

Pemrograman Android Widget – Part 1

[Pemrograman Android Widget 01][Level: Pemula]

Andi Taru Nugroho Nur Wismono S.Kom.,M.Cs.

andi.taru@gmail.com

Lisensi Dokumen:

Copyright ©2012 JavaClopedia.com

Seluruh dokumen di JavaClopedia.com dapat digunakan dan disebarikan secara bebas untuk tujuan non-komersial dan harus menyertakan penulis serta sumber asli dokumen yaitu JavaClopedia.com.

Penulisan ulang tidak diperkenankan tanpa seijin JavaClopedia.com

Persiapan

Sebelum mengikuti pembelajaran yang akan dibahas, dibutuhkan beberapa perangkat lunak yang harus terinstall dengan baik di komputer pembaca. Minimal sudah terinstall dan tersetting dengan benar beberapa perangkat lunak di bawah ini:

- Tutorial sebelumnya. Pemrograman Android Dasar 01 – Instalasi

Pendahuluan

Widget adalah komponen dalam bahasa pemrograman Android yang dapat digunakan untuk menampilkan user interface dan menerima interaksi terhadap user. Pada tutorial kali ini, penulis akan membahas beberapa hal dasar yang berkaitan dengan widget seperti:

- XML Layout
- Layout Width
- Layout Height
- Gravity

XML Layout

Android XML Layout secara default terletak pada bagian res/layout/main.xml. saat kita membuat project baru kurang lebih isinya akan seperti berikut ini:

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

Arti dari kode program di atas adalah layout Widget diatur dalam berkas XML dan menggunakan jenis layout LinearLayout. Sedangkan Widget Textview berada di dalam LinearLayout karena setiap widget harus berada di dalam layout tertentu.

LAYOUT WIDTH dan HEIGHT

Layout width dan height di dalam Android, terdapat 2 macam yaitu:

- Fill Parent
- Wrap Content

Jika **layout_width** di set **fill_parent** maka lebar dari komponen Widget tersebut akan memenuhi layar monitor Device Mobile. Sedangkan jika **layout_height** yang diset **fill_parent** maka tinggi dari Widget tersebut akan memenuhi layar monitor device mobile.

Berbeda dengan **layout_width** jika diset **wrap_content**, maka lebar dari komponen widget tersebut akan menyesuaikan content yang ada di dalamnya. Misalnya teks pada button. Lebar button akan menyesuaikan lebar teks dalam button tersebut. Hal ini berlaku juga untuk layout_height.

GRAVITY

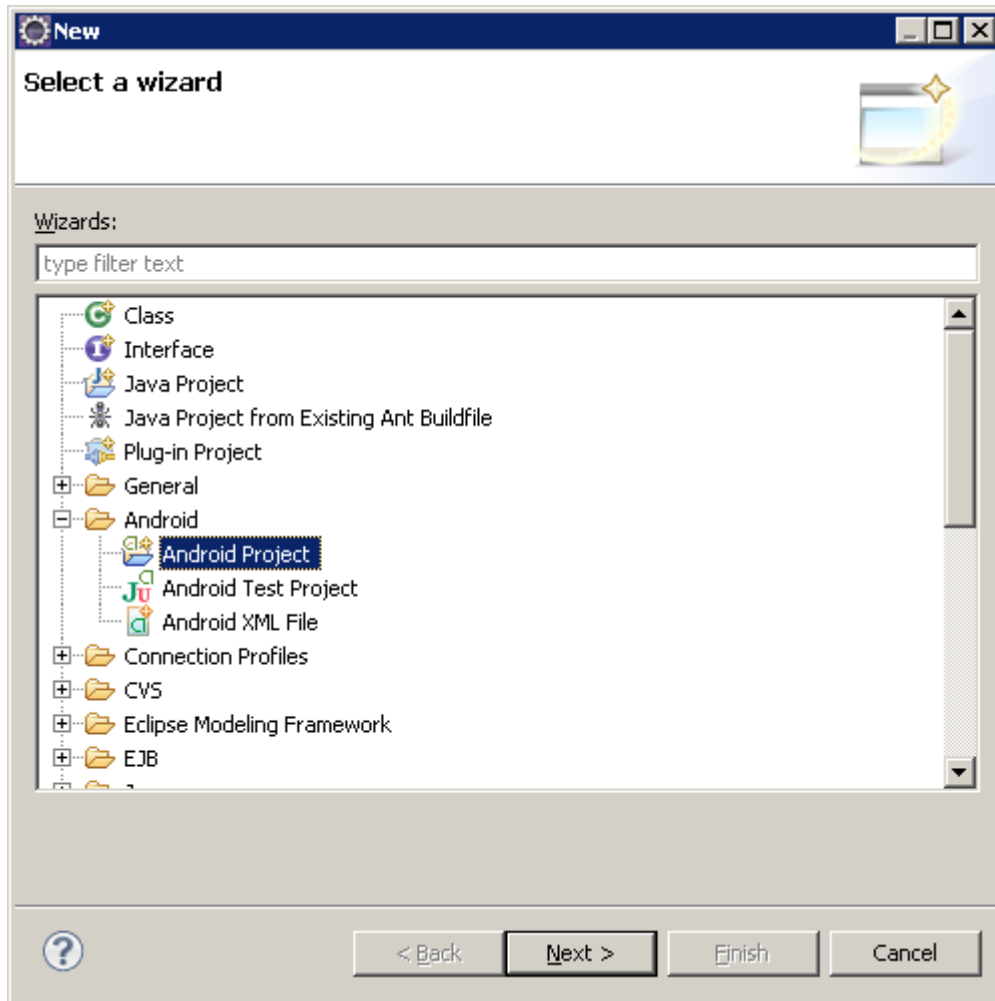
Selanjutnya adalah gravity, gravity merupakan perataan teks pada komponen widget. Misalnya sebuah button di set **gravity = right**, maka posisi teks pada button tersebut akan menjadi rata kanan. Secara default, **gravity** bernilai **left**.

Untuk mencoba XML Layout, Layout Width dan Height serta Gravity kita dapat langsung melanjutkan pada tutorial praktek kita.

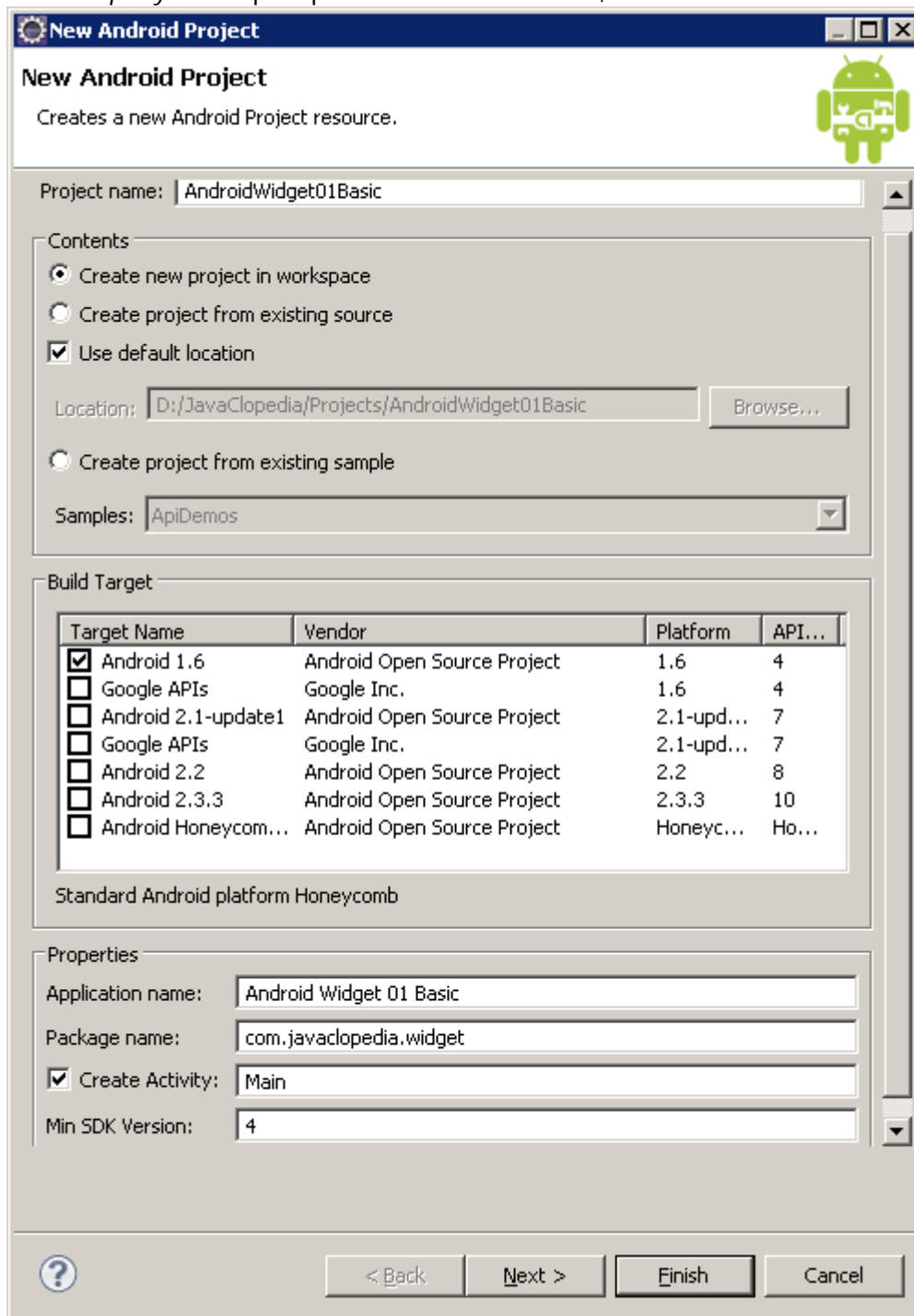
ANDROID IN ACTION!

Saatnya kita mencobanya di Android ☺ Pembaca ikuti langkah-langkah berikut ini yak:

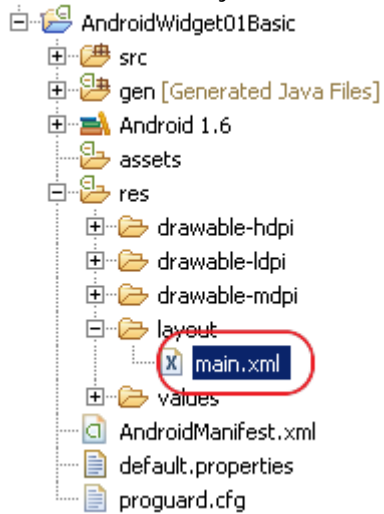
1. Buka IDE Eclipse yang sudah didownload dan tersetting Android.
2. File > New > Other atau dengan shortcut (CTRL + N)
Pilih Android > Android Project > Next



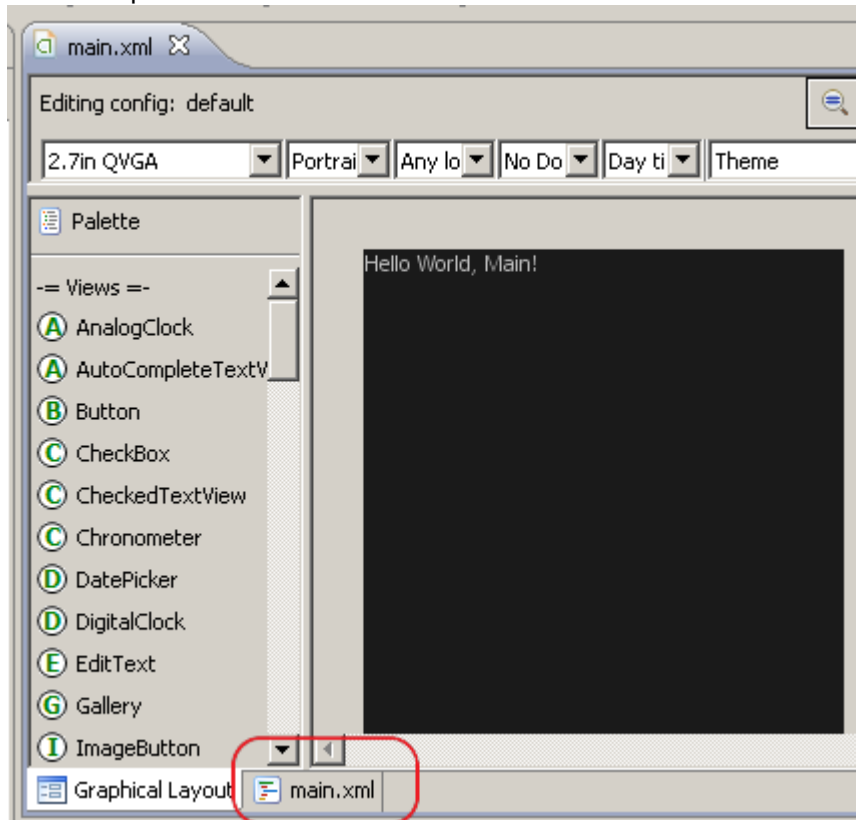
3. Isikan *Property* isian seperti pada Gambar di bawah ini, kemudian tekan *Finish*.



4. Buka src > res > layout > main.xml



5. Kemudian pilih main.xml

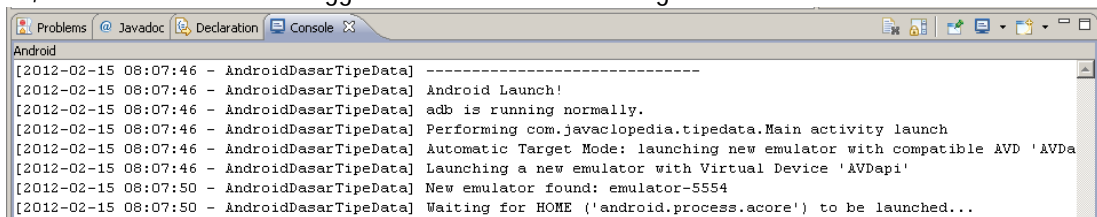


6. Kode Program akan menjadi berikut ini:

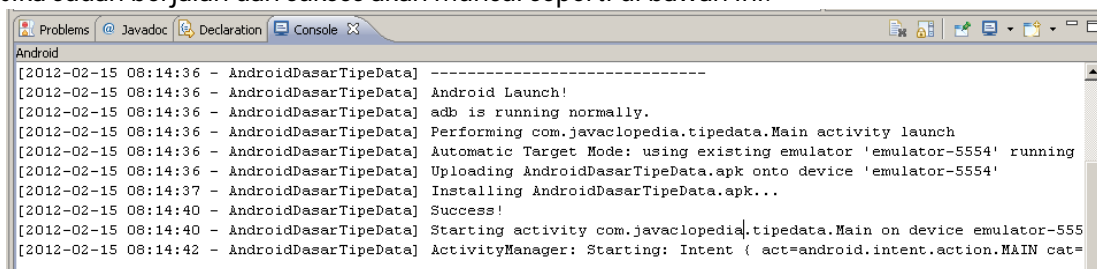
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
```

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

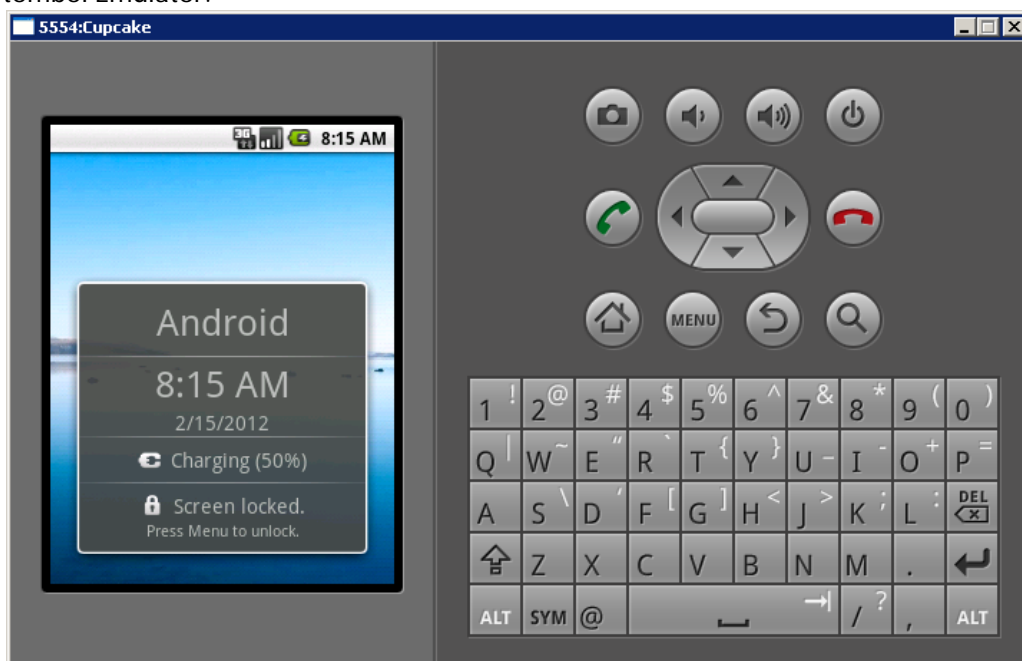
7. Jalankan aplikasi Android dengan cara klik kanan *AndroidWidget01Basic* > *Run As* > *Android Application*. Akan muncul Emulator, memang biasanya agak lama. Jadi harus bersabar ☺ Ada baiknya emulator dijalankan terlebih dahulu (caranya ada pada tutorial sebelum ini).
8. Sambil menunggu bisa tekan *Window* > *Show View* > *Console*. Jika tampilannya masih seperti ini, maka kita harus menunggu Emulator selesai Loading Home.



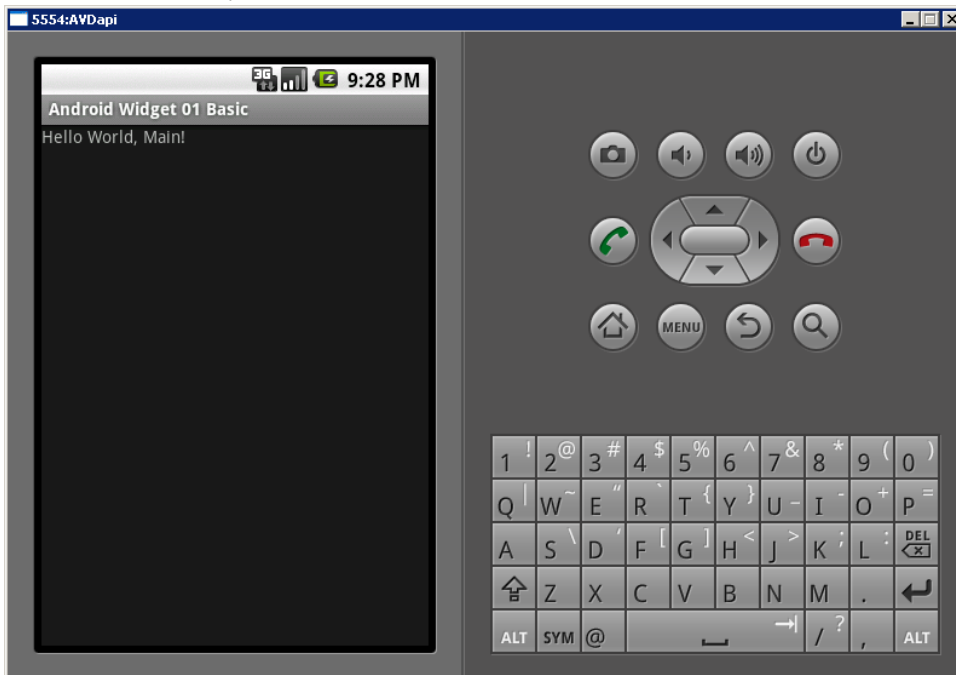
9. Jika sudah berjalan dan sukses akan muncul seperti di bawah ini:



10. Kemudian kita cek Emulatornya, jika muncul seperti Gambar di bawah, tekan MENU di tombol Emulator.



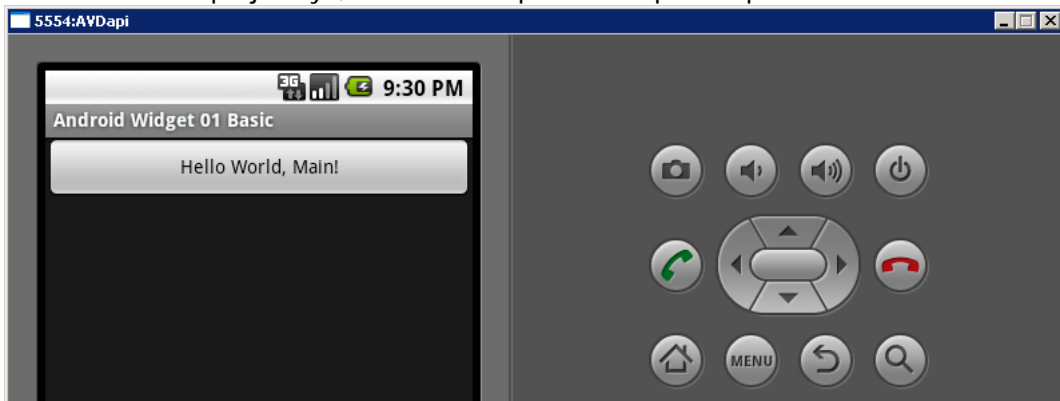
11. Jika tidak ada error, maka akan muncul berikut ini di Emulator:



12. Selanjutnya kita akan mencoba membuat button pada tampilan. Ubah kode main.xml menjadi berikut ini:

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

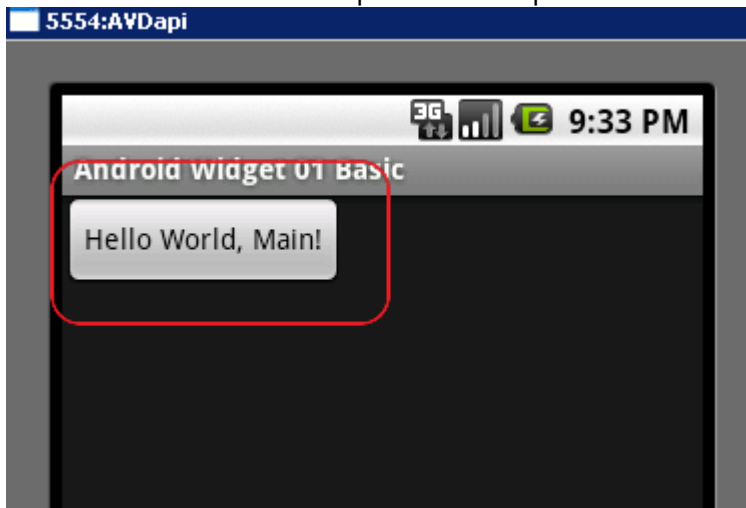
13. Jalankan kembali projectnya, dan akan didapatkan tampilan seperti berikut ini:



14. Untuk mengetahui perbedaan `fill_parent` dan `wrap_content`, kita ganti dalam layout tersebut dengan cara membuat **layout_width** menjadi **wrap_content**. Kode program menjadi seperti berikut ini:

```
main.xml
<?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"
    >
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/hello"
    />
</LinearLayout>
```

15. Jalankan kembali dan akan didapatkan hasil seperti berikut ini:



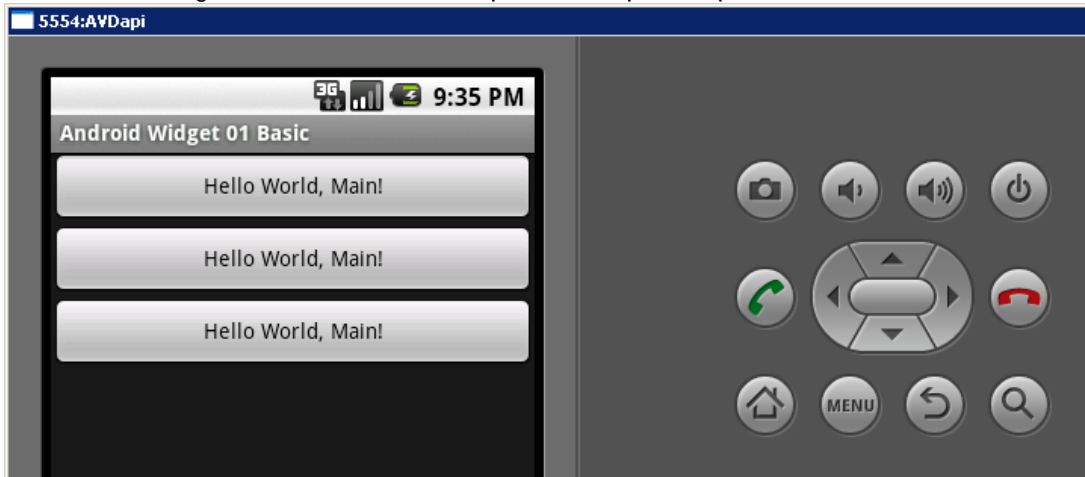
16. Selanjutnya, coba kita tambahkan dua button lagi, edit **main.xml** sehingga menjadi seperti berikut ini:

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



```
<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"
/>
</LinearLayout>
```

17. Kita coba running kembali, dan akan didapatkan tampilan seperti berikut ini:



18. Untuk mencoba Gravity, kita ubah-ubah kode program **main.xml** menjadi seperti berikut ini:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello"
        android:gravity="left"
    />
    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello"
        android:gravity="right"
    />
    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello"
        android:gravity="center"
    />
</LinearLayout>
```

19. Gravity akan menentukan alignment dari suatu teks pada widget, Maka hasilnya akan menjadi seperti berikut ini:



20. Perhatikan berkas main.xml, property android:layout_width, android:layout_height serta fill_parent dan wrap_content. Tak luput pula tentang Gravity, jika pembaca memahami apa yang maksud dari penggunaan tersebut. Selamat!!! Anda telah berhasil belajar dasar widget pada Android.

Kesimpulan

Dari tutorial ini, dapat penulis simpulkan beberapa poin penting berikut ini:

1. Terdapat berkas **main.xml** yang berguna untuk mendesain komponen Widget.
2. **android:layout_width** digunakan untuk menentukan lebar komponen sedangkan **android:layout_height** digunakan untuk menentukan tinggi komponen widget.
3. Nilai **fill_parent** artinya komponen akan di resize penuh sesuai dengan ukuran layar.
4. Nilai **wrap_content** artinya komponen akan diresize sesuai dengan lebar atau tinggi teks.
5. **android:gravity** terdiri dari **left**, **right** dan **center** yang dapat digunakan untuk menentukan alignment atau perataan teks pada suatu komponen widget.

Biografi Penulis



Andi Taru Nugroho Nur Wismono, Lahir di Tuntang, 01 April 1987. Menyelesaikan S1 Fakultas TI-TI pada tahun 2009 dan menyelesaikan S2 Fakultas TI-SI pada tahun 2011. Penulis merupakan founder dari **JavaClopedia.com** juga Founder dan CEO perusahaan IT **EducaStudio** (educastudio.com). Fokus penulis ada pada pemrograman Java baik itu pemrograman **game**, pemrograman **desktop**, pemrograman **mobile** dan pemrograman **enterprise**. Pengalaman belajar Java penulis, dimulai sejak tahun 2005. Ingin konsultasi pemrograman Java dan Android? request tutorial? Kritik dan Saran? Kirimkan email ke andi.taru@gmail.com