data terstruktur melalui suatu koneksi jaringan pada suatu proses yang disebut serialisasi. Aplikasi utamanya adalah pada pemrograman aplikasi web AJAX dengan berperan sebagai alternatif terhadap penggunaan tradisional format XML.

JSON adalah himpunan bagian dari notasi harfiah objek JavaScript. Sejak JSON adalah himpunan bagian dari JavaScript, dapat digunakan dalam bahasa tanpa muss atau fuss.

Mungkin lebih tepatnya kita langsung ke contoh project :

1. Create Project dengan nama JSON_Parsing
2. Buka file res – layout – main.xml

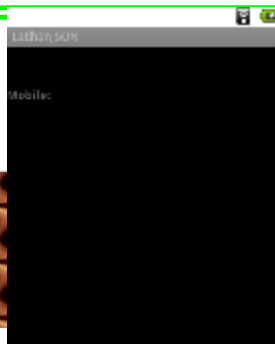
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">
    <!-- Main ListView
        Always give id value as list(@android:id/list)
    -->
    <ListView
        android:id="@android:id/list"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"/>
</LinearLayout>
```



3. Create Android xml untuk membuat tampilan list item contact dengan list_item:



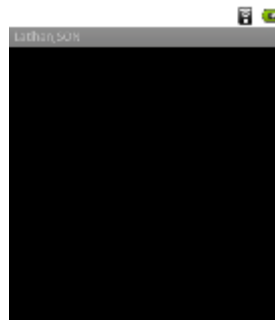
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <LinearLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <!-- Name Label -->
        <TextView
            android:id="@+id/name"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:textColor="#43bd00"
            android:textSize="16sp"
            android:textStyle="bold"
            android:paddingTop="6dip"
            android:paddingBottom="2dip" />
        <!-- Description label -->
        <TextView
            android:id="@+id/email"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:textColor="#acacac"
            android:paddingBottom="2dip">
        </TextView>
        <!-- Linear layout for cost and price Cost: Rs.100 -->
        <LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
            <!-- Cost Label -->
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textColor="#5d5d5d"
                android:gravity="left"
                android:textStyle="bold"
                android:text="Mobile: " >
            </TextView>
            <!-- Price Label -->
            <TextView
                android:id="@+id/mobile"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textColor="#acacac"
                android:textStyle="bold"
                android:gravity="left">
            </TextView>
        </LinearLayout>
    </LinearLayout>
</LinearLayout>
```





4. Create android xml untuk melihat informasi contact dengan nama single list item :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <!-- Name Label -->
    <TextView android:id="@+id/name_label"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="25dip"
        android:textStyle="bold"
        android:paddingTop="10dip"
        android:paddingBottom="10dip"
        android:textColor="#43bd00"/>
    <!-- Description Label -->
    <TextView android:id="@+id/email_label"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textColor="#acacac"/>
    <!-- Price Label -->
    <TextView android:id="@+id/mobile_label"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textStyle="bold"/>
</LinearLayout>
```



5. Create Class pada package src dengan nama JSONParse

```
package JSON.wartec;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONException;
import org.json.JSONObject;
```



```
import android.util.Log;
public class JSONParser {
    static InputStream is = null;
    static JSONObject jObj = null;
    static String json = "";

    // constructor
    public JSONParser() {

    }

    public JSONObject getJSONFromUrl(String url) {

        // Making HTTP request
        try {
            // defaultHttpClient
            DefaultHttpClient httpClient = new DefaultHttpClient();
            HttpPost httpPost = new HttpPost(url);

            HttpResponse httpResponse = httpClient.execute(httpPost);
            HttpEntity httpEntity = httpResponse.getEntity();
            is = httpEntity.getContent();

        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        } catch (ClientProtocolException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }

        try {
            BufferedReader reader = new BufferedReader(new InputStreamReader(
                is, "iso-8859-1"), 8);
            StringBuilder sb = new StringBuilder();
            String line = null;
            while ((line = reader.readLine()) != null) {
                sb.append(line + "\n");
            }
            is.close();
            json = sb.toString();
        } catch (Exception e) {
            Log.e("Buffer Error", "Error converting result " + e.toString());
        }

        // try parse the string to a JSON object
        try {
            jObj = new JSONObject(json);
        }
    }
}
```




```
        } catch (JSONException e) {
            Log.e("JSON Parser", "Error parsing data " + e.toString());
        }

        // return JSON String
        return jsonObj;
    }
}
```

6. Create Class pada package src dengan nama `single_list_item` :

```
package JSON.wartec;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;

public class SingleMenuItemActivity extends Activity {

    // JSON node keys
    private static final String TAG_NAME = "name";
    private static final String TAG_EMAIL = "email";
    private static final String TAG_PHONE_MOBILE = "mobile";
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.single_list_item);

        // getting intent data
        Intent in = getIntent();

        // Get JSON values from previous intent
        String name = in.getStringExtra(TAG_NAME);
        String cost = in.getStringExtra(TAG_EMAIL);
        String description = in.getStringExtra(TAG_PHONE_MOBILE);

        // Displaying all values on the screen
        TextView lblName = (TextView) findViewById(R.id.name_label);
        TextView lblCost = (TextView) findViewById(R.id.email_label);
        TextView lblDesc = (TextView) findViewById(R.id.mobile_label);

        lblName.setText(name);
        lblCost.setText(cost);
        lblDesc.setText(description);
    }
}
```



```
}
```

7. Buka Class pada src – package -JSONParseActivity.java :

```
package JSON.wartec;

import android.app.Activity;
import android.os.Bundle;
import java.util.ArrayList;
import java.util.HashMap;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import android.app.ListActivity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;

public class LatihanJSONActivity extends ListActivity implements OnClickListener{
    /** Called when the activity is first created. */
    // url to make request
    private static String url = "http://api.androidhive.info/contacts/";

    // JSON Node names
    private static final String TAG_CONTACTS = "contacts";
    private static final String TAG_ID = "id";
    private static final String TAG_NAME = "name";
    private static final String TAG_EMAIL = "email";
    private static final String TAG_ADDRESS = "address";
    private static final String TAG_GENDER = "gender";
    private static final String TAG_PHONE = "phone";
    private static final String TAG_PHONE_MOBILE = "mobile";
    private static final String TAG_PHONE_HOME = "home";
    private static final String TAG_PHONE_OFFICE = "office";

    // contacts JSONArray
    JSONArray contacts = null;

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



```
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
// Hashmap for ListView
ArrayList<HashMap<String, String>> contactList = new
ArrayList<HashMap<String, String>>();

// Creating JSON Parser instance
JSONParser jParser = new JSONParser();

// getting JSON string from URL
JSONObject json = jParser.getJSONFromUrl(url);

try {
    // Getting Array of Contacts
    contacts = json.getJSONArray(TAG_CONTACTS);

    // looping through All Contacts
    for(int i = 0; i < contacts.length(); i++){
        JSONObject c = contacts.getJSONObject(i);

        // Storing each json item in variable
        String id = c.getString(TAG_ID);
        String name = c.getString(TAG_NAME);
        String email = c.getString(TAG_EMAIL);
        String address = c.getString(TAG_ADDRESS);
        String gender = c.getString(TAG_GENDER);

        // Phone number is agin JSON Object
        JSONObject phone = c.getJSONObject(TAG_PHONE);
        String mobile = phone.getString(TAG_PHONE_MOBILE);
        String home = phone.getString(TAG_PHONE_HOME);
        String office = phone.getString(TAG_PHONE_OFFICE);

        // creating new HashMap
        HashMap<String, String> map = new HashMap<String, String>();

        // adding each child node to HashMap key => value
        map.put(TAG_ID, id);
        map.put(TAG_NAME, name);
        map.put(TAG_EMAIL, email);
        map.put(TAG_PHONE_MOBILE, mobile);

        // adding HashList to ArrayList
        contactList.add(map);
    }
} catch (JSONException e) {
    e.printStackTrace();
}
```



```

/**
 * Updating parsed JSON data into ListView
 */
ListAdapter adapter = new SimpleAdapter(this, contactList,
    R.layout.list_item,
    new String[] { TAG_NAME, TAG_EMAIL,
TAG_PHONE_MOBILE }, new int[] {
    R.id.name, R.id.email, R.id.mobile });

setListAdapter(adapter);

// selecting single ListView item
ListView lv = getListView();

// Launching new screen on Selecting Single ListItem
lv.setOnItemClickListener(new OnItemClickListener() {

    public void onItemClick(AdapterView<?> parent, View view,
        int position, long id) {
        // getting values from selected ListItem
        String name = ((TextView)
view.findViewById(R.id.name)).getText().toString();
        String cost = ((TextView)
view.findViewById(R.id.email)).getText().toString();
        String description = ((TextView)
view.findViewById(R.id.mobile)).getText().toString();

        // Starting new intent
        Intent in = new Intent(getApplicationContext(),
SingleMenuItemActivity.class);
        in.putExtra(TAG_NAME, name);
        in.putExtra(TAG_EMAIL, cost);
        in.putExtra(TAG_PHONE_MOBILE, description);
        startActivity(in);

    }

});

}

public void onClick(View arg0) {
    // TODO Auto-generated method stub

}

}

```

8. Buka file androidmanifest.xml :



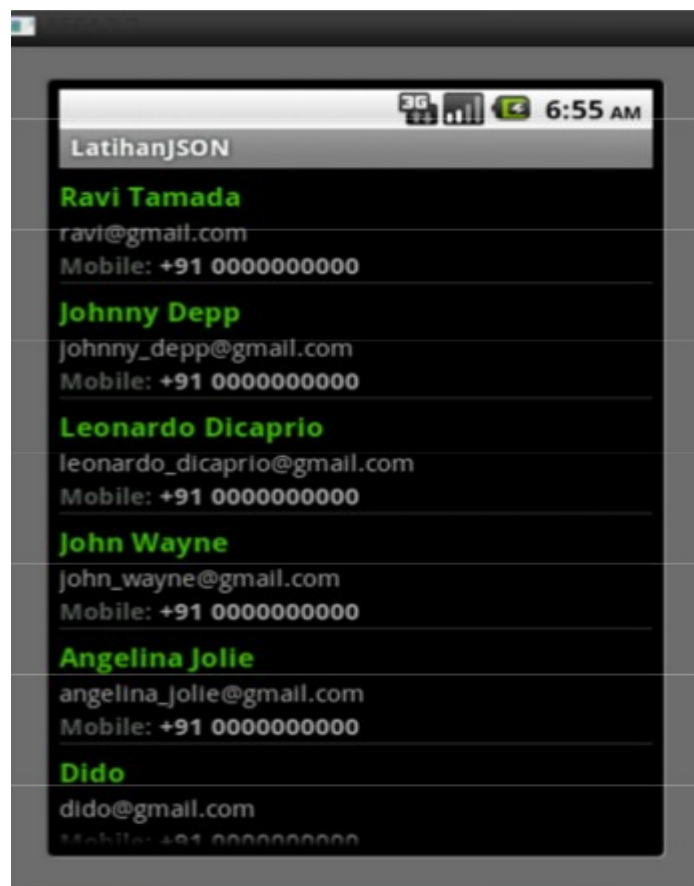
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="JSON.wartec"
    android:versionCode="1"
    android:versionName="1.0" >

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

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".LatihanJSONActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <!-- Single List Item View -->
        <activity
            android:label="Single Menu Item"
            android:name=".SingleMenuItemActivity" >
        </activity>
    </application>
<uses-permission android:name="android.permission.INTERNET" />
</manifest>
```

Run Project:





Google Map

Fitur - fitur yang di berikan oleh google sangat lah banyak , salah satunya Google Map , karena android telah di akusisi oleh Google jadi fitur yang ada di google dapat di develop semua . Google map merupakan fitur yang sangat menjadi andalan pada Google , dari Mesin pencarian , desktop , hingga mobile dapat di gunakan , dalam android Google map sudah di beri kemudahab kita tinggal meletakkan platform Map ke dalam project , karena kita tinggal download pada sdk android dan itu sudah terdapat semua fitur yang ada pada google map , jadi pada saat ingin melanjutkan kita

sebaiknya mendownload Google Map versi 2.2 atau di atas nya . Kita langsung ke lembar kerja.

1. Kita harus mendaftarkan Google Key untuk dapat persetujuan mendvelop Google Map cara nya kita harus melihat key pada Keytool pada sdk android , dengan cara membuka cmd menu+r (run) – cmd dan lakukan perintah berikut ini :

```
c:\<path-to-jdk-dir>\bin\keytool.exe -list -alias androiddebugkey
-keystore "C:\users\<user-name>\.android\debug.keystore"
-storepass android -keypass android
```

```
Contoh :c:\<path-to-jdk-dir>\bin\keytool.exe -list -alias androiddebugkey -keystore
"C:\users\wartec\.android\debug.keystore" -storepass android
-keypass android
```

Dan hasilnya seperti di bawah ini :

```
C:\Program Files (x86)\Java\jdk1.6.0_11\bin>keytool.exe -list -alias
androiddebugkey -keystore "C:\users\ravi\.android\debug.keystore"
-storepass android -keypass android
androiddebugkey, 11 Nov, 2011, PrivateKeyEntry,
Certificate fingerprint (MD5): 37:6E:5A:6E:E7:2A:A0:92:E6:2F:DF:EC:7F:10
```

dengan md5 : 37:6E:5A:6E:E&7

2. Setelah mendapatkan Google Key kita mendaftarkan ke Google dengan cara Copy Certificate Fingerprint MD5 dan create Android Google Map di sini <http://code.google.com/android/maps-api-signup.html> dan masukkan MD5 :

☒ I have read and agree with the terms and conditions ([printable version](#))

My certificate's MD5 fingerprint: 37:6E:5A:6E:E7:2A:A0:92:E6:2F:DF:EC:7F:10:02:0F

3. Dan akan mendapatkan Map Key sebagai berikut :



Setelah mendapatkan map key , kita akan menanamkan key itu ke project yang kita buat , itu di letakkan pada file xml yang di tujukan pada google.map.api

```
<?xml version="1.0" encoding="utf-8"?>
<com.google.android.maps.MapView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/mapView"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:clickable="true"
    android:apiKey="05M-7qOGbEjYduPPUdQgJt9ysL8HToawGdvu_ow"
/>
```

Sebaiknya kita langsung ke contoh aplikasi nya :

1. Create Project dengan nama AndroidGoogleMaps
2. Buka File pada res – layout – main.xml nah disini kode itu di masukkan , karena itu merupakan tampilan yang akan berjalan pada device android sendiri :

```
<?xml version="1.0" encoding="utf-8"?>
<com.google.android.maps.MapView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/mapView"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:clickable="true"
    android:apiKey="05M-7qOGbEjYduPPUdQgJt9ysL8HToawGdvu_ow"
/>
```

3. Buka File pada src – package – AndroidGoogleMapActivity.java dan tuliskan scrip di bawah ini :

```
package com.androidhive.googlemaps;

import java.util.List;
```



```
import android.graphics.drawable.Drawable;
import android.os.Bundle;

import com.google.android.maps.GeoPoint;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapController;
import com.google.android.maps.MapView;
import com.google.android.maps.Overlay;
import com.google.android.maps.OverlayItem;

public class AndroidGoogleMapsActivity extends MapActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        // Displaying Zooming controls
        MapView mapView = (MapView) findViewById(R.id.mapView);
        mapView.setBuiltInZoomControls(true);

        /**
         * Changing Map Type
         */
        // mapView.setSatellite(true); // Satellite View
        // mapView.setStreetView(true); // Street View
        // mapView.setTraffic(true); // Traffic view

        /**
         * showing location by Latitude and Longitude
         */
        MapController mc = mapView.getController();
        double lat = Double.parseDouble("48.85827758964043");
        double lon = Double.parseDouble("2.294543981552124");
        GeoPoint geoPoint = new GeoPoint((int)(lat * 1E6), (int)(lon * 1E6));
        mc.animateTo(geoPoint);
        mc.setZoom(15);
        mapView.invalidate();

        /**
         * Placing Marker
         */
        List<Overlay> mapOverlays = mapView.getOverlays();
        Drawable drawable = this.getResources().getDrawable(R.drawable.mark_red);
        AddItemizedOverlay itemizedOverlay =
            new AddItemizedOverlay(drawable, this);
```



```
OverlayItem overlayitem = new OverlayItem(geoPoint, "Hello", "Sample Overlay item");  
  
itemizedOverlay.addOverlay(overlayitem);  
mapOverlays.add(itemizedOverlay);  
  
}  
  
@Override  
protected boolean isRouteDisplayed() {  
    return false;  
}  
}
```

Run project dan hasil nya seperti di bawah ini :



Sensor

Dalam android banyak sekali sensor yang tertanam pada device android , dari sensor suara , sensor cahaya , sensor gerak , sensor panas , dan sebagainya , insya allah kita akan bahas satu persatu .

Pengenalan Sensor

Kebanyakan Android perangkat memiliki sensor yang bisa memonitor perubahan posisi perangkat dan gerak. Banyak perangkat yang juga memiliki sensor yang memungkinkan Anda menentukan kondisi lingkungan sekitar, seperti suhu, tekanan, kelembaban, dan pencahayaan. Anda dapat mengakses sensor tersebut dan memperoleh data sensor baku dengan menggunakan kerangka sensor Android. Kerangka sensor menyediakan beberapa kelas dan interface yang membantu Anda



melakukan berbagai macam sensor yang berhubungan dengan tugas. Sebagai contoh, Anda dapat menggunakan kerangka sensor untuk melakukan hal berikut: Menentukan sensor yang tersedia pada perangkat. Menentukan kemampuan sebuah sensor individu, seperti rentang maksimum, produsen, kebutuhan daya, dan resolusi. Memperoleh data sensor baku dan menentukan tingkat minimum di mana Anda mendapatkan data sensor. Daftar dan unregister pendengar acara sensor yang memantau

perubahan sensor. Topik ini memberikan gambaran dari sensor yang tersedia di platform Android. Hal ini juga menyediakan pengenalan pada kerangka sensor. Kerangka sensor Android memungkinkan Anda mengakses berbagai jenis sensor. Beberapa sensor yang berbasis hardware dan beberapa software berbasis. Hardware berbasis sensor merupakan komponen fisik dibangun ke handset atau perangkat tablet. Mereka memperoleh data mereka dengan langsung mengukur sifat lingkungan tertentu, seperti percepatan, kekuatan geomagnetik lapangan, atau perubahan sudut. Software berbasis sensor tidak perangkat fisik, meskipun mereka meniru berbasis hardware sensor. Software berbasis sensor data mereka berasal dari satu atau lebih berbasis hardware sensor dan kadang-kadang disebut virtual atau sensor sensor sintetis. Sensor percepatan linier dan sensor gravitasi adalah contoh dari perangkat lunak berbasis sensor. Tabel 1 merangkum sensor yang didukung oleh platform Android. Beberapa perangkat Android memiliki setiap jenis sensor. Sebagai contoh, perangkat handset paling dan tablet memiliki accelerometer dan magnetometer, tapi perangkat yang lebih sedikit memiliki barometer atau termometer. Juga, perangkat dapat memiliki lebih dari satu sensor dari jenis tertentu. Sebagai contoh, perangkat dapat memiliki sensor gravitasi dua, masing-masing memiliki berbagai berbeda.

Sensor Framework

Anda dapat mengakses sensor tersebut dan memperoleh data sensor baku dengan menggunakan kerangka sensor Android. Sensor Framework adalah bagian dari paket `android.hardware` dan termasuk kelas-kelas dan interface berikut:

SensorManager

Anda dapat menggunakan kelas ini untuk membuat sebuah instance dari layanan sensor. Class ini menyediakan berbagai metode untuk mengakses dan daftar sensor, mendaftar dan unregistering pendengar sensor acara, dan memperoleh informasi orientasi. Kelas ini juga menyediakan konstanta sensor beberapa yang digunakan untuk melaporkan akurasi sensor, rangkaian data harga perolehan, dan mengkalibrasi sensor.

➤ Sensor

Anda dapat menggunakan kelas ini untuk membuat sebuah instance dari sebuah sensor tertentu. Class ini menyediakan berbagai metode yang memungkinkan Anda menentukan kemampuan sebuah sensor.

➤ SensorEvent

Sistem ini menggunakan kelas ini untuk membuat objek acara sensor, yang menyediakan informasi tentang peristiwa sensor. Sebuah objek acara sensor mencakup informasi berikut: data sensor baku, jenis sensor yang dihasilkan peristiwa, akurasi data, dan timestamp untuk aktivitas tersebut.

➤ SensorEventListener



Anda dapat menggunakan interface ini untuk menciptakan metode callback dua yang menerima pemberitahuan (peristiwa sensor) ketika sensor nilai-nilai perubahan atau saat perubahan akurasi sensor. Dalam aplikasi khas Anda menggunakan API sensor terkait untuk melakukan dua tugas dasar:

➤ Mengidentifikasi sensor dan kemampuan sensor

Mengidentifikasi sensor dan kemampuan sensor pada saat runtime berguna jika aplikasi Anda memiliki fitur yang bergantung pada jenis sensor tertentu atau kemampuan. Sebagai contoh, Anda mungkin ingin mengidentifikasi semua sensor yang ada pada perangkat dan menonaktifkan fitur aplikasi apapun yang mengandalkan sensor yang tidak hadir. Demikian juga, Anda mungkin ingin mengidentifikasi semua sensor dari jenis tertentu sehingga Anda dapat memilih pelaksanaan sensor yang memiliki kinerja optimal untuk aplikasi Anda.

➤ Memeriksa semua aktivitas sensor

Pemantauan kejadian sensor adalah bagaimana Anda memperoleh data sensor mentah. Sebuah peristiwa terjadi setiap kali sensor mendeteksi perubahan parameter itu adalah mengukur. Sebuah acara sensor menyediakan empat potongan informasi: nama sensor yang memicu acara tersebut, timestamp untuk acara tersebut, akurasi acara, dan data sensor baku yang memicu acara .

Sensor Availability

Sementara ketersediaan sensor bervariasi dari perangkat ke perangkat, juga dapat bervariasi antara versi Android. Hal ini karena sensor Android telah diperkenalkan selama beberapa rilis Platform. Sebagai contoh, banyak sensor diperkenalkan di Android 1,5 (API Level 3), tetapi beberapa tidak dilaksanakan dan tidak tersedia untuk digunakan sampai Android 2.3 (API Tingkat 9). Demikian juga, beberapa sensor diperkenalkan pada Android 2.3 (API Tingkat 9) dan Android 4.0 (API Tingkat 14). Dua sensor telah usang dan diganti dengan yang lebih baru, sensor lebih baik.

Tabel dibawah merangkum ketersediaan setiap sensor secara platform-oleh- platform. Hanya empat platform terdaftar karena mereka adalah platform yang melibatkan perubahan sensor. Sensor yang terdaftar sebagai usang yang masih tersedia pada platform berikutnya (disediakan sensor hadir pada perangkat), yang sejalan dengan kebijakan kompatibilitas ke depan Android.

Table Dibawah . Sensor availability by platform. :



| Sensor | Android 4.0 (API Level 14) | Android 2.3 (API Level 9) | Android 2.2 (API Level 8) | Android 1.5 (API Level 3) |
|---|-------------------------------|------------------------------|------------------------------|------------------------------|
| <u>TYPE_ACCELEROMETER</u> | Yes | Yes | Yes | Yes |
| <u>TYPE_AMBIENT_TEMPERATURE</u> | Yes | n/a | n/a | n/a |
| <u>TYPE_GRAVITY</u> | Yes | Yes | n/a | n/a |
| <u>TYPE_GYROSCOPE</u> | Yes | Yes | n/a ¹ | n/a ¹ |
| <u>TYPE_LIGHT</u> | Yes | Yes | Yes | Yes |
| <u>TYPE_LINEAR_ACCELERATION</u> | Yes | Yes | n/a | n/a |
| <u>TYPE_MAGNETIC_FIELD</u> | Yes | Yes | Yes | Yes |
| <u>TYPE_ORIENTATION</u> | Yes ² | Yes ² | Yes ² | Yes |
| <u>TYPE_PRESSURE</u> | Yes | Yes | n/a ¹ | n/a ¹ |
| <u>TYPE_PROXIMITY</u> | Yes | Yes | Yes | Yes |
| <u>TYPE_RELATIVE_HUMIDITY</u> | Yes | n/a | n/a | n/a |
| <u>TYPE_ROTATION_VECTOR</u> | Yes | Yes | n/a | n/a |
| <u>TYPE_TEMPERATURE</u> | Yes ² | Yes | Yes | Yes |

➤ Mengidentifikasi Sensor dan Sensor Kemampuan

Kerangka sensor Android menyediakan beberapa metode yang membuatnya mudah bagi Anda untuk menentukan pada saat runtime yang sensor pada perangkat. API juga menyediakan metode yang memungkinkan Anda menentukan kemampuan setiap sensor, seperti rentang maksimum, resolusi, dan persyaratan kekuatannya. Untuk mengidentifikasi sensor yang ada di perangkat Anda harus terlebih dahulu mendapatkan referensi ke layanan sensor. Untuk melakukan ini, Anda membuat sebuah instance dari kelas `SensorManager` dengan memanggil `getSystemService ()` metode dan lulus dalam argumen `SENSOR_SERVICE`. Sebagai contoh:

```
private SensorManager mSensorManager;
...
mSensorManager =
(SensorManager)getSystemService(Context.SENSOR_SERVICE);
```

Selanjutnya, Anda dapat mendapatkan daftar dari setiap sensor pada perangkat dengan memanggil `getSensorList ()` metode dan menggunakan konstan `TYPE_ALL`. Sebagai contoh:

```
List<Sensor> deviceSensors = mSensorManager.getSensorList(Sensor.TYPE_ALL);
```

Jika Anda ingin daftar semua sensor dari jenis tertentu, Anda bisa menggunakan yang lain konstan, bukan `TYPE_ALL` seperti `TYPE_GYROSCOPE`, `TYPE_LINEAR_ACCELERATION`, atau `TYPE_GRAVITY`. Anda juga dapat menentukan apakah jenis tertentu dari sensor ada pada perangkat dengan menggunakan `getDefaultSensor ()` metode dan lulus dalam jenis konstan untuk sensor tertentu. Jika perangkat memiliki lebih dari satu sensor dari jenis tertentu, salah satu sensor harus ditunjuk sebagai sensor standar. Jika sensor default tidak ada untuk jenis tertentu dari sensor, pemanggilan metode mengembalikan null, yang berarti perangkat tidak memiliki jenis sensor. Sebagai contoh, kode berikut akan memeriksa apakah ada magnetometer pada perangkat:



```
private SensorManager mSensorManager;

...
mSensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
if (mSensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD) != null)
{
    // Success! There's a magnetometer.
}
else {
    // Failure! No magnetometer.
}
```

Selain daftar sensor yang ada di perangkat, Anda dapat menggunakan metode publik dari kelas `Sensor` untuk menentukan kemampuan dan atribut sensor individu. Ini berguna jika Anda ingin aplikasi Anda untuk berperilaku berbeda berdasarkan yang sensor atau kemampuan sensor yang tersedia pada perangkat. Misalnya, Anda dapat menggunakan `getResolution ()` dan `getMaximumRange ()` metode untuk mendapatkan resolusi sensor dan jangkauan maksimum pengukuran. Anda juga dapat menggunakan `getPower ()` metode untuk mendapatkan kebutuhan daya sebuah sensor. Dua dari metode publik sangat berguna jika Anda ingin mengoptimalkan aplikasi Anda untuk sensor produsen yang berbeda atau versi yang berbeda dari sensor. Sebagai contoh, jika aplikasi Anda perlu memonitor gerak tubuh pengguna seperti miring dan goyang, Anda dapat membuat satu set data aturan penyaringan dan optimasi untuk perangkat yang lebih baru yang memiliki sensor gravitasi vendor tertentu, dan satu set aturan penyaringan data dan optimasi untuk perangkat yang tidak memiliki sensor gravitasi dan hanya memiliki accelerometer. Contoh kode berikut menunjukkan Anda bagaimana Anda dapat menggunakan metode yang `getVendor ()` dan `getVersion ()` untuk melakukan hal ini. Dalam sampel ini, kami sedang mencari sensor gravitasi yang berisi daftar Google Inc sebagai vendor dan memiliki nomor versi 3. Jika sensor tertentu tidak terdapat pada perangkat, kita mencoba untuk menggunakan accelerometer.

```
private SensorManager mSensorManager;
private Sensor mSensor;

...
mSensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
if (mSensorManager.getDefaultSensor(Sensor.TYPE_GRAVITY) != null){
    List<Sensor> gravSensors =
    mSensorManager.getSensorList(Sensor.TYPE_GRAVITY);
    for(int i=0; i<gravSensors.size(); i++) {
        if ((gravSensors.get(i).getVendor().contains("Google Inc. ")) &&
            (gravSensors.get(i).getVersion() == 3)){
            // Use the version 3 gravity sensor.
            mSensor = gravSensors.get(i);
        }
    }
}
else{
    // Use the accelerometer.
}
```




```

if (mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER) !=
null){
mSensor =
mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
}
else{
// Sorry, there are no accelerometers on your device.
// You can't play this game.
}
}
}

```

Monitoring Sensor Events

Untuk memonitor data sensor baku Anda perlu untuk mengimplementasikan metode callback dua yang terpapar melalui antarmuka `SensorEventListener`: `onAccuracyChanged ()` dan `onSensorChanged ()`. Sistem Android memanggil metode ini setiap kali berikut terjadi:

➤ Sebuah sensor akurasi perubahan.

Dalam hal ini sistem memanggil method `onAccuracyChanged ()`, menyediakan Anda dengan referensi ke objek `Sensor` yang mengubah dan akurasi baru dari sensor. Akurasi diwakili oleh salah satu dari empat konstanta status: `SENSOR_STATUS_ACCURACY_LOW`, `SENSOR_STATUS_ACCURACY_MEDIUM`, `SENSOR_STATUS_ACCURACY_HIGH`, atau `SENSOR_STATUS_UNRELIABLE`.

➤ Sebuah sensor melaporkan nilai baru.

Dalam hal ini sistem memanggil method `onSensorChanged ()`, memberikan Anda dengan benda `SensorEvent`. Sebuah objek `SensorEvent` berisi informasi tentang data sensor baru, termasuk: keakuratan data, sensor yang dihasilkan data, timestamp di mana data yang dihasilkan, dan data baru bahwa sensor dicatat. Kode berikut menunjukkan bagaimana menggunakan metode `onSensorChanged ()` untuk memantau data dari sensor cahaya. Contoh ini menampilkan data sensor baku dalam `TextView` yang didefinisikan dalam file `main.xml` sebagai `sensor_data`.

```

public class SensorActivity extends Activity implements SensorEventListener {
private SensorManager mSensorManager;
private Sensor mLight;
@Override
public final void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
mLight = mSensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);
}
@Override
public final void onAccuracyChanged(Sensor sensor, int accuracy) {
// Do something here if sensor accuracy changes.
}
}

```




```
@Override
public final void onSensorChanged(SensorEvent event) {
// The light sensor returns a single value.
// Many sensors return 3 values, one for each axis.
float lux = event.values[0];
// Do something with this sensor value.
}
@Override
protected void onResume() {
super.onResume();
mSensorManager.registerListener(this, mLight,
SensorManager.SENSOR_DELAY_NORMAL);
}
@Override
protected void onPause() {
super.onPause();
mSensorManager.unregisterListener(this);
}
}
```

Dalam contoh ini, data default delay (SENSOR_DELAY_NORMAL) ditetapkan ketika registerListener () method dipanggil. Keterlambatan data (atau sampling rate) mengatur interval dimana peristiwa sensor dikirim ke aplikasi Anda melalui metode callback onSensorChanged (). Penundaan data standar cocok untuk memantau perubahan orientasi layar khas dan menggunakan penundaan 200.000 mikrodetik. Anda dapat menentukan penundaan data lain, seperti SENSOR_DELAY_GAME (20.000 penundaan mikrodetik), SENSOR_DELAY_UI (60.000 penundaan mikrodetik), atau SENSOR_DELAY_FASTEST (0 penundaan mikrodetik). Pada Android 3.0 (API Tingkat 11) Anda juga dapat menentukan penundaan sebagai nilai absolut (dalam mikrodetik). Penundaan yang Anda tetapkan hanya penundaan yang disarankan. Sistem Android dan aplikasi lain dapat mengubah keterlambatan ini. Sebagai praktik terbaik, Anda harus menentukan delay terbesar yang Anda dapat karena sistem biasanya menggunakan delay yang lebih kecil dari yang Anda tentukan (yaitu, Anda harus memilih sampling rate paling lambat yang masih memenuhi kebutuhan aplikasi Anda). Menggunakan penundaan lebih besar membebankan beban yang lebih rendah pada prosesor dan karena itu menggunakan daya yang lebih kecil. Tidak ada metode umum untuk menentukan tingkat di mana kerangka sensor mengirim peristiwa sensor untuk aplikasi Anda, namun Anda dapat menggunakan cap yang berkaitan dengan setiap aktivitas sensor untuk menghitung laju sampling atas beberapa acara. Anda tidak harus mengubah sampling rate (menunda) setelah Anda mengaturnya. Jika karena alasan tertentu Anda perlu mengubah penundaan itu, Anda harus unregister dan reregister pendengar sensor. Ini juga penting untuk dicatat bahwa contoh ini menggunakan metode yang onResume () dan onPause () panggilan balik untuk mendaftar dan membatalkan registrasi pendengar acara sensor. Sebagai praktik terbaik Anda harus selalu menonaktifkan sensor Anda tidak perlu, terutama bila aktivitas Anda dihentikan sementara. Kegagalan untuk melakukannya dapat menguras baterai hanya dalam beberapa jam karena beberapa sensor memiliki kebutuhan daya yang cukup besar dan bisa menghabiskan daya baterai dengan cepat. Sistem tidak akan menonaktifkan sensor secara otomatis ketika layarnya akan mati.

Handling Different Sensor Configurations



Android tidak menentukan konfigurasi standar untuk perangkat sensor, yang berarti produsen perangkat dapat menggabungkan konfigurasi sensor yang mereka inginkan ke Android perangkat mereka. Akibatnya, perangkat dapat mencakup berbagai sensor dalam berbagai konfigurasi. Sebagai contoh, Xoom Motorola memiliki sensor tekanan, tetapi Samsung Nexus S tidak. Demikian pula, Xoom dan Nexus S memiliki giroskop, tapi Nexus HTC Satu tidak. Jika aplikasi Anda bergantung pada jenis tertentu dari sensor, anda harus memastikan bahwa sensor hadir pada perangkat sehingga aplikasi Anda dapat berjalan dengan sukses. Anda memiliki dua pilihan untuk memastikan bahwa sensor diberikan hadir pada perangkat:

- Gunakan Google Putar filter untuk perangkat target dengan konfigurasi sensor tertentu.
- Mendeteksi sensor pada runtime dan mengaktifkan atau menonaktifkan fitur-fitur aplikasi yang sesuai.

Setiap opsi dibahas dalam bagian berikut.

➤ Detecting sensors at runtime

Jika aplikasi Anda menggunakan jenis tertentu dari sensor, tetapi tidak bergantung pada itu, Anda dapat menggunakan kerangka sensor untuk mendeteksi sensor pada saat runtime dan kemudian menonaktifkan atau mengaktifkan fitur aplikasi yang sesuai. Sebagai contoh, sebuah aplikasi navigasi mungkin menggunakan sensor suhu, sensor tekanan, sensor GPS, dan sensor geomagnetik lapangan untuk menampilkan temperatur, tekanan udara, lokasi, dan bantalan kompas. Jika perangkat tidak memiliki sensor tekanan, Anda dapat menggunakan kerangka sensor untuk mendeteksi adanya sensor tekanan pada saat runtime dan kemudian menonaktifkan porsi UI aplikasi Anda yang menampilkan tekanan. Sebagai contoh, kode berikut akan memeriksa apakah ada sensor tekanan pada perangkat:

```
private SensorManager mSensorManager;
...
mSensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
if (mSensorManager.getDefaultSensor(Sensor.TYPE_PRESSURE) != null){
// Success! There's a pressure sensor.
}
else {
// Failure! No pressure sensor.
Using Google Play filters to target specific sensor configurations
}
```

➤ Using Google Play filters to target specific sensor configurations

Jika Anda mempublikasikan aplikasi Anda Play Google Anda dapat menggunakan elemen <uses-feature> dalam file manifest Anda untuk menyaring aplikasi Anda dari perangkat yang tidak memiliki konfigurasi sensor yang sesuai untuk aplikasi Anda. Unsur <uses-feature> memiliki deskriptor beberapa hardware yang memungkinkan Anda aplikasi penyaring berdasarkan adanya sensor tertentu. Sensor Anda bisa daftar meliputi: accelerometer, barometer, kompas (geomagnetic lapangan), giroskop, cahaya, dan kedekatan. Berikut ini adalah contoh nyata entri yang menyaring aplikasi yang tidak memiliki accelerometer:

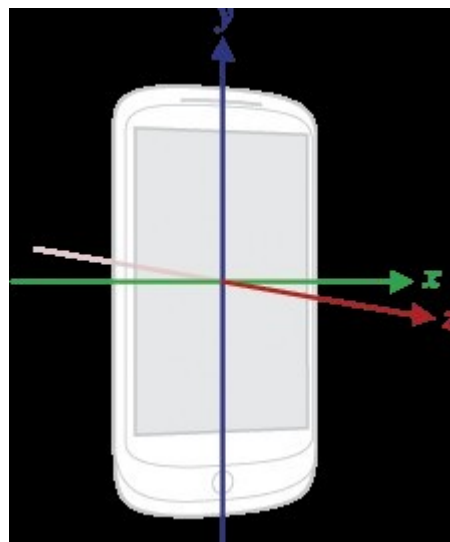


```
<uses-feature android:name="android.hardware.sensor.accelerometer"
android:required="true" />
```

Jika Anda menambahkan elemen dan deskriptor untuk mewujudkan aplikasi Anda, pengguna akan melihat aplikasi Anda di Google Bermain hanya jika perangkat mereka memiliki accelerometer. Anda harus menetapkan deskriptor untuk android: dibutuhkan = "true" hanya jika aplikasi Anda bergantung sepenuhnya pada sensor tertentu. Jika aplikasi Anda menggunakan sebuah sensor untuk beberapa fungsionalitas, tapi masih berjalan tanpa sensor, Anda harus daftar sensor dalam elemen `<uses-feature>`, tetapi mengatur deskriptor untuk android: dibutuhkan = "false". Hal ini membantu memastikan bahwa perangkat dapat menginstal aplikasi Anda, bahkan jika mereka tidak memiliki sensor tertentu. Ini juga manajemen proyek praktek terbaik yang membantu Anda melacak fitur aplikasi Anda menggunakan. Perlu diingat, jika aplikasi Anda menggunakan sensor tertentu, tetapi masih berjalan tanpa sensor, maka Anda harus mendeteksi sensor pada runtime dan menonaktifkan atau mengaktifkan fitur aplikasi yang sesuai.

Sensor Coordinate System

Secara umum, kerangka sensor menggunakan sistem 3-sumbu koordinat standar untuk mengekspresikan nilai-nilai data. Untuk sensor paling, sistem koordinat didefinisikan relatif terhadap layar perangkat bila perangkat dipegang dalam orientasi default (lihat gambar 1). Ketika sebuah perangkat dipegang dalam orientasi default, sumbu X adalah horizontal dan poin ke kanan, sumbu Y adalah vertikal dan titik, dan titik- titik sumbu Z ke arah luar wajah layar. Dalam sistem ini, koordinat belakang layar memiliki nilai negatif Z. Sistem koordinat yang digunakan oleh sensor berikut:



(Coordinate system (relative to a device) that's used by the Sensor API.)

- Acceleration sensor
- Gravity sensor
- Gyroscope
- Linear acceleration sensor
- Geomagnetic field sensor



Hal yang paling penting untuk memahami tentang sistem koordinat adalah bahwa sumbu tidak bertukar ketika orientasi layar perangkat perubahan- yaitu, sistem koordinat sensor tidak pernah berubah sebagai perangkat bergerak. Perilaku ini adalah sama dengan perilaku sistem koordinat OpenGL. Hal lain yang perlu dipahami adalah bahwa aplikasi Anda tidak harus mengasumsikan bahwa alam sebuah perangkat (default) adalah orientasi potret. Orientasi alami untuk perangkat tablet banyak adalah lansekap. Dan sistem koordinat sensor selalu didasarkan pada orientasi alami perangkat. Akhirnya, jika aplikasi Anda sesuai dengan data sensor untuk tampilan di layar, Anda perlu menggunakan `getRotation ()` metode untuk menentukan rotasi layar, dan kemudian menggunakan `remapCoordinateSystem ()` metode untuk sensor peta koordinat ke koordinat layar. Anda perlu melakukan ini bahkan jika Anda menentukan potret nyata-hanya tampilan. Untuk informasi lebih lanjut tentang sistem koordinat sensor, termasuk informasi tentang bagaimana menangani rotasi layar, lihat Satu Hidupkan Layar Deserves lain. Catatan: Beberapa sensor dan metode menggunakan sistem koordinat yang relatif ke bingkai dunia dari referensi (yang bertentangan dengan kerangka perangkat dari referensi). Sensor ini dan metode mengembalikan data yang mewakili gerakan perangkat atau perangkat posisi relatif terhadap bumi. Untuk informasi lebih lanjut, lihat metode `getOrientation ()`, yang `getRotationMatrix ()` metode, Sensor Orientasi, dan Sensor Rotasi Vector.

Best Practices for Accessing and Using Sensors

Ketika Anda merancang implementasi sensor Anda, pastikan untuk mengikuti petunjuk yang dibahas dalam bagian ini. Pedoman ini direkomendasikan praktik terbaik untuk siapa saja yang menggunakan kerangka sensor untuk mengakses dan memperoleh data sensor. Unregister pendengar sensor Pastikan untuk unregister pendengar sensor ketika Anda selesai menggunakan sensor atau aktivitas sensor jeda. Jika pendengar sensor terdaftar dan kegiatannya dihentikan sementara, sensor akan terus mendapatkan data dan menggunakan sumber daya baterai kecuali Anda membatalkan registrasi sensor. Kode berikut memperlihatkan bagaimana menggunakan `onPause ()` metode untuk unregister pendengar:

```
private SensorManager mSensorManager;
...
@Override
protected void onPause() {
    super.onPause();
    mSensorManager.unregisterListener(this);
}
```

Untuk informasi lebih lanjut, lihat `unregisterListener(SensorEventListener)`

➤ Jangan menguji kode Anda pada emulator

Saat ini Anda tidak dapat menguji kode sensor pada emulator karena emulator tidak dapat mengemulasi sensor. Anda harus menguji kode sensor Anda pada perangkat fisik. Namun demikian, sensor simulator yang dapat Anda gunakan untuk mensimulasikan output sensor.

➤ Jangan memblokir metode `onSensorChanged ()`

Data sensor dapat mengubah pada tingkat tinggi, yang berarti sistem dapat memanggil metode



(SensorEvent) `onSensorChanged` cukup sering. Sebagai praktik terbaik, Anda harus melakukan sesedikit mungkin dalam metode (SensorEvent) `onSensorChanged` sehingga Anda tidak memblokirnya. Jika aplikasi Anda mengharuskan Anda untuk melakukan penyaringan data atau pengurangan data sensor, Anda harus melakukan pekerjaan yang luar metode (SensorEvent) `onSensorChanged`.

➤ Hindari menggunakan metode usang atau jenis sensor

Beberapa metode dan konstanta telah usang. Secara khusus, jenis sensor `TYPE_ORIENTATION` telah usang. Untuk mendapatkan data orientasi Anda harus menggunakan `getOrientation ()` metode sebagai gantinya. Demikian juga, jenis sensor `TYPE_TEMPERATURE` telah usang. Anda harus menggunakan jenis sensor `TYPE_AMBIENT_TEMPERATURE` hanya pada perangkat yang menjalankan Android 4.0.

➤ Pastikan sensor sebelum Anda menggunakannya

Selalu memverifikasi bahwa sensor tersebut ada pada perangkat sebelum Anda mencoba untuk memperoleh data dari itu. Jangan berasumsi bahwa sensor tersebut ada hanya karena itu sensor yang sering digunakan. Produsen perangkat tidak diharuskan untuk memberikan sensor tertentu dalam perangkat mereka.

➤ Pilih penundaan sensor hati-hati

Bila Anda mendaftarkan sensor dengan metode `registerListener ()`, pastikan Anda memilih tingkat pengiriman yang sesuai untuk aplikasi Anda atau penggunaan kasus. Sensor dapat memberikan data pada tingkat yang sangat tinggi. Memungkinkan sistem untuk mengirim data tambahan bahwa Anda tidak membutuhkan sumber daya sistem limbah dan menggunakan daya baterai.

Motion Sensor

Platform Android menyediakan beberapa sensor yang bisa memonitor gerakan perangkat. Dua sensor ini selalu berbasis hardware (akselerometer dan giroskop), dan tiga sensor ini dapat berupa hardware-based atau berbasis software (gravitasi, percepatan linier, dan sensor rotasi vektor). Sebagai contoh, pada beberapa perangkat lunak berbasis sensor data mereka berasal dari akselerometer dan magnetometer, tapi pada perangkat lain mereka juga dapat menggunakan giroskop untuk mendapatkan data mereka. Kebanyakan Android perangkat memiliki akselerometer, dan banyak sekarang termasuk giroskop. Keberadaan software berbasis sensor lebih bervariasi karena mereka sering bergantung pada satu atau lebih sensor hardware untuk mendapatkan data mereka. Sensor gerak yang berguna untuk gerakan pemantauan perangkat, seperti miring, goyang, rotasi miring, atau. Gerakan ini biasanya merupakan refleksi dari masukan pengguna langsung (misalnya, pengguna kemudi mobil dalam permainan atau pengguna mengontrol bola dalam permainan), tetapi juga dapat menjadi refleksi dari lingkungan fisik di mana perangkat yang duduk (misalnya, bergerak dengan Anda saat Anda mengemudikan mobil Anda). Dalam kasus pertama, Anda sedang memantau gerak relatif terhadap frame perangkat acuan atau frame aplikasi Anda dari referensi; dalam kasus kedua anda monitor relatif gerak ke frame dunia dari referensi. Gerak sensor sendiri tidak biasanya digunakan untuk memantau posisi perangkat, tetapi mereka dapat digunakan dengan sensor lain, seperti sensor medan geomagnetik, untuk menentukan posisi perangkat relatif terhadap frame dunia dari referensi (lihat Sensor Posisi untuk informasi lebih lanjut). Semua sensor gerak kembali multi-dimensi array dari nilai-nilai sensor untuk setiap `SensorEvent`. Sebagai contoh, selama acara sensor tunggal akselerometer kembali data



percepatan berlaku untuk tiga sumbu koordinat, dan giroskop mengembalikan tingkat data rotasi untuk tiga sumbu koordinat. Nilai-nilai data yang dikembalikan dalam array float (nilai) bersama dengan parameter SensorEvent lainnya. Tabel 1 merangkum sensor gerak yang tersedia di platform Android.

Vektor rotasi sensor dan sensor gravitasi adalah sensor yang paling sering digunakan untuk mendeteksi gerakan dan pemantauan. Sensor vektor rotasi sangat serbaguna dan dapat digunakan untuk berbagai gerak yang berhubungan dengan tugas, seperti mendeteksi gerak tubuh, memantau perubahan sudut, dan memantau perubahan orientasi relatif. Sebagai contoh, sensor rotasi vektor sangat ideal jika Anda sedang mengembangkan sebuah game, aplikasi augmented reality, kompas 2-dimensi atau 3-dimensi, atau aplikasi stabilisasi kamera. Dalam kebanyakan kasus, menggunakan sensor ini adalah pilihan yang lebih baik daripada menggunakan accelerometer dan sensor medan geomagnetik atau sensor orientasi.

- Android Open Source Project Sensors

Android Open Source Project (AOSP) menyediakan tiga sensor gerak berbasis software: sensor gravitasi, sensor percepatan linier, dan sensor vektor rotasi. Sensor ini diperbarui pada Android 4.0 dan sekarang menggunakan giroskop perangkat (selain sensor lain) untuk meningkatkan stabilitas dan kinerja. Jika Anda ingin mencoba sensor ini, Anda dapat mengidentifikasi mereka dengan menggunakan `getVendor()` metode dan metode `getVersion()` (vendor adalah Google Inc; nomor versi 3). Mengidentifikasi sensor ini dengan nomor vendor dan versi ini diperlukan karena sistem Android menganggap ketiga sensor menjadi sensor sekunder. Sebagai contoh, jika produsen perangkat menyediakan sensor gravitasi mereka sendiri, maka sensor gravitasi AOSP muncul sebagai sensor gravitasi sekunder. Ketiga sensor mengandalkan giroskop: jika perangkat tidak memiliki giroskop, sensor ini tidak muncul dan tidak tersedia untuk digunakan....

- Using the Accelerometer

Sebuah sensor percepatan mengukur percepatan yang diterapkan untuk perangkat, termasuk gaya gravitasi. Kode berikut menunjukkan cara untuk mendapatkan contoh dari sensor percepatan default:

```
private SensorManager mSensorManager;  
private Sensor mSensor;  
...  
mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);  
mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
```

Secara konseptual, sebuah sensor akselerasi menentukan percepatan yang diterapkan ke perangkat (Ad) dengan mengukur kekuatan yang diterapkan pada sensor itu sendiri (Fs) dengan menggunakan hubungan berikut:

$$A_d = - \sum F_s / \text{mass}$$

Namun, gaya gravitasi selalu mempengaruhi percepatan diukur sesuai dengan hubungan berikut:

$$A_d = -g - \sum F / \text{mass}$$



Untuk alasan ini, saat perangkat sedang duduk di atas meja (dan bukan percepatan), accelerometer membaca besarnya $g = 9,81 \text{ m/s}^2$. Begitu pula saat perangkat dalam terjun bebas dan karena itu cepat mempercepat menuju tanah di $9,81 \text{ m/s}^2$, accelerometer yang membaca besarnya $g = 0 \text{ m/s}^2$. Oleh karena itu, untuk mengukur percepatan nyata dari perangkat, kontribusi gaya gravitasi harus dihapus dari data accelerometer. Hal ini dapat dicapai dengan menerapkan filter high-pass. Sebaliknya, low-pass filter dapat digunakan untuk mengisolasi gaya gravitasi. Contoh berikut menunjukkan bagaimana Anda dapat melakukan ini:

```
public void onSensorChanged(SensorEvent event){
// In this example, alpha is calculated as t / (t + dT),
// where t is the low-pass filter's time-constant and
// dT is the event delivery rate.
final float alpha = 0.8;
// Isolate the force of gravity with the low-pass filter.
gravity[0] = alpha * gravity[0] + (1 - alpha) * event.values[0];
gravity[1] = alpha * gravity[1] + (1 - alpha) * event.values[1];
gravity[2] = alpha * gravity[2] + (1 - alpha) * event.values[2];
// Remove the gravity contribution with the high-pass filter.
linear_acceleration[0] = event.values[0] - gravity[0];
linear_acceleration[1] = event.values[1] - gravity[1];
linear_acceleration[2] = event.values[2] - gravity[2];
}
```

Catatan: Anda dapat menggunakan teknik yang berbeda untuk menyaring data sensor. Sampel kode di atas menggunakan konstan penyaring praktis (alfa) untuk membuat low-pass filter. Konstan ini filter berasal dari konstanta waktu (t), yang merupakan representasi kasar latency bahwa filter menambah peristiwa sensor, dan acara pengiriman sensor rate (dt). Sampel kode menggunakan nilai alpha sebesar 0,8 untuk tujuan demonstrasi. Jika Anda menggunakan metode penyaringan Anda mungkin perlu memilih nilai alpha yang berbeda. Accelerometers menggunakan sistem sensor koordinat standar. Dalam prakteknya, ini berarti bahwa kondisi berikut berlaku bila perangkat adalah meletakkan datar di atas meja dalam orientasi alam: Jika Anda mendorong perangkat di sisi kiri (sehingga bergerak ke kanan), nilai percepatan x adalah positif. Jika Anda mendorong perangkat di bagian bawah (jadi bergerak menjauh dari Anda), nilai percepatan y adalah positif. Jika Anda mendorong perangkat ke langit dengan percepatan $\text{m/s}^2 A$, nilai percepatan z sama dengan $9,81 A$, yang sesuai dengan percepatan perangkat ($A \text{ m/s}^2$) dikurangi gaya gravitasi ($-9,81 \text{ m/s}^2$). Perangkat stasioner akan memiliki nilai percepatan $9,81$, yang sesuai dengan percepatan perangkat (0 m/s^2 minus gaya gravitasi, yang merupakan $-9,81 \text{ m/s}^2$). Secara umum, accelerometer adalah sebuah sensor yang baik untuk digunakan jika Anda sedang memantau gerakan perangkat. Hampir setiap Android handset dan tablet memiliki accelerometer, dan menggunakan daya sekitar 10 kali lebih kecil dari sensor gerak lainnya. Satu kelemahan adalah bahwa Anda mungkin harus menerapkan filter low-pass dan high-pass untuk menghilangkan gaya gravitasi dan mengurangi kebisingan. SDK Android memberikan contoh aplikasi yang menunjukkan bagaimana menggunakan sensor percepatan (Play Accelerometer).

Gravity Sensor

Sensor gravitasi menyediakan vektor dimensi tiga yang menunjukkan arah dan besarnya gravitasi. Kode berikut menunjukkan cara untuk mendapatkan contoh dari sensor gravitasi default:



```
private SensorManager mSensorManager;  
private Sensor mSensor;  
...  
mSensorManager = (SensorManager)  
getSystemService(Context.SENSOR_SERVICE);  
mSensor =  
mSensorManager.getDefaultSensor(Sensor.TYPE_GRAVITY);
```

Unit-unit yang sama dengan yang digunakan oleh sensor percepatan (m/s^2), dan sistem koordinat adalah sama dengan yang digunakan oleh sensor percepatan. Catatan: Bila perangkat sedang beristirahat, output dari sensor gravitasi harus identik dengan accelerometer.

Gyroscope

Giroskop mengukur tingkat atau rotasi dalam rad/s sekitar x sebuah perangkat, y, dan sumbu z. Kode berikut menunjukkan cara untuk mendapatkan sebuah instance dari giroskop default:

```
private SensorManager mSensorManager;  
private Sensor mSensor;  
...  
mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);  
mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE_GYROSCOPE);
```

Sistem koordinat sensor adalah sama dengan yang digunakan untuk sensor percepatan. Rotasi adalah positif dalam arah berlawanan arah jarum jam, yaitu, seorang pengamat melihat dari beberapa lokasi yang positif terhadap x, y atau sumbu z pada perangkat diposisikan pada asal akan melaporkan rotasi positif jika perangkat tampaknya berputar berlawanan arah jarum jam. Ini adalah definisi matematika standar rotasi positif dan tidak sama dengan definisi untuk gulungan yang digunakan oleh sensor orientasi. Biasanya, output dari giroskop yang terintegrasi dari waktu ke waktu untuk menghitung rotasi menggambarkan perubahan sudut lebih timestep tersebut. Sebagai contoh:...

```
// Create a constant to convert nanoseconds to seconds.  
private static final float NS2S = 1.0f / 1000000000.0f;  
private final float[] deltaRotationVector = new float[4]();  
private float timestamp;  
public void onSensorChanged(SensorEvent event) {  
    // This timestep's delta rotation to be multiplied by the current rotation  
    // after computing it from the gyro sample data.  
    if (timestamp != 0) {  
        final float dT = (event.timestamp - timestamp) * NS2S;  
        // Axis of the rotation sample, not normalized yet.  
        float axisX = event.values[0];  
        float axisY = event.values[1];  
        float axisZ = event.values[2];  
        // Calculate the angular speed of the sample
```



```

float omegaMagnitude = sqrt(axisX*axisX + axisY*axisY + axisZ*axisZ);
// Normalize the rotation vector if it's big enough to get the axis
// (that is, EPSILON should represent your maximum allowable margin of error)
if (omegaMagnitude > EPSILON) {
    axisX /= omegaMagnitude;
    axisY /= omegaMagnitude;
    axisZ /= omegaMagnitude;
}
// Integrate around this axis with the angular speed by the timestep
// in order to get a delta rotation from this sample over the timestep
// We will convert this axis-angle representation of the delta rotation
// into a quaternion before turning it into the rotation matrix.
float thetaOverTwo = omegaMagnitude * dT / 2.0f;
float sinThetaOverTwo = sin(thetaOverTwo);
float cosThetaOverTwo = cos(thetaOverTwo);
deltaRotationVector[0] = sinThetaOverTwo * axisX;
deltaRotationVector[1] = sinThetaOverTwo * axisY;
deltaRotationVector[2] = sinThetaOverTwo * axisZ;
deltaRotationVector[3] = cosThetaOverTwo;
}
timestamp = event.timestamp;
float[] deltaRotationMatrix = new float[9];
SensorManager.getRotationMatrixFromVector(deltaRotationMatrix, deltaRotationVector);
// User code should concatenate the delta rotation we computed with the current rotation
// in order to get the updated rotation.
// rotationCurrent = rotationCurrent * deltaRotationMatrix;
}
}

```

Giroskop Standar menyediakan data rotasi mentah tanpa penyaringan atau koreksi untuk kebisingan dan drift (bias). Dalam prakteknya, giroskop kebisingan dan drift akan memperkenalkan kesalahan yang perlu dikompensasi. Anda biasanya menentukan penyimpangan (bias) dan kebisingan dengan memantau sensor lain, seperti sensor gravitasi atau accelerometer.

Linear Accelerometer

Sensor percepatan linear menyediakan Anda dengan vektor percepatan tiga dimensi yang mewakili masing-masing sepanjang sumbu perangkat, tidak termasuk gravitasi. Kode berikut menunjukkan cara untuk mendapatkan contoh dari sensor percepatan standar linear:

```

private SensorManager mSensorManager;
private Sensor mSensor;
...
mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE_LINEAR_ACCELERATION);

```




Secara konseptual, sensor ini menyediakan Anda dengan data percepatan sesuai dengan hubungan berikut:

$$\text{linear acceleration} = \text{acceleration} - \text{acceleration due to gravity}$$

Anda biasanya menggunakan sensor ini bila Anda ingin memperoleh data percepatan tanpa pengaruh gravitasi. Misalnya, Anda bisa menggunakan sensor ini untuk melihat seberapa cepat mobil Anda pergi. Sensor percepatan linear selalu telah offset, yang Anda butuhkan untuk menghapus. Cara paling mudah untuk melakukan ini adalah untuk membangun langkah kalibrasi ke dalam aplikasi Anda. Selama kalibrasi Anda dapat meminta pengguna untuk mengatur perangkat di atas meja, dan kemudian membaca offset untuk ketiga sumbu. Anda kemudian dapat mengurangi bahwa offset dari pembacaan langsung sensor percepatan untuk mendapatkan percepatan linier yang sebenarnya. Sistem koordinat sensor adalah sama dengan yang digunakan oleh sensor percepatan, seperti satuan ukuran (m/det²).

Rotation Vector Sensor

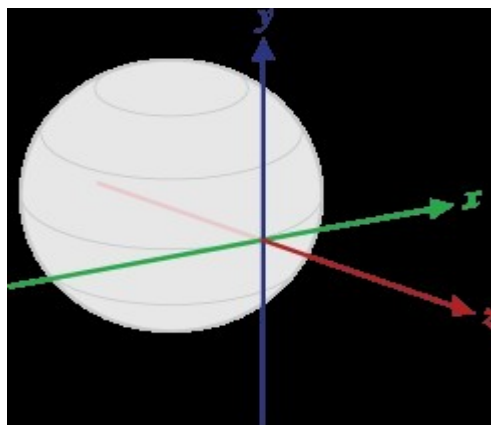
```
private SensorManager mSensorManager;
private Sensor mSensor;
...
mSensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
mSensor =
mSensorManager.getDefaultSensor(Sensor.TYPE_ROTATION_VECTOR);
```

Vektor rotasi merupakan orientasi perangkat sebagai kombinasi dari sudut dan sumbu, di mana perangkat telah diputar dengan sudut θ sekitar sebuah sumbu (x, y, atau z). Kode berikut menunjukkan cara untuk mendapatkan contoh dari sensor rotasi standar vektor:

The three elements of the rotation vector are expressed as follows:

$$\begin{aligned} &x \cdot \sin(\theta/2) \\ &y \cdot \sin(\theta/2) \\ &z \cdot \sin(\theta/2) \end{aligned}$$

Dimana besarnya dari vektor rotasi sama dengan dosa ($\theta / 2$), dan arah dari vektor rotasi sama dengan arah sumbu rotasi.





Tiga elemen dari vektor rotasi sama dengan tiga komponen terakhir dari angka empat unit ($\cos(\theta/2)$, $x \cdot \sin(\theta/2)$, $y \cdot \sin(\theta/2)$, $z \cdot \sin(\theta/2)$). Elemen dari vektor rotasi unitless. X, y, dan sumbu z didefinisikan dengan cara yang sama sebagai sensor percepatan. Referensi sistem koordinat didefinisikan sebagai basis ortonormal langsung (lihat gambar 1). Sistem koordinat memiliki karakteristik sebagai berikut: X didefinisikan sebagai produk vektor $Y \times Z$. Ini adalah tangensial ke tanah di lokasi perangkat dan menunjukkan sekitar Timur. Y adalah tangensial ke tanah di lokasi perangkat dan mengarah ke Kutub Utara geomagnetik. Z menunjuk ke langit dan tegak lurus terhadap bidang tanah. SDK Android memberikan contoh aplikasi yang menunjukkan bagaimana menggunakan sensor rotasi vektor. Aplikasi sampel terletak di API Demo kode (OS – RotationVectorDemo).

```

/*
 * Copyright (C) 2007 The Android Open Source Project
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
package com.example.android.apis.os;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;
import java.nio.FloatBuffer;
import javax.microedition.khronos.egl.EGLConfig;
import javax.microedition.khronos.opengles.GL10;
import android.app.Activity;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.opengl.GLSurfaceView;
import android.os.Bundle;
/**
 * Wrapper activity demonstrating the use of the new
 * {@link SensorEvent#values rotation vector sensor}
 * ({@link Sensor#TYPE_ROTATION_VECTOR TYPE_ROTATION_VECTOR}).
 */

```



```
* @see Sensor
* @see SensorEvent
* @see SensorManager
*
*/

public class RotationVectorDemo extends Activity {
    private GLSurfaceView mGLSurfaceView;
    private SensorManager mSensorManager;
    private MyRenderer mRenderer;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // Get an instance of the SensorManager
        mSensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
        // Create our Preview view and set it as the content of our
        // Activity
        mRenderer = new MyRenderer();
        mGLSurfaceView = new GLSurfaceView(this);
        mGLSurfaceView.setRenderer(mRenderer);
        setContentView(mGLSurfaceView);
    }
    @Override
    protected void onResume() {
        // Ideally a game should implement onResume() and onPause()
        // to take appropriate action when the activity looses focus
        super.onResume();
        mRenderer.start();
        mGLSurfaceView.onResume();
    }
    @Override
    protected void onPause() {
        // Ideally a game should implement onResume() and onPause()
        // to take appropriate action when the activity looses focus
        super.onPause();
        mRenderer.stop();
        mGLSurfaceView.onPause();
    }
    class MyRenderer implements GLSurfaceView.Renderer, SensorEventListener {
        private Cube mCube;
        private Sensor mRotationVectorSensor;
        private final float[] mRotationMatrix = new float[16];
        public MyRenderer() {

            // find the rotation-vector sensor
            mRotationVectorSensor = mSensorManager.getDefaultSensor(
                Sensor.TYPE_ROTATION_VECTOR);
            mCube = new Cube();
            // initialize the rotation matrix to identity
        }
    }
}
```



```
mRotationMatrix[ 0] = 1;
mRotationMatrix[ 4] = 1;
mRotationMatrix[ 8] = 1;
mRotationMatrix[12] = 1;
}
public void start() {
// enable our sensor when the activity is resumed, ask for
// 10 ms updates.
mSensorManager.registerListener(this, mRotationVectorSensor, 10000);
}
public void stop() {
// make sure to turn our sensor off when the activity is paused
mSensorManager.unregisterListener(this);
}
public void onSensorChanged(SensorEvent event) {
// we received a sensor event. it is a good practice to check
// that we received the proper event
if (event.sensor.getType() == Sensor.TYPE_ROTATION_VECTOR) {
// convert the rotation-vector to a 4x4 matrix. the matrix
// is interpreted by Open GL as the inverse of the
// rotation-vector, which is what we want.
SensorManager.getRotationMatrixFromVector(
mRotationMatrix , event.values);
}
}
public void onDrawFrame(GL10 gl) {
// clear screen
gl.glClear(GL10.GL_COLOR_BUFFER_BIT);
// set-up modelview matrix
gl.glMatrixMode(GL10.GL_MODELVIEW);
gl.glLoadIdentity();
gl.glTranslatef(0, 0, -3.0f);
gl.glMultMatrixf(mRotationMatrix, 0);

// draw our object
gl.glEnableClientState(GL10.GL_VERTEX_ARRAY);
gl.glEnableClientState(GL10.GL_COLOR_ARRAY);
mCube.draw(gl);
}
public void onSurfaceChanged(GL10 gl, int width, int height) {
// set view-port
gl.glViewport(0, 0, width, height);
// set projection matrix
float ratio = (float) width / height;
gl.glMatrixMode(GL10.GL_PROJECTION);
gl.glLoadIdentity();
gl.glFrustumf(-ratio, ratio, -1, 1, 1, 10);
}
public void onSurfaceCreated(GL10 gl, EGLConfig config) {
```




```
// dither is enabled by default, we don't need it
gl.glDisable(GL10.GL_DITHER);
// clear screen in white
gl.glClearColor(1,1,1,1);
}
class Cube {
// initialize our cube
private FloatBuffer mVertexBuffer;
private FloatBuffer mColorBuffer;
private ByteBuffer mIndexBuffer;
public Cube() {
final float vertices[] = {
-1, -1, -1,
1, -1, -1,
1, 1, -1,
-1, 1, -1,
-1, -1, 1,
1, -1, 1,
1, 1, 1, -1, 1, 1,
};
final float colors[] = {
0, 0, 0, 1, 1, 0, 1
1, 1, 0, 1, 0, 1, 1
0, 0, 1, 1, 1, 0, 1
1, 1, 1, 1, 0, 1, 1
};
final byte indices[] = {
0, 4, 5, 0, 5, 1,
1, 5, 6, 1, 6, 2,
2, 6, 7, 2, 7, 3,
3, 7, 4, 3, 4, 0,
4, 7, 6, 4, 6, 5,
3, 0, 1, 3, 1, 2
};
ByteBuffer vbb = ByteBuffer.allocateDirect(vertices.length*4);
vbb.order(ByteOrder.nativeOrder());
mVertexBuffer = vbb.asFloatBuffer();
mVertexBuffer.put(vertices);
mVertexBuffer.position(0);
ByteBuffer cbb = ByteBuffer.allocateDirect(colors.length*4);
cbb.order(ByteOrder.nativeOrder());
mColorBuffer = cbb.asFloatBuffer();
mColorBuffer.put(colors);
mColorBuffer.position(0);
mIndexBuffer = ByteBuffer.allocateDirect(indices.length);
mIndexBuffer.put(indices);
mIndexBuffer.position(0);
}
```



```

public void draw(GL10 gl) {
gl.glEnable(GL10.GL_CULL_FACE);
gl.glFrontFace(GL10.GL_CW);
gl.glShadeModel(GL10.GL_SMOOTH);
gl.glVertexPointer(3, GL10.GL_FLOAT, 0, mVertexBuffer);
gl.glColorPointer(4, GL10.GL_FLOAT, 0, mColorBuffer);
gl.glDrawElements(GL10.GL_TRIANGLES, 36, GL10.GL_UNSIGNED_BYTE,
mIndexBuffer);
}
}
public void onAccuracyChanged(Sensor sensor, int accuracy) {
}
}
}

```

Sensor Suara

Dalam sebuah teknologi sebelumnya saya sudah memaparkan sensor apa saja yang berada pada device android ,ternyata banyak dan banyak sekali , karena android merupakan smartphone dan akan menjadi superphone , sensor suara pada suatu teknologi di dapat contohnya pada teknologi pembuka pintu dengan menggunakan suara yang di kenalnya . Sedangkan pada teknologi stave job yang disebut dengan siri nya itu juga menggunakan sensor suara , kita akan membuat sensor suara itu pada android , langsung ke lembar kerja :

1. Create project dengan nama Speak
2. Buka file pada res – layout – main.xml untuk membuat tampilan :

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:background="#ffffff">

    <TextView android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Text To Speech"
        android:padding="15dip"
        android:textColor="#0587d9"
        android:textSize="26dip"
        android:gravity="center"
        android:textStyle="bold"/>

    <EditText android:id="@+id/txtText"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="Enter some text to speak"
        android:layout_marginTop="20dip"
        android:layout_margin="10dip"/>

    <Button android:id="@+id/btnSpeak"

```



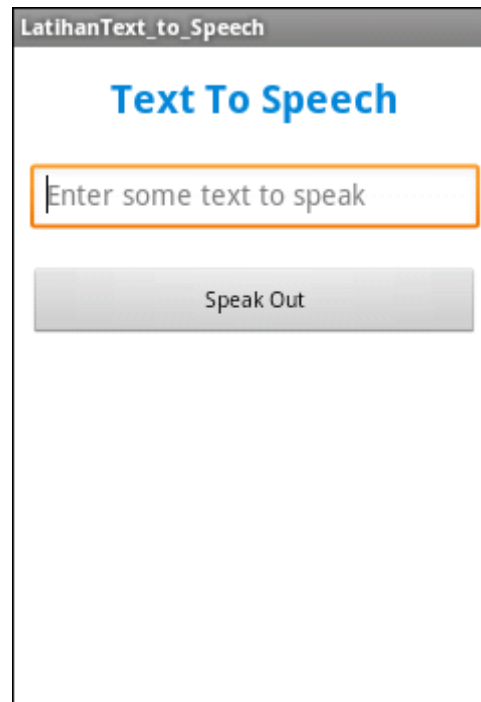
```

        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Speak Out"
        android:layout_margin="10dip"/>

```

```
</LinearLayout>
```

Dan hasil nya seperti ini ,karena gambar yang dimasukkan dapat menggunakan gambar lain :



3. Buka file pada src – package – SpeakActivity.java

```

package speect.wartec;

import java.util.Locale;

import android.app.Activity;

import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.speech.tts.TextToSpeech.OnInitListener;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class LatihanText_to_SpeechActivity extends Activity implements OnInitListener{
    /** Called when the activity is first created. */
    private TextToSpeech tts;
    private Button btnSpeak;

```



```

        private EditText txtText;
        @Override
        public void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.main);
            tts = new TextToSpeech(this, this);

            btnSpeak = (Button) findViewById(R.id.btnSpeak);

            txtText = (EditText) findViewById(R.id.txtText);

            // button on click event
            btnSpeak.setOnClickListener(new View.OnClickListener() {

                public void onClick(View arg0) {
                    speakOut();
                }

            });
        }

        @Override
        public void onDestroy() {
            // Don't forget to shutdown!
            if (tts != null) {
                tts.stop();
                tts.shutdown();
            }
            super.onDestroy();
        }

        public void onInit(int status) {
            // TODO Auto-generated method stub
            // TODO Auto-generated method stub

            if (status == TextToSpeech.SUCCESS) {

                int result = tts.setLanguage(Locale.US);

                // tts.setPitch(5); // set pitch level

                // tts.setSpeechRate(2); // set speech speed rate

                if (result == TextToSpeech.LANG_MISSING_DATA
                    || result ==
TextToSpeech.LANG_NOT_SUPPORTED) {
                    Log.e("TTS", "Language is not supported");
                } else {
                    btnSpeak.setEnabled(true);
                    speakOut();
                }
            }
        }
    }

```




```

    }

    } else {
        Log.e("TTS", "Initilization Failed");
    }

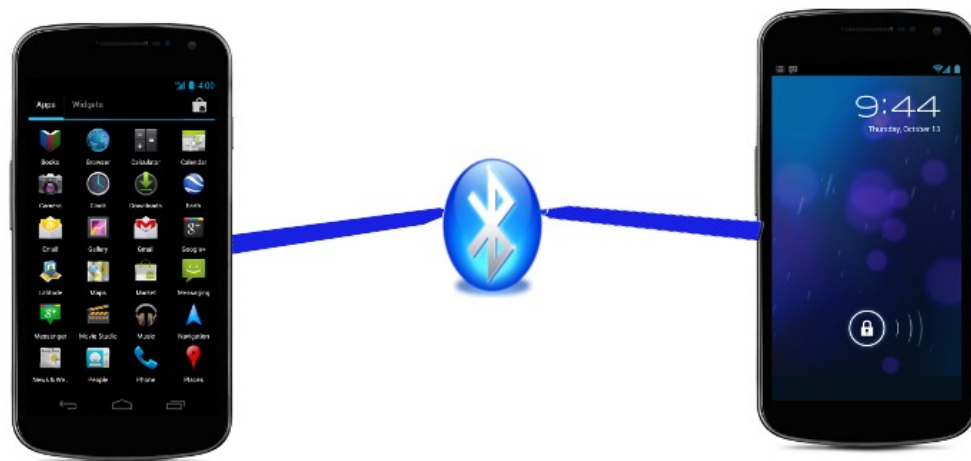
}

private void speakOut() {
    // TODO Auto-generated method stub
    String text = txtText.getText().toString();

    tts.speak(text, TextToSpeech.QUEUE_FLUSH, null);
}
}

```

Bluetooth Socket



Socket Bluetooth

Socket merupakan suatu device yang terhubung dengan device lainnya dengan sebuah alat bantu seperti Wireless , Bluetooth , atau pun yang lainnya , dalam sebuah kehidupan sehari hari socket sangat sering kita lihat , dari hal kita mengirim data dari handphone dengan handphone yang lainnya itu termasuk socket , kali ini kita akan membahas socket pada bluetooth , aplikasi kali ini dinamakan chat Bluetooth :

```

package Bluetoothchat.wartec;
import java.io.IOException;

```



```
import java.io.InputStream;
import java.io.OutputStream;
import java.util.UUID;

import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothServerSocket;
import android.bluetooth.BluetoothSocket;
import android.content.Context;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.util.Log;
public class BluetoothChatService {
    // Debugging
    private static final String TAG = "BluetoothChatService";
    private static final boolean D = true;

    // Name for the SDP record when creating server socket
    private static final String NAME = "BluetoothChat";

    // Unique UUID for this application
    private static final UUID MY_UUID = UUID.fromString("fa87c0d0-afac-11de-8a39-0800200c9a66");

    // Member fields
    private final BluetoothAdapter mAdapter;
    private final Handler mHandler;
    private AcceptThread mAcceptThread;
    private ConnectThread mConnectThread;
    private ConnectedThread mConnectedThread;
    private int mState;

    // Constants that indicate the current connection state
    public static final int STATE_NONE = 0; // we're doing nothing
    public static final int STATE_LISTEN = 1; // now listening for incoming connections
    public static final int STATE_CONNECTING = 2; // now initiating an outgoing connection
    public static final int STATE_CONNECTED = 3; // now connected to a remote device

    /**
     * Constructor. Prepares a new BluetoothChat session.
     * @param context The UI Activity Context
     * @param handler A Handler to send messages back to the UI Activity
     */
    public BluetoothChatService(Context context, Handler handler) {
        mAdapter = BluetoothAdapter.getDefaultAdapter();
        mState = STATE_NONE;
        mHandler = handler;
    }
}
```



```
/**
 * Set the current state of the chat connection
 * @param state An integer defining the current connection state
 */
private synchronized void setState(int state) {
    if (D) Log.d(TAG, "setState() " + mState + " -> " + state);
    mState = state;

    // Give the new state to the Handler so the UI Activity can update
    mHandler.obtainMessage(BluetoothChatActivity.MESSAGE_STATE_CHANGE, state,
-1).sendToTarget();
}

/**
 * Return the current connection state. */
public synchronized int getState() {
    return mState;
}

/**
 * Start the chat service. Specifically start AcceptThread to begin a
 * session in listening (server) mode. Called by the Activity onResume() */
public synchronized void start() {
    if (D) Log.d(TAG, "start");

    // Cancel any thread attempting to make a connection
    if (mConnectThread != null) {mConnectThread.cancel(); mConnectThread = null;}

    // Cancel any thread currently running a connection
    if (mConnectedThread != null) {mConnectedThread.cancel(); mConnectedThread = null;}

    // Start the thread to listen on a BluetoothServerSocket
    if (mAcceptThread == null) {
        mAcceptThread = new AcceptThread();
        mAcceptThread.start();
    }
    setState(STATE_LISTEN);
}

/**
 * Start the ConnectThread to initiate a connection to a remote device.
 * @param device The BluetoothDevice to connect
 */
public synchronized void connect(BluetoothDevice device) {
    if (D) Log.d(TAG, "connect to: " + device);

    // Cancel any thread attempting to make a connection
    if (mState == STATE_CONNECTING) {
```



```

        if (mConnectThread != null) {mConnectThread.cancel(); mConnectThread = null;}
    }

    // Cancel any thread currently running a connection
    if (mConnectedThread != null) {mConnectedThread.cancel(); mConnectedThread = null;}

    // Start the thread to connect with the given device
    mConnectThread = new ConnectThread(device);
    mConnectThread.start();
    setState(STATE_CONNECTING);
}

/**
 * Start the ConnectedThread to begin managing a Bluetooth connection
 * @param socket The BluetoothSocket on which the connection was made
 * @param device The BluetoothDevice that has been connected
 */
public synchronized void connected(BluetoothSocket socket, BluetoothDevice device) {
    if (D) Log.d(TAG, "connected");

    // Cancel the thread that completed the connection
    if (mConnectThread != null) {mConnectThread.cancel(); mConnectThread = null;}

    // Cancel any thread currently running a connection
    if (mConnectedThread != null) {mConnectedThread.cancel(); mConnectedThread = null;}

    // Cancel the accept thread because we only want to connect to one device
    if (mAcceptThread != null) {mAcceptThread.cancel(); mAcceptThread = null;}

    // Start the thread to manage the connection and perform transmissions
    mConnectedThread = new ConnectedThread(socket);
    mConnectedThread.start();

    // Send the name of the connected device back to the UI Activity
    Message msg =
mHandler.obtainMessage(BluetoothchatActivity.MESSAGE_DEVICE_NAME);
    Bundle bundle = new Bundle();
    bundle.putString(BluetoothchatActivity.DEVICE_NAME, device.getName());
    msg.setData(bundle);
    mHandler.sendMessage(msg);

    setState(STATE_CONNECTED);
}

/**
 * Stop all threads
 */
public synchronized void stop() {
    if (D) Log.d(TAG, "stop");

```




```
        if (mConnectThread != null) {mConnectThread.cancel(); mConnectThread = null;}
        if (mConnectedThread != null) {mConnectedThread.cancel(); mConnectedThread = null;}
        if (mAcceptThread != null) {mAcceptThread.cancel(); mAcceptThread = null;}
        setState(STATE_NONE);
    }

    /**
     * Write to the ConnectedThread in an unsynchronized manner
     * @param out The bytes to write
     * @see ConnectedThread#write(byte[])
     */
    public void write(byte[] out) {
        // Create temporary object
        ConnectedThread r;
        // Synchronize a copy of the ConnectedThread
        synchronized (this) {
            if (mState != STATE_CONNECTED) return;
            r = mConnectedThread;
        }
        // Perform the write unsynchronized
        r.write(out);
    }

    /**
     * Indicate that the connection attempt failed and notify the UI Activity.
     */
    private void connectionFailed() {
        setState(STATE_LISTEN);

        // Send a failure message back to the Activity
        Message msg = mHandler.obtainMessage(BluetoothchatActivity.MESSAGE_TOAST);
        Bundle bundle = new Bundle();
        bundle.putString(BluetoothchatActivity.TOAST, "Unable to connect device");
        msg.setData(bundle);
        mHandler.sendMessage(msg);
    }

    /**
     * Indicate that the connection was lost and notify the UI Activity.
     */
    private void connectionLost() {
        setState(STATE_LISTEN);

        // Send a failure message back to the Activity
        Message msg = mHandler.obtainMessage(BluetoothchatActivity.MESSAGE_TOAST);
        Bundle bundle = new Bundle();
        bundle.putString(BluetoothchatActivity.TOAST, "Device connection was lost");
        msg.setData(bundle);
        mHandler.sendMessage(msg);
    }
}
```



```
}

/**
 * This thread runs while listening for incoming connections. It behaves
 * like a server-side client. It runs until a connection is accepted
 * (or until cancelled).
 */
private class AcceptThread extends Thread {
    // The local server socket
    private final BluetoothServerSocket mmServerSocket;

    public AcceptThread() {
        BluetoothServerSocket tmp = null;

        // Create a new listening server socket
        try {
            tmp = mAdapter.listenUsingRfcommWithServiceRecord(NAME, MY_UUID);
        } catch (IOException e) {
            Log.e(TAG, "listen() failed", e);
        }
        mmServerSocket = tmp;
    }

    public void run() {
        if (D) Log.d(TAG, "BEGIN mAcceptThread" + this);
        setName("AcceptThread");
        BluetoothSocket socket = null;

        // Listen to the server socket if we're not connected
        while (mState != STATE_CONNECTED) {
            try {
                // This is a blocking call and will only return on a
                // successful connection or an exception
                socket = mmServerSocket.accept();
            } catch (IOException e) {
                Log.e(TAG, "accept() failed", e);
                break;
            }

            // If a connection was accepted
            if (socket != null) {
                synchronized (BluetoothChatService.this) {
                    switch (mState) {
                        case STATE_LISTEN:
                        case STATE_CONNECTING:
                            // Situation normal. Start the connected thread.
                            connected(socket, socket.getRemoteDevice());
                            break;
                        case STATE_NONE:

```



```

        case STATE_CONNECTED:
            // Either not ready or already connected. Terminate new socket.
            try {
                socket.close();
            } catch (IOException e) {
                Log.e(TAG, "Could not close unwanted socket", e);
            }
            break;
        }
    }
}

if (D) Log.i(TAG, "END mAcceptThread");
}

public void cancel() {
    if (D) Log.d(TAG, "cancel " + this);
    try {
        mmServerSocket.close();
    } catch (IOException e) {
        Log.e(TAG, "close() of server failed", e);
    }
}

}

/**
 * This thread runs while attempting to make an outgoing connection
 * with a device. It runs straight through; the connection either
 * succeeds or fails.
 */
private class ConnectThread extends Thread {
    private final BluetoothSocket mmSocket;
    private final BluetoothDevice mmDevice;

    public ConnectThread(BluetoothDevice device) {
        mmDevice = device;
        BluetoothSocket tmp = null;

        // Get a BluetoothSocket for a connection with the
        // given BluetoothDevice
        try {
            tmp = device.createRfcommSocketToServiceRecord(MY_UUID);
        } catch (IOException e) {
            Log.e(TAG, "create() failed", e);
        }
        mmSocket = tmp;
    }
}

```



```
public void run() {
    Log.i(TAG, "BEGIN mConnectThread");
    setName("ConnectThread");

    // Always cancel discovery because it will slow down a connection
    mAdapter.cancelDiscovery();

    // Make a connection to the BluetoothSocket
    try {
        // This is a blocking call and will only return on a
        // successful connection or an exception
        mmSocket.connect();
    } catch (IOException e) {
        connectionFailed();
        // Close the socket
        try {
            mmSocket.close();
        } catch (IOException e2) {
            Log.e(TAG, "unable to close() socket during connection failure", e2);
        }
        // Start the service over to restart listening mode
        BluetoothChatService.this.start();
        return;
    }

    // Reset the ConnectThread because we're done
    synchronized (BluetoothChatService.this) {
        mConnectThread = null;
    }

    // Start the connected thread
    connected(mmSocket, mmDevice);
}

public void cancel() {
    try {
        mmSocket.close();
    } catch (IOException e) {
        Log.e(TAG, "close() of connect socket failed", e);
    }
}

/**
 * This thread runs during a connection with a remote device.
 * It handles all incoming and outgoing transmissions.
 */
private class ConnectedThread extends Thread {
    private final BluetoothSocket mmSocket;
```




```
private final InputStream mmInStream;
private final OutputStream mmOutStream;

public ConnectedThread(BluetoothSocket socket) {
    Log.d(TAG, "create ConnectedThread");
    mmSocket = socket;
    InputStream tmpIn = null;
    OutputStream tmpOut = null;

    // Get the BluetoothSocket input and output streams
    try {
        tmpIn = socket.getInputStream();
        tmpOut = socket.getOutputStream();
    } catch (IOException e) {
        Log.e(TAG, "temp sockets not created", e);
    }

    mmInStream = tmpIn;
    mmOutStream = tmpOut;
}

public void run() {
    Log.i(TAG, "BEGIN mConnectedThread");
    byte[] buffer = new byte[1024];
    int bytes;

    // Keep listening to the InputStream while connected
    while (true) {
        try {
            // Read from the InputStream
            bytes = mmInStream.read(buffer);

            // Send the obtained bytes to the UI Activity
            mHandler.obtainMessage(BluetoothchatActivity.MESSAGE_READ, bytes, -1, buffer)
                .sendToTarget();
        } catch (IOException e) {
            Log.e(TAG, "disconnected", e);
            connectionLost();
            break;
        }
    }
}

/**
 * Write to the connected OutStream.
 * @param buffer The bytes to write
 */
public void write(byte[] buffer) {
    try {
```



```

        mmOutputStream.write(buffer);

        // Share the sent message back to the UI Activity
        mHandler.obtainMessage(BluetoothchatActivity.MESSAGE_WRITE, -1, -1, buffer)
            .sendToTarget();
    } catch (IOException e) {
        Log.e(TAG, "Exception during write", e);
    }
}

public void cancel() {
    try {
        mmSocket.close();
    } catch (IOException e) {
        Log.e(TAG, "close() of connect socket failed", e);
    }
}
}
}

```

Aplikasi SMS Sain

Aplikasi SMS sain ini merupakan aplikasi yang terbanyak rating di Samsung Apps yang di creator oleh saya sendiri , aplikasi ini bertujuan untuk memudahkan pengguna dalam ber sms dengan rekannya , fitur utama dari aplikasi ini yaitu rumus matematika dasar , saya berterima kasih kepada team wartech yang telah membuat desain dari tampilan awal yaitu reki susanto TF 10 , semoga aplikasi ini bermanfaat :

1. Create Project dengan nama SmsSain
2. Buka File pada res – layout – main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:focusable="true"
    android:visibility="visible"
    android:background="#159f07"
    android:alwaysDrawnWithCache="true">

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="48dp"
        android:gravity="center_horizontal|bottom"
        android:background="#b9c6b8"
        android:orientation="horizontal" >
    <Button
        android:id="@+id/btnkontak"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

```



```
        android:background="@drawable/ct"
    />
    <EditText
        android:id="@+id/txtPhoneNo"
        android:layout_width="405dp"
        android:layout_height="wrap_content"
        android:layout_weight="0.07"
        android:cursorVisible="true"
        android:numeric="integer"
        android:visibility="visible" />

</LinearLayout>

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="133dp"
    android:gravity="center_horizontal|bottom"
    android:orientation="horizontal" >

    <EditText
        android:id="@+id/txtMessage"
        android:layout_width="wrap_content"
        android:layout_height="121dp"
        android:layout_weight="0.45"
        android:cursorVisible="true"
        android:visibility="visible" />

    <Button
        android:id="@+id/btnSendSMS"
        android:layout_width="92dp"
        android:layout_height="72dp"
        android:background="@drawable/ms" />

</LinearLayout>

<SlidingDrawer
    android:id="@+id/slidingDrawer1"
    android:layout_width="fill_parent"
    android:layout_height="match_parent"
    android:content="@+id/content"
    android:handle="@+id/handle"
    android:longClickable="true" >

    <Button
        android:id="@+id/handle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/ff" />

    <LinearLayout
        android:id="@+id/content"
        android:layout_width="fill_parent"
        android:layout_height="118dp"
        android:orientation="horizontal"
        android:keepScreenOn="true" >
        <TabHost
            android:id="@+id/tabhost"
            android:layout_width="match_parent"
```



```

        android:layout_height="match_parent" >

#Tabhost
<TabHost android:id="@+id/tabhost"
android:layout_width="fill_parent"
android:layout_height="fill_parent">

<TabWidget android:id="@android:id/tabs"
android:layout_width="fill_parent"
android:layout_height="wrap_content"/>

<FrameLayout android:id="@android:id/tabcontent"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:paddingTop="62px">

#tab1
<LinearLayout
                android:id="@+id/tab1"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:gravity="left"
                android:orientation="horizontal">

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

    <Button
        android:id="@+id/vektor"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/persegi"
        android:text="L"/>

    <Button
        android:id="@+id/segitiga"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/segitiga"
        android:text="L"/>

    <Button
        android:id="@+id/lingkaran"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/lingkaran"
        android:text="L"/>

    <Button
        android:id="@+id/trapesium"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/trapesium"
        android:text="L"/>

    <Button
        android:id="@+id/jajar"
        android:layout_width="wrap_content"

```




```

        android:layout_height="wrap_content"
        android:background="@drawable/jajar"
        android:text="L"/>

</LinearLayout>
#tab2

<LinearLayout
    android:id="@+id/tab2"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="left"
    android:orientation="horizontal">

    <Button
        android:id="@+id/kubus"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/kubus"
        android:text="L"/>

    <Button
        android:id="@+id/balok"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/balok"
        android:text="L"/>

    <Button
        android:id="@+id/prisma"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/prisma"
        android:text="L"/>

    <Button
        android:id="@+id/kerucut"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/kerucut"
        android:text="L"/>

    <Button
        android:id="@+id/Tabung"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/tabung"
        android:text="L"/>

    <Button
        android:id="@+id/bola"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/bola"
        android:text="L"/>

    <Button
        android:id="@+id/limas"

```



```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/limas"
        android:text="L"/>
    </LinearLayout>

```

#tab3

```

<LinearLayout
    android:id="@+id/tab3"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="left"
    android:orientation="horizontal">

    <Button
        android:id="@+id/vkubus"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/kubus"
        android:text="L"/>

        <Button
            android:id="@+id/vbalok"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:background="@drawable/balok"
            android:text="L"/>

            <Button
                android:id="@+id/vprisma"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:background="@drawable/prisma"
                android:text="L"/>

                <Button
                    android:id="@+id/vkerucut"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:background="@drawable/kerucut"

                    android:text="L"/>
        <Button
            android:id="@+id/vTabung"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:background="@drawable/tabung"
            android:text="L"/>

            <Button
                android:id="@+id/vbola"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:background="@drawable/bola"
                android:text="L"/>

```



```

        <Button
            android:id="@+id/vlimas"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:background="@drawable/limas"
            android:text="L"/>
    </LinearLayout>

</FrameLayout>
</TabHost>

</TabHost>

    </LinearLayout>
</SlidingDrawer>
</LinearLayout>

```



2. Create Android xml dan beri nama inbox

```

<?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" android:id="@+id/MainLayout"
    android:background="@drawable/background">

    <Button android:layout_height="wrap_content"
        android:layout_width="match_parent" android:id="@+id/create"
        android:layout_margin="2dip"
        android:text="Create SMS Science" />

    <ListView android:id="@+id/SMSList"
        android:layout_height="wrap_content"
        android:layout_width="match_parent"
        android:layout_margin="2dip" />

</LinearLayout>

```



3. Create android xml dan beri nama webview

```
<?xml version="1.0" encoding="utf-8"?>
<WebView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/webwartech"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

</WebView>
```



4. Create Class pada src – package dengan nama Smsreceiver

```
package smssain.wartech.com;

import android.content.BroadcastReceiver;
import android.content.ContentResolver;
import android.content.ContentValues;
import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.telephony.SmsMessage;
import android.widget.Toast;
```




```
public class SmsReceiver extends BroadcastReceiver {
    public static final String SMS_EXTRA_NAME = "pdus";
    public static final String SMS_URI = "content://sms";

    public static final String ADDRESS = "address";
    public static final String PERSON = "person";
    public static final String DATE = "date";
    public static final String READ = "read";
    public static final String STATUS = "status";
    public static final String TYPE = "type";
    public static final String BODY = "body";
    public static final String SEEN = "seen";

    public static final int MESSAGE_TYPE_INBOX = 1;
    public static final int MESSAGE_TYPE_SENT = 2;

    public static final int MESSAGE_IS_NOT_READ = 0;
    public static final int MESSAGE_IS_READ = 1;

    public static final int MESSAGE_IS_NOT_SEEN = 0;
    public static final int MESSAGE_IS_SEEN = 1;

    // Change the password here or give a user possibility to change it
    public static final byte[] PASSWORD = new byte[]{ 0x20, 0x32, 0x34, 0x47, (byte) 0x84, 0x33,
0x58 };

    public void onReceive( Context context, Intent intent )
    {
        // Get SMS map from Intent
        Bundle extras = intent.getExtras();

        String messages = "";

        if ( extras != null )
        {
            // Get received SMS array
            Object[] smsExtra = (Object[]) extras.get( SMS_EXTRA_NAME );

            // Get ContentResolver object for pushing encrypted SMS to incoming folder
            ContentResolver contentResolver = context.getContentResolver();

            for ( int i = 0; i < smsExtra.length; ++i )
            {
                SmsMessage sms = SmsMessage.createFromPdu((byte[])smsExtra[i]);

                String body = sms.getMessageBody().toString();
                String address = sms.getOriginatingAddress();
            }
        }
    }
}
```



```

        messages += "SMS from " + address + " :\n";
        messages += body + "\n";

        // Here you can add any your code to work with incoming SMS
        // I added encrypting of all received SMS

        putSmsToDatabase( contentResolver, sms );
    }

    // Display SMS message
    Toast.makeText( context, messages, Toast.LENGTH_SHORT ).show();
}

// WARNING!!!
// If you uncomment next line then received SMS will not be put to incoming.
// Be careful!
// this.abortBroadcast();
}

private void putSmsToDatabase( ContentResolver contentResolver, SmsMessage sms )
{
    // Create SMS row
    ContentValues values = new ContentValues();
    values.put( ADDRESS, sms.getOriginatingAddress() );
    values.put( DATE, sms.getTimestampMillis() );
    values.put( READ, MESSAGE_IS_NOT_READ );
    values.put( STATUS, sms.getStatus() );
    values.put( TYPE, MESSAGE_TYPE_INBOX );
    values.put( SEEN, MESSAGE_IS_NOT_SEEN );
    try
    {
        String encryptedPassword = StringCrytor.encrypt( new String(PASSWORD),
sms.getMessageBody().toString() );
        values.put( BODY, encryptedPassword );
    }
    catch ( Exception e )
    {
        e.printStackTrace();
    }

    // Push row into the SMS table
    contentResolver.insert( Uri.parse( SMS_URI ), values );
}
}

```

5. Create Class pada src – package dan bernama StringCripton

```

package smssain.wartech.com;
import java.security.SecureRandom;

```



```

import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import android.util.Base64;
public class StringCryptor {
    private static final String CIPHER_ALGORITHM = "AES";
    private static final String RANDOM_GENERATOR_ALGORITHM = "SHA1PRNG";
    private static final int RANDOM_KEY_SIZE = 128;

    // Encrypts string and encode in Base64
    public static String encrypt( String password, String data ) throws Exception
    {
        byte[] secretKey = generateKey( password.getBytes() );
        byte[] clear = data.getBytes();

        SecretKeySpec secretKeySpec = new SecretKeySpec( secretKey,
CIPHER_ALGORITHM );
        Cipher cipher = Cipher.getInstance( CIPHER_ALGORITHM );
        cipher.init( Cipher.ENCRYPT_MODE, secretKeySpec );

        byte[] encrypted = cipher.doFinal( clear );
        String encryptedString = Base64.encodeToString( encrypted, Base64.DEFAULT );

        return encryptedString;
    }

    // Decrypts string encoded in Base64
    public static String decrypt( String password, String encryptedData ) throws Exception
    {
        byte[] secretKey = generateKey( password.getBytes() );

        SecretKeySpec secretKeySpec = new SecretKeySpec( secretKey,
CIPHER_ALGORITHM );
        Cipher cipher = Cipher.getInstance( CIPHER_ALGORITHM );
        cipher.init( Cipher.DECRYPT_MODE, secretKeySpec );

        byte[] encrypted = Base64.decode( encryptedData, Base64.DEFAULT );
        byte[] decrypted = cipher.doFinal( encrypted );

        return new String( decrypted );
    }

    public static byte[] generateKey( byte[] seed ) throws Exception
    {
        KeyGenerator keyGenerator =
KeyGenerator.getInstance( CIPHER_ALGORITHM );
        SecureRandom secureRandom =
SecureRandom.getInstance( RANDOM_GENERATOR_ALGORITHM );

```



```
        secureRandom.setSeed( seed );
        keyGenerator.init( RANDOM_KEY_SIZE, secureRandom );
        SecretKey secretKey = keyGenerator.generateKey();
        return secretKey.getEncoded();
    }
}
```

6. Create Class dengan nama main

```
package smssain.wartech.com;

import java.util.ArrayList;

import org.apache.http.cookie.SM;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.ContentResolver;
import android.content.Intent;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.telephony.SmsMessage;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.Toast;
import android.widget.AdapterView.OnItemClickListener;

public class main extends Activity implements OnClickListener, OnItemClickListener {

    Button create;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);

        setTheme( android.R.style.Theme_Light );
        setContentView(R.layout.inbox);

        create = (Button)findViewById(R.id.create);
    }
}
```




```

        create.setOnClickListener(this);

        ContentResolver contentResolver = getContentResolver();
        Cursor cursor = contentResolver.query( Uri.parse( "content://sms/inbox" ),
null, null, null, null);

        int indexBody = cursor.getColumnIndex( SmsReceiver.BODY );
        int indexAddr = cursor.getColumnIndex( SmsReceiver.ADDRESS);
        int indexName = cursor.getColumnIndex(SmsReceiver.PERSON);

        if ( indexBody < 0 || !cursor.moveToFirst() ) return;

        smsList.clear();

        do
        {
            String str = cursor.getString(indexName) +
cursor.getString( indexAddr ) + "\n" + cursor.getString( indexBody );
            smsList.add( str);
        }
        while( cursor.moveToNext() );

        ListView smsListView = (ListView) findViewById( R.id.SMSList );
        smsListView.setAdapter( new ArrayAdapter<String>( this,
android.R.layout.simple_list_item_1, smsList) );
        smsListView.setOnItemClickListener( this );

        /**
         * You can also register your intent filter here.
         * And here is example how to do this.
         */
        IntentFilter filter = new
IntentFilter( "android.provider.Telephony.SMS_RECEIVED" );
        * filter.setPriority( IntentFilter.SYSTEM_HIGH_PRIORITY );
        * registerReceiver( new SmsReceiver(), filter );
        **/

    }

    ArrayList<String> smsList = new ArrayList<String>();
    public void onItemClick(AdapterView<?> parent, View view, int pos, long id ) {
        // TODO Auto-generated method stub
        try

```



```

        {
            String[] splitted = smsList.get( pos ).split("\n");
            String sender = splitted[0];
            String encryptedData = "";
            for ( int i = 1; i < splitted.length; ++i )
            {
                encryptedData += splitted[i];
            }
            String data = sender + "\n" + StringCrytor.decrypt( new
String(SmsReceiver.PASSWORD), encryptedData );
            Toast.makeText( this, data, Toast.LENGTH_SHORT ).show();
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
    public void onClick(View v) {
        // TODO Auto-generated method stub
        Intent i = new Intent(main.this , SMSainActivity.class);

        startActivity(i);
    }
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.main,menu);
        return true;
    }

    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle item selection
        switch (item.getItemId()) {
            case R.id.about:
                about();
                return true;
            case R.id.update:
                update();
                return true;

            default:
                return super.onOptionsItemSelected(item);
        }
    }

    private void about() {
        // TODO Auto-generated method stub
        AlertDialog.Builder build = new AlertDialog.Builder(main.this);
        build.setTitle("About")
        .setIcon(R.drawable.fficon)

```



```
.setMessage("Wartech-id is The Warung Teknologi \n" +
            "" +
            "wartech-id.com\n\n" +
            "copyright(c)2012")
.setPositiveButton("ok",null)
.show();
}
private void update() {
    // TODO Auto-generated method stub
    Intent web = new Intent(main.this,WebViewActivity.class);
    startActivity(web);
}
public void rumus(){

}

}
```

7. Create Class dengan nama WebViewActivity

```
package smssain.wartech.com;

import java.sql.Date;

import android.app.Activity;
import android.os.Bundle;
import android.webkit.WebSettings;
import android.webkit.WebView;

public class WebViewActivity extends Activity{
    private WebView webview;

    final String mimeType = "text/html";
    final String encoding = "UTF-8";
    String htmldata = "<html><body>x³</body></html>";

    public void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.webview);

        webview = (WebView)findViewById(R.id.webwartech);
        webview.getSettings().setJavaScriptEnabled(true);
        webview.loadUrl("http://www.wartech-id.com");

    }
```



```
}
```

8. Buka File SMSsainActivity.java

```
package smssain.wartech.com;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.Dialog;
import android.app.PendingIntent;
import android.app.AlertDialog.Builder;
import android.content.BroadcastReceiver;
import android.content.ContentResolver;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.IntentFilter;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.telephony.SmsMessage;
import android.telephony.gsm.SmsManager;
import android.text.style.UpdateAppearance;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.webkit.WebSettings.TextSize;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.SlidingDrawer;
import android.widget.SlidingDrawer.OnDrawerCloseListener;
import android.widget.SlidingDrawer.OnDrawerOpenListener;
import android.widget.TabHost;
import android.widget.Toast;
import android.os.Bundle;
import android.provider.Contacts.People;
import android.provider.ContactsContract;

public class SMSsainActivity extends Activity implements
    OnDrawerOpenListener, OnDrawerCloseListener {
    /** Called when the activity is first created. */
    Button btnSendSMS,contact;
    EditText txtPhoneNo;
    EditText txtMessage;
```




```

TabHost tabs;
Button vektor,segitiga,lingkaran,trapesium,jajar;
Button kubus,balok,bola,limas,prisma,kerucut,tabung;
Button vkubus,vbalok,vbola,vlimas,vprisma,vkerucut,vtabung;
SlidingDrawer slidingDrawer;
@Override
public void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    txtPhoneNo = (EditText)findViewById(R.id.txtPhoneNo);
    contact=(Button)findViewById(R.id.btnkontak);

    contact.setOnClickListener(new View.OnClickListener() {

        public void onClick(View arg0) {
            // TODO Auto-generated method stub
            Intent intent = new Intent(Intent.ACTION_PICK,
ContactsContract.Contacts.CONTENT_URI);
            startActivityForResult(intent, 1);
        }
    });

    tabs = (TabHost)findViewById(R.id.tabhost);
    tabs.setup();

    TabHost.TabSpec spec = tabs.newTabSpec("Tag1");

    //Kita akan mendeklarasikan tab 1

    spec.setContent(R.id.tab1);
    spec.setIndicator("Two Dimensions");
    tabs.addTab(spec);

    //tab 2

    spec = tabs.newTabSpec("Tag 2");
    spec.setContent(R.id.tab2);
    spec.setIndicator("Three Dimensions");
    tabs.addTab(spec);
    tabs.setCurrentTab(0);

    //tab 3
    spec = tabs.newTabSpec("Tag 3");
    spec.setContent(R.id.tab3);
    spec.setIndicator("Volume");
    tabs.addTab(spec);

```



```

tabs.setCurrentTab(0);

btnSendSMS = (Button) findViewById(R.id.btnSendSMS);
btnSendSMS.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        String phoneNo = txtPhoneNo.getText().toString();
        String message = txtMessage.getText().toString();
        if (phoneNo.length() > 0 && message.length() > 0)
            sendSMS(phoneNo, message);
        else
            Toast.makeText(getApplicationContext(),
                "Please enter both phone number and
message.",
                    Toast.LENGTH_SHORT).show();
    }
});
txtPhoneNo = (EditText) findViewById(R.id.txtPhoneNo);
txtMessage = (EditText) findViewById(R.id.txtMessage);
txtMessage.setText("");
contact = (Button) findViewById(R.id.btnkontak);
vektor = (Button) findViewById(R.id.vektor);
segitiga = (Button) findViewById(R.id.segitiga);
lingkaran = (Button) findViewById(R.id.lingkaran);
trapesium = (Button) findViewById(R.id.trapesium);
jajar = (Button) findViewById(R.id.jajar);

jajar.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        // TODO Auto-generated method stub
        txtMessage.setText("Broad parallelogram = base(b) x height(h) ");
        btnSendSMS.setVisibility(ListView.VISIBLE);
        txtMessage.setVisibility(ListView.VISIBLE);
    }
});

trapesium.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        // TODO Auto-generated method stub
        txtMessage.setText("Broad Trapezoidal = total side parallel x
width(w) / 2");

        btnSendSMS.setVisibility(ListView.VISIBLE);
        txtMessage.setVisibility(ListView.VISIBLE);
    }
});

```



```
lingkaran.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        // TODO Auto-generated method stub
txtMessage.setText("Broad circle =  $\phi(\pi) \times r^2$ ");
btnSendSMS.setVisibility(ListView.VISIBLE);
txtMessage.setVisibility(ListView.VISIBLE);
    }
});
slidingDrawer = (SlidingDrawer) this.findViewById(R.id.slidingDrawer1);

//Listen for open event

slidingDrawer.setOnDrawerOpenListener(this);

// Listen for close event

slidingDrawer.setOnDrawerCloseListener(this);

vektor.setOnClickListener(new View.OnClickListener() {

    public void onClick(View arg0) {
        // TODO Auto-generated method stub

        txtMessage.setText("Broad square = lengthy(l) x width(w)");
        btnSendSMS.setVisibility(ListView.VISIBLE);
        txtMessage.setVisibility(ListView.VISIBLE);
    }
});
segitiga.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        // TODO Auto-generated method stub
        txtMessage.setText("Broad triangle = base(b) x height(h) / 2");
        btnSendSMS.setVisibility(ListView.VISIBLE);
        txtMessage.setVisibility(ListView.VISIBLE);
    }
});

kubus = (Button)findViewById(R.id.kubus);
kubus.setOnClickListener(new View.OnClickListener() {
```



```

        public void onClick(View v) {
            // TODO Auto-generated method stub
            txtMessage.setText("Wide cube = 6 x side(s) x
side(s)");

            btnSendSMS.setVisibility(ListView.VISIBLE);
            txtMessage.setVisibility(ListView.VISIBLE);
        }
    });
    balok = (Button)findViewById(R.id.balok);
    balok.setOnClickListener(new View.OnClickListener() {

        public void onClick(View v) {
            // TODO Auto-generated method stub
            txtMessage.setText("Wide beam = 2x length x width +
2 x length x height + 2 x width x height");
            btnSendSMS.setVisibility(ListView.VISIBLE);
            txtMessage.setVisibility(ListView.VISIBLE);
        }
    });
    bola = (Button)findViewById(R.id.bola);
    bola.setOnClickListener(new View.OnClickListener() {

        public void onClick(View v) {
            // TODO Auto-generated method stub
            txtMessage.setText("Wide ball = 4 x phi( $\pi$ ) x r2");
            btnSendSMS.setVisibility(ListView.VISIBLE);
            txtMessage.setVisibility(ListView.VISIBLE);
        }
    });
    limas = (Button)findViewById(R.id.limas);
    limas.setOnClickListener(new View.OnClickListener() {

        public void onClick(View v) {
            // TODO Auto-generated method stub
            txtMessage.setText("Limas wide = wide base + wide
side");
            btnSendSMS.setVisibility(ListView.VISIBLE);
            txtMessage.setVisibility(ListView.VISIBLE);
        }
    });
    prisma = (Button)findViewById(R.id.prisma);
    prisma.setOnClickListener(new View.OnClickListener() {

        public void onClick(View v) {
            // TODO Auto-generated method stub
            txtMessage.setText("Wide Limas = wide base + wide
side");
            btnSendSMS.setVisibility(ListView.VISIBLE);

```




```

        txtMessage.setVisibility(ListView.VISIBLE);
    }
});
kerucut = (Button)findViewById(R.id.kerucut);
kerucut.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        // TODO Auto-generated method stub
        txtMessage.setText("Wide cone =  $\phi(\pi) \times r^2 +$ 
hypotenuse");

        btnSendSMS.setVisibility(ListView.VISIBLE);
        txtMessage.setVisibility(ListView.VISIBLE);
    }
});
tabung = (Button)findViewById(R.id.Tabung);
tabung.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        // TODO Auto-generated method stub
        txtMessage.setText("Wide Tubes = 2 x wide base ( $\pi \times$ 
 $r^2$ ) + wide blanket ( $\pi \times (d) \times h$ ");

        btnSendSMS.setVisibility(ListView.VISIBLE);
        txtMessage.setVisibility(ListView.VISIBLE);
    }
});

vkubus = (Button)findViewById(R.id.vkubus);
vkubus.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        // TODO Auto-generated method stub
        txtMessage.setText("Volume Cube =  $s^3$ ");
        btnSendSMS.setVisibility(ListView.VISIBLE);
        txtMessage.setVisibility(ListView.VISIBLE);
    }
});

vbalok = (Button)findViewById(R.id.vbalok);
vbalok.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        // TODO Auto-generated method stub
        txtMessage.setText("Volume Beam = length(l) x
width(w) x height(h)");

        btnSendSMS.setVisibility(ListView.VISIBLE);
        txtMessage.setVisibility(ListView.VISIBLE);
    }
});
vbola = (Button)findViewById(R.id.vbola);

```



```

vbola.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        // TODO Auto-generated method stub
        txtMessage.setText("Volume Ball =  $\frac{4}{3} \times \pi \times$ 
jari(r)3");

        btnSendSMS.setVisibility(ListView.VISIBLE);
        txtMessage.setVisibility(ListView.VISIBLE);
    }
});
vlimas = (Button)findViewById(R.id.vlimas);
vlimas.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        // TODO Auto-generated method stub
        txtMessage.setText("Volume pyramid =  $\frac{1}{3} \times$ 
base(b)area x height(h)");

        btnSendSMS.setVisibility(ListView.VISIBLE);
        txtMessage.setVisibility(ListView.VISIBLE);
    }
});
vprisma = (Button)findViewById(R.id.vprisma);
vprisma.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        // TODO Auto-generated method stub
        txtMessage.setText("Volume Prism = area of base(b) x
height(h)");

        btnSendSMS.setVisibility(ListView.VISIBLE);
        txtMessage.setVisibility(ListView.VISIBLE);
    }
});
vkerucut = (Button)findViewById(R.id.vkerucut);
vkerucut.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        // TODO Auto-generated method stub
        txtMessage.setText("Volume Cone = area of base(b) x
height(h) / 3");

        btnSendSMS.setVisibility(ListView.VISIBLE);
        txtMessage.setVisibility(ListView.VISIBLE);
    }
});
vtabung = (Button)findViewById(R.id.vTabung);
vtabung.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        // TODO Auto-generated method stub
        txtMessage.setText("Volume Tube = area of base(b) x

```



```

height(h));

                                btnSendSMS.setVisibility(ListView.VISIBLE);
                                txtMessage.setVisibility(ListView.VISIBLE);
                                }
                                });
                                }

private void sendSMS(String phoneNumber, String message) {
    String SENT = "SMS SENT";
    String DELIVERED = "SMS DELIVERED";

    PendingIntent sentPI = PendingIntent.getBroadcast(this, 0, new Intent(
        SENT), 0);

    PendingIntent deliveredPI = PendingIntent.getBroadcast(this, 0,
        new Intent(DELIVERED), 0);

    // ---when the SMS has been sent---
    registerReceiver(new BroadcastReceiver() {
        @Override
        public void onReceive(Context arg0, Intent arg1) {
            switch (getResultCode()) {
                case Activity.RESULT_OK:
                    Toast.makeText(getBaseContext(), "SMS sent",
                        Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_GENERIC_FAILURE:
                    Toast.makeText(getBaseContext(), "Generic failure",
                        Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_NO_SERVICE:
                    Toast.makeText(getBaseContext(), "No service",
                        Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_NULL_PDU:
                    Toast.makeText(getBaseContext(), "Null PDU",
                        Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_RADIO_OFF:
                    Toast.makeText(getBaseContext(), "Radio off",
                        Toast.LENGTH_SHORT).show();
                    break;
            }
        }
    }, new IntentFilter(SENT));

    // ---when the SMS has been delivered---
    registerReceiver(new BroadcastReceiver() {

```



```

        @Override
        public void onReceive(Context arg0, Intent arg1) {
            switch (getResultCode()) {
                case Activity.RESULT_OK:
                    Toast.makeText(getBaseContext(), "SMS delivered",
                                   Toast.LENGTH_SHORT).show();
                    break;
                case Activity.RESULT_CANCELED:
                    Toast.makeText(getBaseContext(), "SMS not delivered",
                                   Toast.LENGTH_SHORT).show();
                    break;
            }
        }
    }, new IntentFilter(DELIVERED));

    SmsManager sms = SmsManager.getDefault();
    sms.sendTextMessage(phoneNumber, null, message, sentPI, deliveredPI);
}

public void onDrawerClosed() {
    // TODO Auto-generated method stub

    contact.setVisibility(Button.VISIBLE);
    btnSendSMS.setVisibility(Button.VISIBLE);
    txtMessage.setVisibility(EditText.VISIBLE);
    txtPhoneNo.setVisibility(EditText.VISIBLE);

}

public void onDrawerOpened() {
    // TODO Auto-generated method stub

    contact.setVisibility(Button.GONE);
    btnSendSMS.setVisibility(Button.GONE);
    txtMessage.setVisibility(EditText.GONE);
    txtPhoneNo.setVisibility(EditText.GONE);

}

@Override
public void onActivityResult(int reqCode, int resultCode, Intent data) {

    ContentResolver cr = getContentResolver();
    Cursor cur = cr.query(ContactsContract.Contacts.CONTENT_URI,
        null, null, null, null);
    if (cur.getCount() > 0) {
        while (cur.moveToNext()) {
            String id = cur.getString(cur.getColumnIndex(ContactsContract.Contacts._ID));

```


10. Buka File AndroidManifest.xml



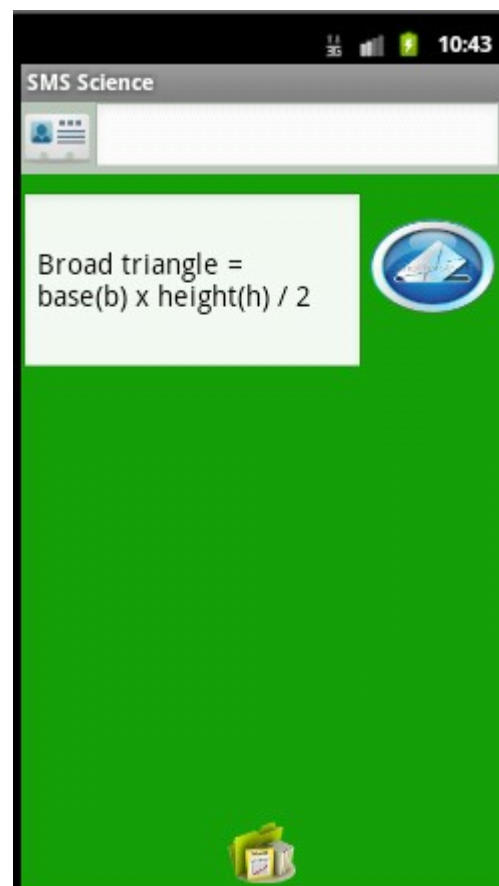
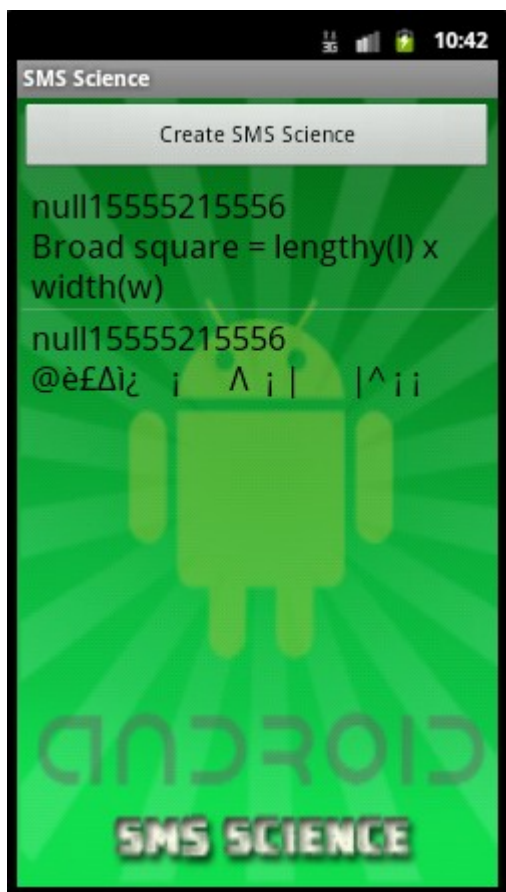
```

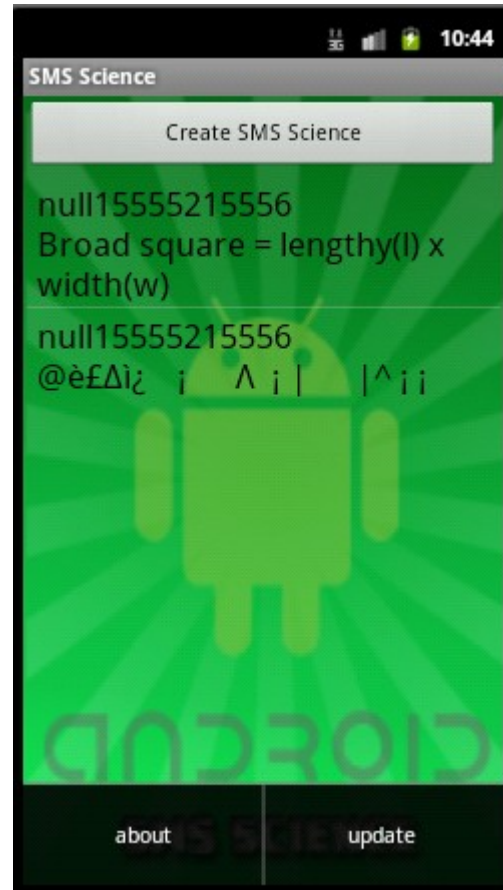
        android:theme="@android:style/Theme.NoTitleBar"/>
    </application>

    <uses-permission android:name="android.permission.WRITE_SMS" />
    <uses-permission android:name="android.permission.READ_SMS" />
    <uses-permission android:name="android.permission.RECEIVE_SMS" />
    <uses-permission android:name="android.permission.SEND_SMS">
    </uses-permission>
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.WRITE_CONTACTS"/>
<uses-permission android:name="android.permission.INTERNET"/>
</manifest>

```

Run Aplikasi :





Aplikasi Bean Shell Mini

Bean Shell Merupakan scrip yang berbahasan shell , disini kita hanya menggunakan class yang telah di sediakan oleh beanshell <http://beanshell.org/developer.html> , untuk lebih jelas nya beanshell dapat.

1. Download BeanShell di sini <http://beanshell.org/developer.html>
2. Masukkan file bsh.jar ke dalam jdk anda , jdk/lib
3. Create Project dengan Nama BeanShellMini
4. Tambahkan package bns.jar pada build
5. Buka File main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

<TextView android:text="(BeanShell Mini)"
    android:id="@+id/rm"
    android:layout_width="wrap_content"
    android:textColor="#aeac00"
    android:textSize="20px"
    android:textStyle="bold"
    android:layout_height="wrap_content"
```



```

        android:layout_gravity="center"></TextView>

<ImageView
    android:id="@+id/imageView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/aa"
        android:layout_gravity="center"></ImageView>

#Tabhost
<TabHost android:id="@+id/tabhost"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

<TabWidget android:id="@android:id/tabs"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"/>

<FrameLayout android:id="@android:id/tabcontent"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:paddingTop="62px">

#tab1
<LinearLayout
    android:id="@+id/tab1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">

    <Button
        android:id="@+id/eval"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Go!"
    />
    <EditText
        android:id="@+id/script"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:singleLine="false"
        android:gravity="top"
    />

</LinearLayout>

#tab2
<TextView android:id="@+id/tab2"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_gravity="center"
    android:text="What is Bean Shell ?\n\n
    BeanShell is a small, free, embeddable Java source interpreter with object
    scripting language features, written in Java. BeanShell dynamically executes
    standard Java syntax and extends it with common scripting conveniences such as
    loose types, commands, and method closures like those in Perl and
    JavaScript.\n\n
    http://www.http://beanshell.org/"></TextView>

#tab3
<TextView
    android:id="@+id/tab3"
        android:layout_width="fill_parent"

```

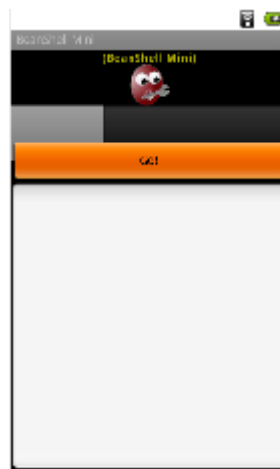



```

        android:layout_height="fill_parent"
        android:text="Thank God Bean Shell Applications is made with
such a result
, Made with passion and full to ceriaan with friends who have helped.
Made in a small house that most of my early dreams god willing, one by one
can be achieved with the best - good. amen. \n\n
Copyright@ aerdy@gmail.com"/>

</FrameLayout>
</TabHost>
</LinearLayout>

```



5. Buka File BeanShellMiniActivity pada src/package

```

package aerdy.wartec.com;

import bsh.Interpreter;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.AlertDialog.Builder;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TabHost;

public class BeanShellMiniActivity extends Activity {
    /** Called when the activity is first created. */
    private Interpreter i=new Interpreter();

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        TabHost tabs=(TabHost)findViewById(R.id.tabhost);

```



```

tabs.setup();

Button btn=(Button)findViewById(R.id.eval);
final EditText script=(EditText)findViewById(R.id.script);
btn.setOnClickListener(new View.OnClickListener() {

    public void onClick(View view) {
        String src=script.getText().toString();
        try {
            i.set("context",BeanShellMiniActivity.this);
            i.eval(src);
            AlertDialog.Builder builder= new AlertDialog.Builder(BeanShellMiniActivity.this);

            builder
                .setTitle("Compile")
                .setMessage((String) i.eval(src))
                .setPositiveButton("OK",null);
        }
        catch (bsh.EvalError e) {
            AlertDialog.Builder builder=
            new AlertDialog.Builder(BeanShellMiniActivity.this);
            builder
                .setTitle("Error!!!")
                .setMessage(e.toString())
                .setPositiveButton("OK", null)
                .show();
        }
    }
});

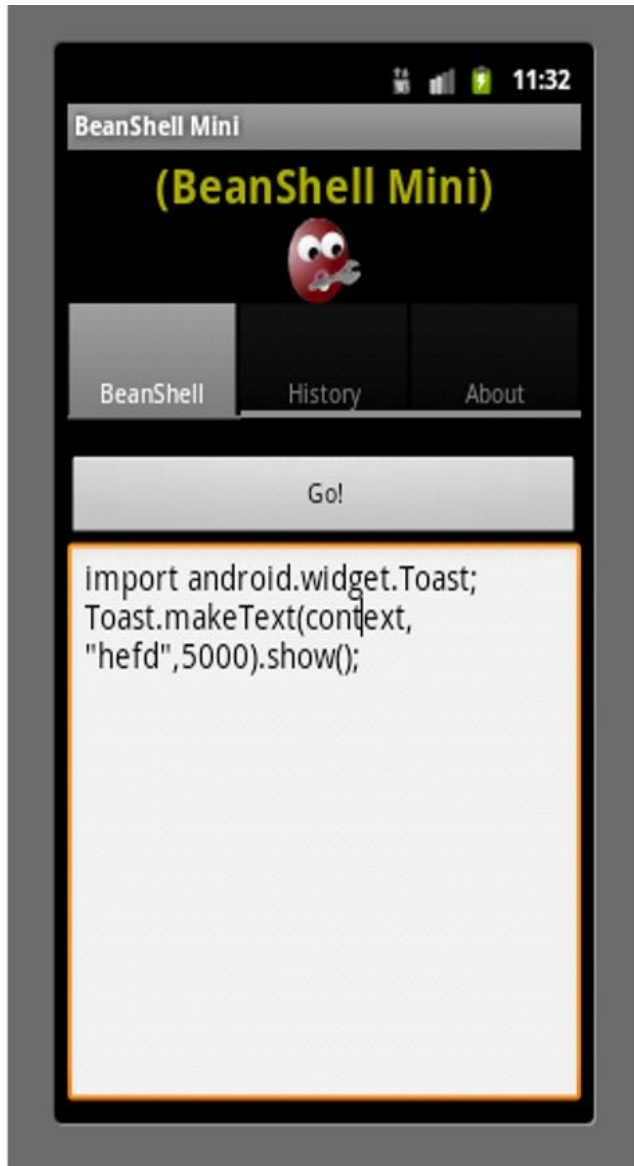
TabHost.TabSpec spec=tabs.newTabSpec("tag1");
spec.setContent(R.id.tab1);
spec.setIndicator("BeanShell");
tabs.addTab(spec);

spec=tabs.newTabSpec("tag2");
spec.setContent(R.id.tab2);
spec.setIndicator("History");
tabs.addTab(spec);

spec=tabs.newTabSpec("tag2");
spec.setContent(R.id.tab3);
spec.setIndicator("About");
tabs.addTab(spec);
tabs.setCurrentTab(0);
}
}

```

Run Project dan hasil nya seperti di bawah ini :



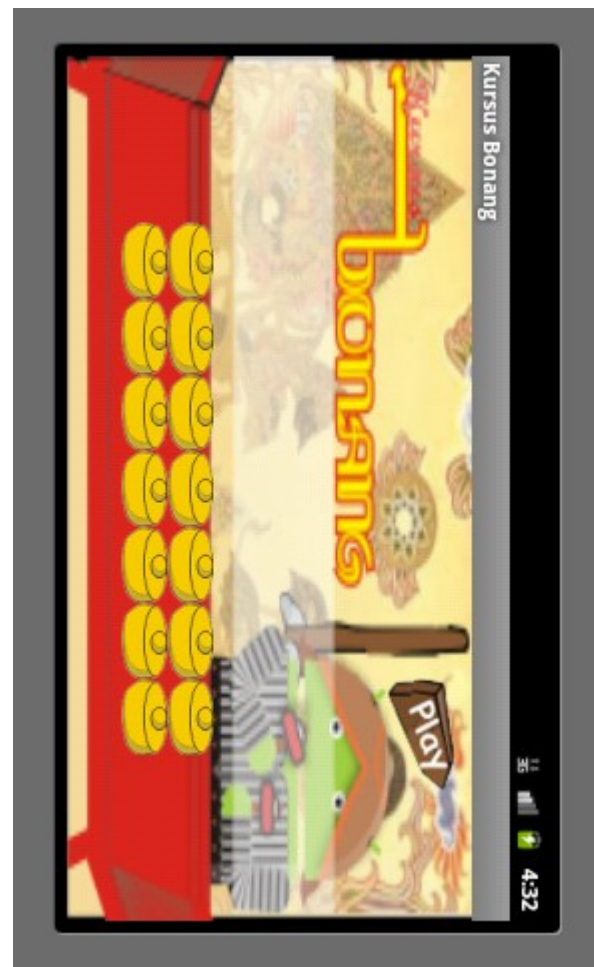
ANDROBOT

Androbot merupakan android robot , ini sebuah game yang saya develop dengan mengcopy left dari jetboy yang di develop oleh team android develop , di bawah ini merupakan gambar develop nya :



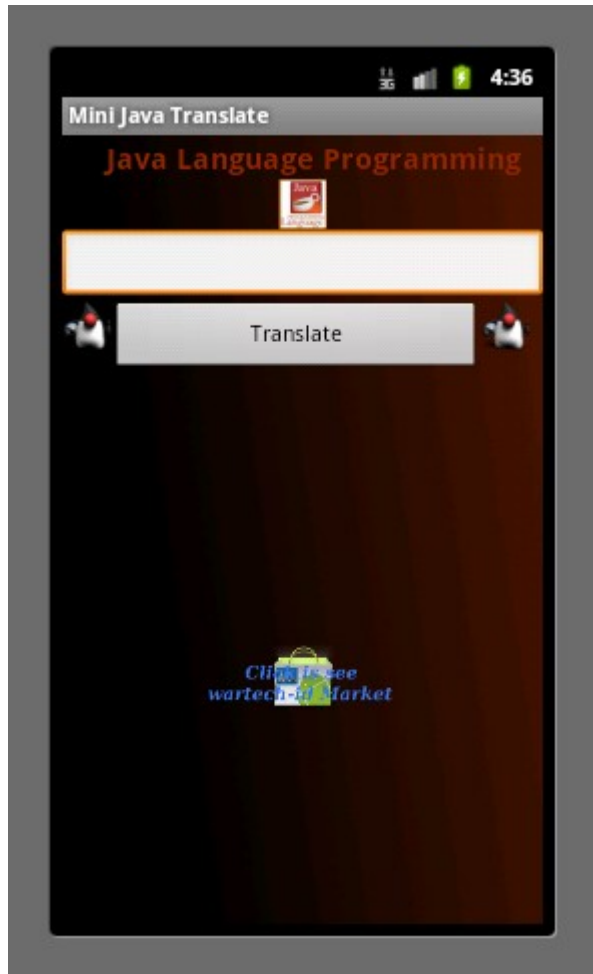
Kursus Bonang

Kursus Bonang Merupakan aplikasi yang di buat bersama team , yaitu Saya Anna arthdi putra ,Reki susanti , Ariska hidayar, aplikasi ini alhamdulillah membawa final pada saat kompetisi di salatiga , aplikasi ini bertema membawa kita untuk belajar bonang lanjutan yang merupakan asli alat musik jawa . Untuk tampilan nya seperti di bawah ini :



Mini Java Translate

Mini Java Translate Merupakan kamus programming java yang saya buat , aplikasi ini bertujuan agar dimana saja kita bisa memahami programming java dengan menggunakan ponsel android , aplikasi ini sudah mencapai versi ke 2.4 , setiap kalinya mengalami perkembangan hingga insya allah kita akan bisa koding di mobile dengan programing java .amin





Catatan :