

# **XAMARIN ANDROID : Bekerja Dengan Expandable 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.

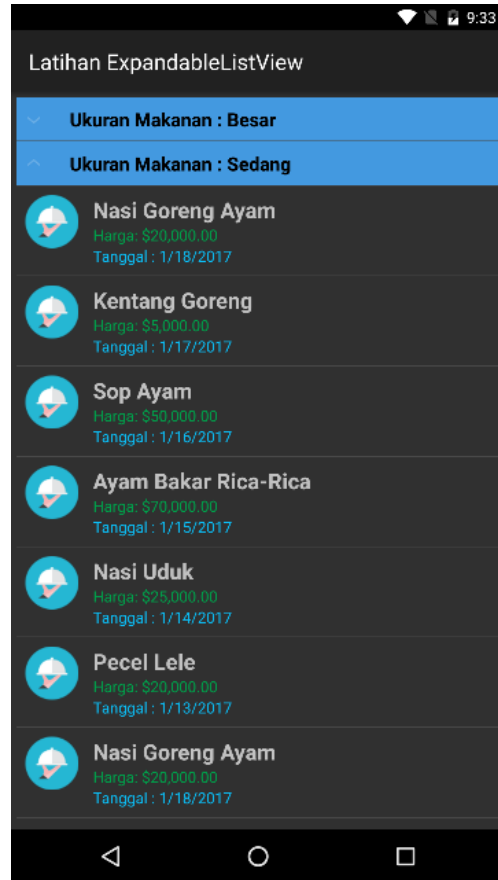
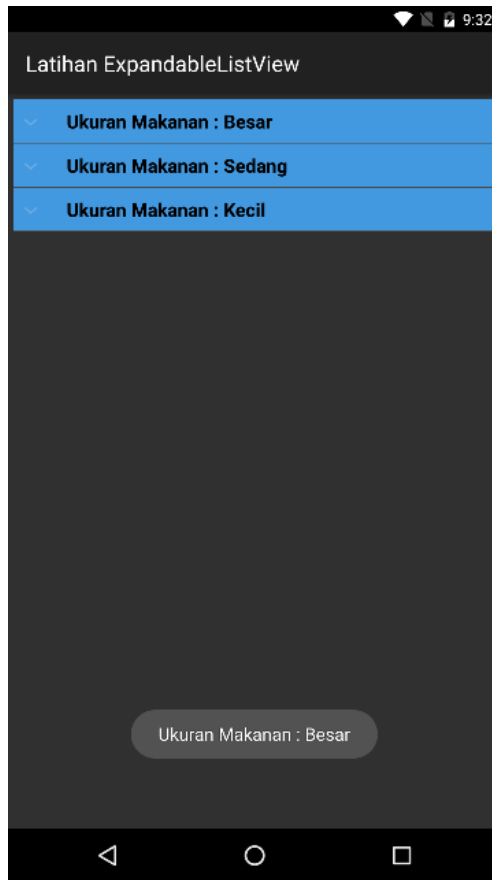
## Pendahuluan

Sebelum memulai latihan pada artikel ini, disarankan untuk terlebih dahulu membaca dan membuat latihan pada artikel sebelumnya yang bisa didapatkan disini <http://junindar.blogspot.com/2018/04/xamarin-android-bekerja-dengan-listview.html> .

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

ExpandableListView adalah widget yang fungsinya hampir sama dengan ListView yang telah dijelaskan sebelumnya. Perbedaan dari kedua widget ini adalah, pada ExpandableListView terdapat fungsi expand dan collapsible data. Hal ini berguna jika kita mau mengelompokkan data sesuai dengan groupnya. Sebagai contoh pada latihan ListView yang sebelumnya kita memiliki data daftar makanan, kalau kita lihat disetiap item terdapat Size/ukuran dari makanan. Untuk mempermudah pengguna dalam menggunakan aplikasi atau mencari daftar makanan, kita dapat melakukan group makanan berdasarkan size/ukuran dari makanan tersebut. Jadi untuk menggunakan ExpandableListView kita memerlukan layout untuk header sebagai group dari daftar makanan. Pada ExpandableListView kita juga akan menggunakan custom Adapter, dengan melakukan inherits ke “BaseExpandableListAdapter”. Jika pada BaseAdapter hanya terdapat 4 method saja, maka pada BaseExpandableListAdapter terdapat 10 method yang harus dibuat setelah kita melakukan inherits. Untuk memudahkan dalam membahas widget ini, kita akan melakukan latihan membuat screen yang menggunakan ExpandableListView.

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

ExpandableListView adalah widget yang fungsinya hampir sama dengan ListView yang telah dijelaskan sebelumnya. Perbedaan dari kedua widget ini adalah, pada ExpandableListView terdapat fungsi expand dan collapsible data. Hal ini berguna jika kita mau mengelompokkan data sesuai dengan groupnya.

Untuk memudahkan memahami isi dari artikel ini, kita akan membuat sebuah project latihan.

Ikuti langkah-langkah dibawah ini.

1. Buat sebuah project Android dengan nama “Latihan ExpandableListView”.
2. Ganti sintak pada Main.xml seperti dibawah ini.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/andro
id"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
    <ExpandableListView
        android:id="@+id/expandLV"
        android:layout_height="match_parent"
        android:layout_width="match_parent"
        android:layout_margin="5dp" />
</LinearLayout>
```

Dapat dilihat pada sintaks diatas, kita tambahkan sebuah component ExpandableListView pada screen dengan nama “expandLV”

3. Jika pada custom ListView pada latihan sebelumnya kita hanya memerlukan sebuah layout (customRowListView.xml) untuk menampilkan item pada ListView. Sedangkan untuk ExpandableListView kita juga memerlukan layout untuk header pada setiap groupnya. Tambahkan sebuah layout dengan nama HeaderExpandLayout dan ganti sintaks nya seperti dibawah.

```
<?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="wrap_content"
    android:orientation="vertical"
    android:padding="8dp"
    android:background="#4399e0">
    <TextView
        android:id="@+id/lblLvHeader"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingLeft="?android:attr/expandableListPreferredItemPaddingLeft"
        android:textSize="17dp"
        android:textColor="#000000"
        android:textStyle="bold" />
</LinearLayout>
```

Pada layout diatas terdapat sebuah TextView untuk menampilkan data Group (Size). Sedangkan untuk layout ini (customRowListView.xml) dapat dilihat pada latihan sebelumnya.

4. Tambahkan sebuah class dengan nama "Size".

```
public class Size
{
    public string SizeMakanan { get; set; }
    public List<Makanan> MakananDetails { get; set; } = new List<Makanan>();
    public override string ToString()
    {
        return "Ukuran Makanan : " + SizeMakanan;
    }
}
```

Pada class ini kita telah membuat relationship antara class ini dan class makanan atau yang lebih dikenal dengan Asosiasi. Relasi ini memungkinkan kita untuk membuat relasi yang biasa dikenal parentchild, dimana satu Size bisa memiliki banyak Makanan. Seperti pada kelas makanan, pada class ini juga dilakukan override string ToString(), untuk digunakan pada group ataupun header pada widget. Buat sebuah class SizeRepository, yang digunakan untuk menambahkan data Size pada List<T> Class. Untuk detail sintaks bisa dilihat pada file latihan (SizeRepository.cs).

5. Selanjutnya kita akan membuat custom Adapter, tambahkan sebuah class dengan nama "SizeAdapter", seperti dibawah.

```
public class SizeAdapter : BaseExpandableListAdapter
{
    private readonly Activity _context;
    private readonly List<Size> _listSize;

    public SizeAdapter(Activity context, List<Size> listsize)
    {
        _context = context;
        _listSize = listsize;
    }
}
```

Untuk menghilangkan error pada class, jangan lupa untuk meng-*generate* method-method bawaan dari BaseExpandableListAdapter. Terdapat dua method yang penting pada class ini yaitu GetGroupView dan GetChildView. Untuk method GetGroupView kita akan melakukan override item pada group. Dengan menggunakan LayoutInflater, kita akan mengidentifikasi layout (HeaderExpandLayout.xml) yang akan digunakan untuk mengisi setiap group. Sedangkan pada GetChildView fungsinya sama seperti GetView pada BaseAdapter.

```
public override View GetGroupView(int groupPosition, bool isExpanded,
View convertView, ViewGroup parent)
{
    convertView = convertView ?? _context.LayoutInflater.Inflate
        (Resource.Layout.HeaderExpandLayout, null);
    var headerTitle = (string)GetGroup(groupPosition);
    var lblListHeader = (TextView)convertView.FindViewById(Resource.Id.lblLvHeader);
    lblListHeader.Text = headerTitle;

    return convertView;
}
```

Untuk mendapatkan data kita gunakan method GetGroup(int groupPosition) berdasarkan posisi dari group.

```
public override long GetGroupId(int groupPosition)
{
    return groupPosition;
}
```

Selanjutnya sintaks GetChildView dapat dilihat dibawah.

```
public override View GetChildView(int groupPosition, int childPosition,
bool isLastChild, View convertView, ViewGroup parent)
{
    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);

    var makanan = _listSize[groupPosition].MakananDetails[childPosition];

    iconImage.SetImageResource(Resource.Drawable.makanan);
    makananTextView.Text = makanan>NamaMakanan;
    hargaTextView.Text = "Harga: " + makanan.Harga.ToString("C");
    tanggalTextView.Text = "Tanggal : " + makanan.TanggalRelease.ToShortDateString();

    return view;
}
```

Seperti diberitahukan sebelumnya method ini hampir sama seperti GetView pada BaseAdapter, hanya pada saat mendapatkan data, kita menggunakan groupPosition dan ChildPosition. Pada penjelasan sebelumnya kita telah membuat asosiasi antar class Size dan Makanan. Dimana satu Size akan terdapat banyak Makanan. Jadi untuk mendapatkan data kita memerlukan dua variable, satu untuk posisi parent dan satu lagi untuk posisi child. Untuk detail sintaks adapter dapat dilihat pada file “SizeAdapter.cs”.

6. Selanjutnya buka MainActivity.cs Pada method OnCreate yang perlu kita lakukan adalah dengan membuat object custom Adapter dengan menggunakan SizeAdapter. Selanjutnya dengan menggunakan method SetAdapter pada ExpandableListView kita set listAdapter kedalamnya.

```
SetContentView(Resource.Layout.Main);
_expListView = FindViewById<ExpandableListView>(Resource.Id.expandLV);
var listAdapter = new SizeAdapter(this, SizeRepository.ListSize);
_expListView.SetAdapter(listAdapter);
_expListView.FastScrollEnabled = true;
ClickEvents();
```

Lalu kita buat method untuk event-event yang ada pada widget. Seperti ChildClick, GroupCollapse dan GroupExpand.

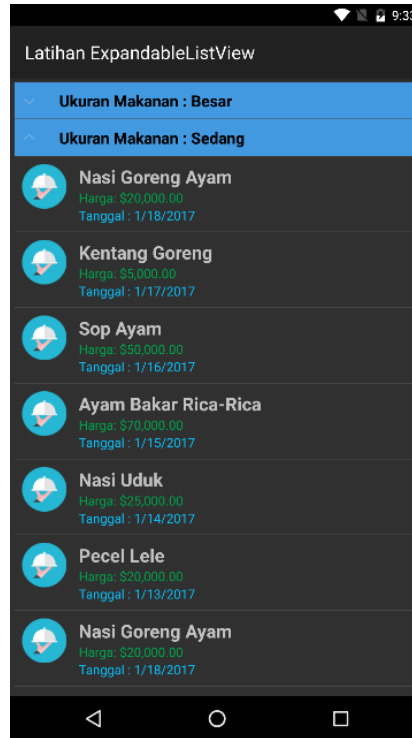
```
void ClickEvents()
{
    _expListView.ChildClick += delegate
    (object sender, ExpandableListView.ChildClickEventArgs e)
    {
        Toast.MakeText(this, SizeRepository.ListSize[e.GroupPosition].
        MakananDetails[e.ChildPosition].ToString(), ToastLength.Short).Show();
    };

    _expListView.GroupCollapse += delegate (object sender,
    ExpandableListView.GroupCollapseEventArgs e)
    {
        var i = e.GroupPosition;
        if (i == -1) return;
        Toast.MakeText(this, SizeRepository.ListSize[i].ToString(),
        ToastLength.Short).Show();
    };

    _expListView.GroupExpand += delegate (object sender,
    ExpandableListView.GroupExpandEventArgs e) {
        int previousGroup = -1;
        if (e.GroupPosition != previousGroup)
        _expListView.CollapseGroup(previousGroup);
        var i = e.GroupPosition;
        if (i == -1) return;
        Toast.MakeText(this, SizeRepository.ListSize[i].ToString(),
        ToastLength.Short).Show();
    };
}
```

Jalankan program untuk melihat hasil dari latihan ini. Terdapat tiga group yaitu Besar, Sedang dan Kecil. Jika salah satu group di Expand maka group yang lain akan otomatis collapse dengan sendirinya. Jika item pada Child diklik maka akan keluar pesan sesuai dengan item yang diklik.



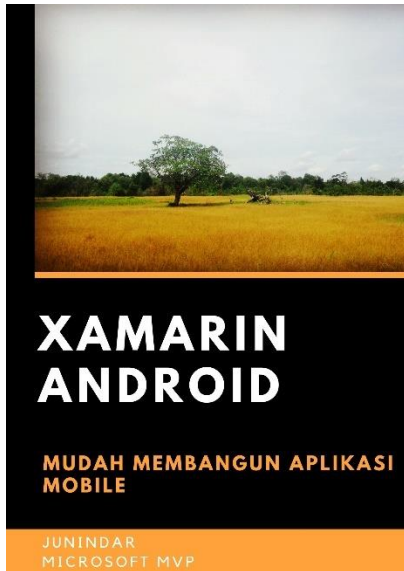


## **Penutup**

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.com/2019/01/xamarin-android-bekerja-dengan.html>

## Referensi



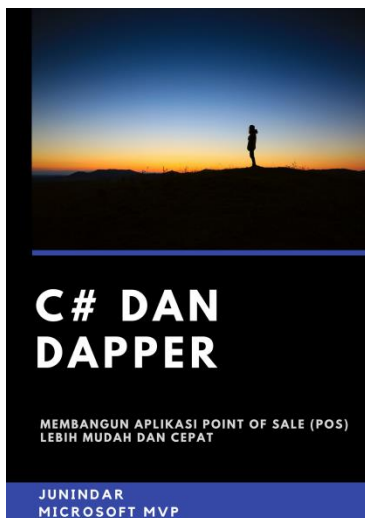
<https://play.google.com/store/books/details?id=G4tFDgAAQBAJ>



<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)



[https://play.google.com/store/books/details/Junindar\\_ASP\\_NET\\_MVC\\_Membangun\\_Aplikasi\\_Web\\_Lebih?id=XLlyDwAAQBAJ](https://play.google.com/store/books/details/Junindar_ASP_NET_MVC_Membangun_Aplikasi_Web_Lebih?id=XLlyDwAAQBAJ)

## 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”**.