

XAMARIN ANDROID : Bekerja Dengan ListView

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Junindar, ST, MCPD, MOS, MCT, MVP

junindar@gmail.com

<http://junindar.blogspot.com>

Abstrak

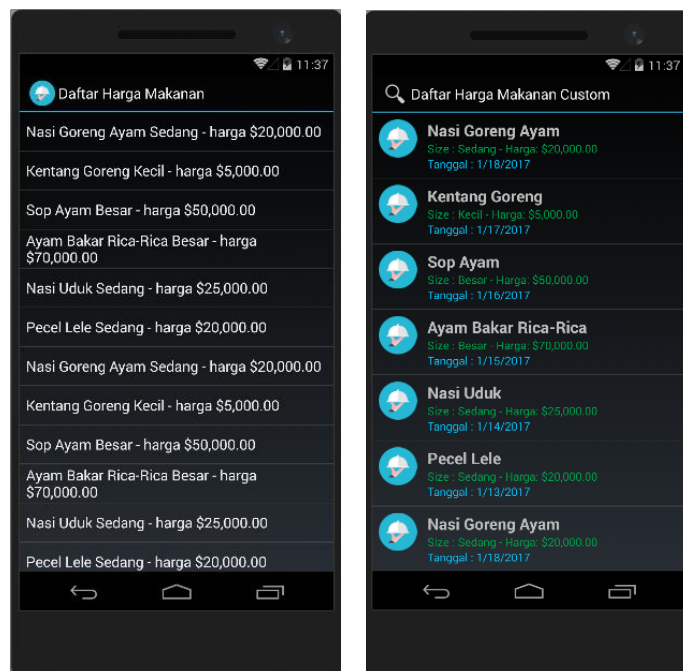
Android adalah salah satu *mobile oprating system* yang saat banyak digunakan di berbagai *smartphone*, baik yang berspesifikasi rendah maupun mumpuni. Dikarenakan android saat ini merupakan salah satu mobile OS paling digemari, maka tidak dipungkiri sangat banyak sekali aplikasi-aplikasi mobile untuk android. Untuk membuat aplikasi mobile ini ada beberapa tools yang bisa digunakan dan salah satunya adalah dengan menggunakan Xamarin.

Pendahuluan

Pada artikel ini akan dijelaskan bagaimana bekerja dengan ListView pada aplikasi android dengan menggunakan Xamarin.

ListView adalah widget yang berfungsi untuk menampilkan data berbentuk item-item dari dari sumber data baik array, List ataupun dari sebuah database yang layoutnya tersusun kebawah ataupun kesamping.

ListView mendapatkan data melalui sebuah Adapter. Adapter adalah sebuah class yang mengatur item-item pada ListView, dan bisa dikatakan sebuah adapter berfungsi untuk mengikat antara ListView dan data yang akan ditampilkan pada ListView. Ada beberapa tipe dari Adapter seperti ArrayAdapter, SimpleAdapter, BaseAdapter dan CursorAdapter. Tools yang kita gunakan untuk membangun aplikasi ini adalah Xamarin. Xamarin adalah salah tools untuk membuat aplikasi mobile dimana bahasa pemograman yang digunakan adalah C#. Xamarin merupakan perusahaan perangkat lunak yang berdiri pada bulan Mei tahun 2011. *Integrated development environment* (IDE) pada Xamarin di kenal dengan nama Xamarin Studio. Tetapi selain Xamarin Studio kita juga dapat menggunakan Visual Studio dan pada artikel ini kita akan menggunakan Visual Studio 2015.



Isi

Pada latihan ini kita akan membuat sebuah aplikasi Mobile dengan menggunakan ListView, dimana ListView bisa digunakan untuk menampilkan data baik berupa text maupun gambar.

Untuk memudahkan memahami isi dari artikel ini, kita akan membuat sebuah project latihan, untuk latihan pertama adalah kita akan menampilkan data berupa pada ListView dengan menggunakan default layout untuk row-nya sedangkan untuk adapter akan menggunakan ArrayAdapter.

Ikuti langkah-langkah dibawah ini.

1. Buat sebuah project Android dengan nama “LatihanListView”.
2. Tambahkan sebuah button pada Main.xml.

```
<Button  
    android:id="@+id/btnListView"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="List View" />
```

Button ini digunakan untuk menampilkan screen ListView yang akan kita buat setelah ini.

3. Tambahkan sebuah layout dengan nama ListViewLayout.xml. Ganti sintaks layout menjadi seperti dibawah.

```
<?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">  
    <ListView  
        android:minWidth="25px"  
        android:minHeight="25px"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        android:id="@+id/makananListView" />  
</LinearLayout>
```

4. Tambahkan sebuah Activity dengan nama ListViewActivity.
5. Sebelum kita bekerja dengan activity yang kita buat diatas, terlebih dahulu kita buat sebuah class dengan nama “Makanan” dan “MakananRepository”.

Class Makanan ini merupakan representasi dari Item-item yang akan ditampilkan oleh screen. Pada class ini kita menggunakan 4 property, masing-masing seperti berikut.

```
public string NamaMakanan { get; set; }  
public double Harga { get; set; }  
public string SizeMakanan { get; set; }  
public DateTime TanggalRelease { get; set; }
```

Karena pada latihan ini kita menggunakan ArrayAdapter, dimana untuk menampilkan data ArrayAdapter ini memanggil method “ToString()”, maka pada kelas Makanan kita harus melakukan Override string seperti dibawah.

```
public override string ToString()  
{  
    return NamaMakanan + " " + SizeMakanan + " - harga  
    " + Harga.ToString("C");  
}
```

Selanjutnya kita akan membuat class Repository dari class Makanan. Untuk diketahui pada latihan ini kita menggunakan POCO (Plain Old CLR Objects) dan *Repository Pattern*.

- Buat sebuah class dengan nama “MakananRepository”. Class ini digunakan untuk mengisi data-data makanan dan disimpan kedalam List <T> Class. Didalam class ini terdapat variable public static seperti dibawah.

```
public static List<Makanan> ListMakanan { get; }
```

Dan buat sebuah static method untuk menambahkan data kedalam ListMakanan dengan menggunakan method Add.

```
private static void TambahMakanan()  
{  
    ListMakanan.Add(new Makanan()  
    {  
        NamaMakanan = "Nasi Goreng Ayam",  
        Harga = 20000,  
        SizeMakanan = "Sedang",  
        TanggalRelease = new DateTime(2017, 01, 18)  
    });  
  
    ListMakanan.Add(new Makanan()  
    {  
        NamaMakanan = "Kentang Goreng",  
        Harga = 5000,  
        SizeMakanan = "Kecil",  
        TanggalRelease = new DateTime(2017, 01, 17)  
    });  
}
```

Pada contoh diatas telah kita buat sebuah method untuk mengisi data kedalam List<T> Class, sebagai latihan tambahkan beberapa data lagi kedalam method diatas. Dan terakhir buat sebuah constructor seperti dibawah. Terdapat method “TambahMakanan” dan pembuatan object baru untuk List<Makanan>. Sehingga pada saat class ini diinisialisasi secara otomatis ListMakananan akan terisi oleh data-data yang telah kita buat pada method TambahMakanan.

```
static MakananRepository()  
{  
    ListMakanan = new List<Makanan>();  
    TambahMakanan();  
}
```

- Langkah terakhir menambahkan sintaks pada Activity. Buka ListViewActivity yang telah kita buat sebelumnya. Buat sebuah event method yang digunakan pada saat item pada ListView diklik. Pada saat item diklik maka akan menampilkan ispada item yang diklik dengan menggunakan Toast.

```
void listViewMakanan_ItemClick(object sender,  
AdapterView.ItemClickEventArgs e)  
{  
    Toast.MakeText(this,  
MakananRepository.ListMakanan[e.Position].ToString(),  
ToastLength.Short).Show();  
}
```

Selanjutnya tambahkan sintaks dibawah ini, pada method OnCreate().

```
SetContentView(Resource.Layout.ListViewLayout);  
var listViewMakanan = FindViewById<ListView>  
(Resource.Id.makananListView);  
var makananAdapter = new ArrayAdapter<Makanan>(this, An-  
droid.Resource.Layout.SimpleListItem1, MakananReposito-  
ry.ListMakanan);  
listViewMakanan.Adapter = makananAdapter;  
listViewMakanan.ItemClick += listViewMakanan_ItemClick;
```

Dapat dilihat pada sintaks diatas, kita membuat sebuah ArrayAdapter dengan nama “makananAdapter”. Dan untuk mengambil datanya kita gunakan class MakananRepository yang telah kita buat sebelumnya. Lalu ArrayAdapter tersebut akan digunakan pada listViewMakananan (ListView). Dan pada baris terakhir adalah dimana kita akan menambahkan sebuah event ItemClick untuk ListView. Untuk melihat hasilnya jalankan program latihan ini. Jika berhasil tampilan pada screen akan seperti gambar dibawah. Dimana setiap item terdiri dari 1 baris yang isinya Nama

Makanan , Ukuran (Size) dan harga atau sesuai dengan override string ToString() yang telah kita buat pada class Makanan sebelumnya.

Jika dilihat tampilah dari ListView ini terlalu standard atau biasa saja, lalu bagaimana untuk memperindah tampilah dari ListView? Untuk mengubah tampilah dari ListView kita dapat menggunakan custom layout untuk menampung setiap item. Sebagai contoh bisa kita tambahkan sebuah gambar pada setiap item, atau mengganti font untuk masing-masing data-data yang ditampilkan disetiap item. Misalnya nama makanan menggunakan font yang lebih besar dari harga, dan mengganti warna font untuk setiap datanya. Setelah berhasil membuat latihan ListView pertama ini, maka lanjutkan dengan latihan berikutnya.



Untuk latihan kedua pada ListView, kita akan menggunakan Activity baru dengan nama ListViewCustomActivity.

1. Sebelumnya buat sebuah layout dengan nama customRowListView.xml. Pada layout ini terdapat sebuah ImageView dan tiga buah TextView. Untuk detailnya dapat dilihat pada sintaks xml dibawah.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="8dp">
    <ImageView
        android:id="@+id/item_icon"
        android:layout_width="48dp"
        android:layout_height="48dp"
        android:layout_marginRight="8dp" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <TextView
            android:id="@+id/makananTextView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="20dp"
            android:textStyle="bold"
            android:paddingLeft="5dp" />
        <TextView
            android:id="@+id/hargaTextView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textColor="#00A14B"
            android:paddingLeft="5dp" />
        <TextView
            android:id="@+id/tanggalTextView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textColor="#0cbdf5"
            android:paddingLeft="5dp" />
    </LinearLayout>
</LinearLayout>
```

Dapat dilihat pada layout diatas terdapat dua buah LinearLayout, LinearLayout yang pertama menggunakan orientasi Horizontal dan terdapat ImageView dan LinearLayout kedua. Sedangkan LinearLayout yang kedua menggunakan orientasi vertical dan terdapat tiga buah TextView didalamnya.

2. Jika pada latihan pertama kita menggunakan ArrayAdapter, pada custom ListView ini kita akan menggunakan BaseAdapter dengan membuat custom Adapter sendiri dengan nama "MakananAdapter" yang nantinya akan digunakan oleh ListView melalui adapter. Pada MakananAdapter ini, kita akan melakukan inherits dari

BaseAdapter, sehingga pada class ini akan secara otomatis terdapat 4 buah method, sebagai berikut :

- GetItemId(int position) : Mendapatkan Id berdasarkan dari posisi item saat ini.
 - GetView(int position, View convertView, ViewGroup parent) : Method yang paling penting didalam adapter. Dengan method ini kita akan melakukan override baris item pada ListView. Dengan menggunakan LayoutInflater, kita akan mengidentifikasi layout apa yang akan digunakan untuk mengisi baris item pada ListView. Pada latihan ini kita akan menggunakan “customRowListView.axml” yang telah kita buat sebelumnya.
 - Count : Jumlah item yang ada pada ListView.
 - this[int position] : merupakan array yang menggunakan index (posisi) untuk menampilkan data.
3. Buat sebuah class dengan nama MakananAdapter dan inherits BaseAdapter<T> class
- ```
public class MakananAdapter : BaseAdapter<Makanan>
{
}
```

Setelah kita membuat class seperti diatas, maka class tersebut akan error. Seperti yang telah dijelaskan diatas terdapat beberapa method didalam Adapter, sehingga kita harus membuat method-method tersebut kedalam class kita. Dengan menggunakan Visual Studio kita dapat menggenerate method-method tersebut secara otomatis.



```
public override Makanan this[int position]
{
 get
 {
 throw new NotImplementedException();
 }
}

public override int Count
{
 get
 {
 throw new NotImplementedException();
 }
}

public override long GetItemId(int position)
{
 throw new NotImplementedException();
}

public override View GetView(int position, View
convertView, ViewGroup parent)
{
 throw new NotImplementedException();
}
```

Pada buku ini akan dijelaskan sintaks didalam method GetView saja, sedangkan untuk detail sintaks dapat dilihat pada file project latihan.

```
var view = convertView ?? _context.LayoutInflater.Inflate
(Resource.Layout.customRowListView, parent, false);
var makananTextView = view.FindViewById<TextView>
(Resource.Id.makananTextView);
var hargaTextView = view.FindViewById<TextView>
(Resource.Id.hargaTextView);
var tanggalTextView = view.FindViewById<TextView>
(Resource.Id.tanggalTextView);
var iconImage = view.FindViewById<ImageView>
(Resource.Id.item_icon);
iconImage.SetImageResource
(Resource.Drawable.makanan);
makananTextView.Text = _listMakanan[position].NamaMakanan;
hargaTextView.Text = "Size : " +
_listMakanan[position].SizeMakanan + " - Harga: " +
_listMakanan[position].Harga.ToString("C");
tanggalTextView.Text = "Tanggal : " +
_listMakanan[position].TanggalRelease.ToShortDateString();

return view;
```

Pada baris pertama adalah sintaks untuk mengecek apakah View bernilai null atau tidak. Dan jika bernilai null makan view akan diisi oleh layout

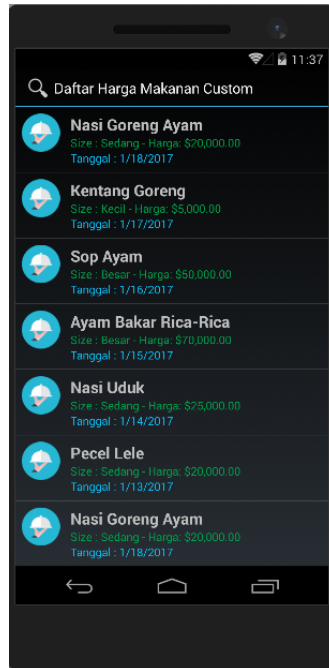
“customRowListView”. Selanjutnya method yang sudah biasa kita gunakan yaitu FindViewById untuk setiap widget yang ada pada layout.

Setelah itu untuk ImageView kita akan SetImage dari Resource > Drawable. Sedangkan untuk textview akan di set property text nya sesuai yang diinginkan. Sebagai contoh untuk makananTextView akan diisi dengan property NamaMakanan, sedangkan hargaTextView akan diisi oleh Size (Ukuran) dan harga sesuai dengan sintaks diatas. Sampai disini kita telah membuat custom Adapter yang akan digunakan oleh ListView pada Activity yang akan kita buat.

4. Buat sebuah activity baru dengan nama “ListViewCustomActivity”. Sama seperti latihan pertama buat sebuah event handler item click untuk ListView.
5. Dan pada method OnCreate(), kita akan menambahkan beberapa baris sintaks seperti dibawah.

```
SetContentView(Resource.Layout.ListViewLayout);
var listViewMakanan =
FindViewById<ListView>(Resource.Id.makananListView);
var makananAdapter = new MakananAdapter(this,
MakananRepository.ListMakanan);
listViewMakanan.FastScrollEnabled = true;
listViewMakanan.Adapter = makananAdapter;
listViewMakanan.ItemClick +=
listViewMakanan_ItemClick;
```

Dapat kita lihat pada sintaks diatas, kita masih menggunakan layout yang sama seperti pada latihan yang pertama. Nah pada adapter terdapat perbedaan dengan latihan pertama, dimana pada latihan ini kita tidak menggunakan ArrayAdapter, melainkan custom adapter yang telah kita buat sebelumnya. Lalu kita menggunakan method FastScrollEnabled dengan mengubah valuenya menjadi true.

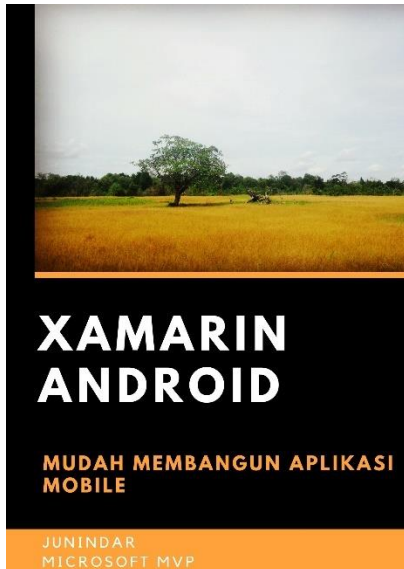


## **Penutup**

Sedangkan untuk memudahkan dalam memahami isi artikel, maka penulis juga menyertakan dengan full source code project latihan ini, dan dapat di download disini

<http://junindar.blogspot.co.id/2018/04/xamarin-android-bekerja-dengan-listview.html>

## Referensi



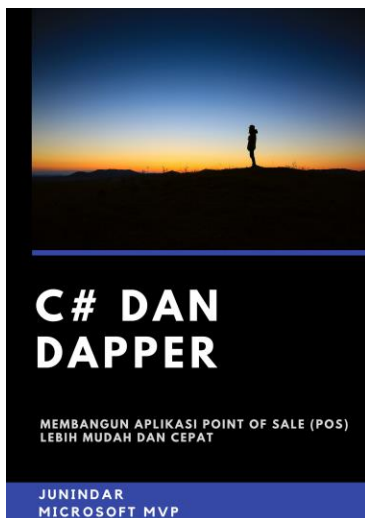
<https://play.google.com/store/books/details?id=G4tFDgAAQBAJ>



<https://play.google.com/store/books/details?id=VSLiDQAAQBAJ>



[https://play.google.com/store/books/details/Junindar\\_Xamarin\\_Forms?id=6Wg-DwAAQBAJ](https://play.google.com/store/books/details/Junindar_Xamarin_Forms?id=6Wg-DwAAQBAJ)



[https://play.google.com/store/books/details/Junindar\\_C dan Dapper Membangun Aplikasi POS Point?id=6TErDwAAQBAJ](https://play.google.com/store/books/details/Junindar_C_dan_Dapper_Membangun_Aplikasi_POS_Point?id=6TErDwAAQBAJ)

## Biografi Penulis.



Junindar Lahir di Tanjung Pinang, 21 Juni 1982. Menyelesaikan Program S1 pada jurusan Teknik Inscreenatika di Sekolah Tinggi Sains dan Teknologi Indonesia (ST-INTEN-Bandung). Junindar mendapatkan Award Microsoft MVP VB pertanggal 1 oktober 2009 hingga saat ini. Senang mengutak-atik computer yang berkaitan dengan bahasa pemrograman. Keahlian, sedikit mengerti beberapa bahasa pemrograman seperti : VB.Net, C#, SharePoint, ASP.NET, VBA. Reporting: Crystal Report dan Report Builder. Database: MS Access, MY SQL dan SQL Server. Simulation / Modeling Packages: Visio Enterprise, Rational Rose dan Power Designer. Dan senang bermain gitar, karena untuk bisa menjadi pemain gitar dan seorang programmer sama-sama membutuhkan seni. Pada saat ini bekerja di salah satu Perusahaan Consulting dan Project Management di Malaysia sebagai Senior Consultant. Memiliki beberapa sertifikasi dari Microsoft yaitu Microsoft Certified Professional Developer (MCPD – SharePoint 2010), MOS (Microsoft Office Specialist) dan MCT (Microsoft Certified Trainer) Mempunyai moto hidup: **“Jauh lebih baik menjadi Orang Bodoh yang giat belajar, dari pada orang Pintar yang tidak pernah mengimplementasikan ilmunya”**.