

XAMARIN ANDROID : Membangun Aplikasi Mobile Dengan GridView

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

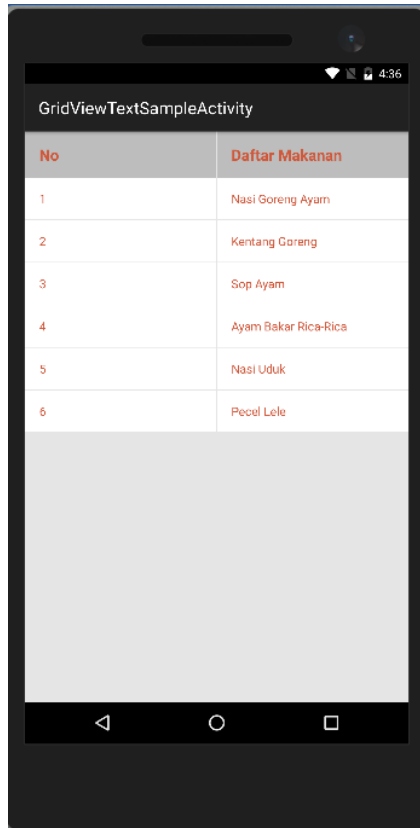
Android adalah salah satu *mobile oprating system* yang saat banyak digunakan di berbagai *smartphone*, baik yang berspesifikasi rendah maupun mumpuni. Dikarenakan android saat ini merupakan salah satu mobile OS paling digemari, maka tidak dipungkiri sangat banyak sekali aplikasi-aplikasi mobile untuk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

GridView merupakan komponen pada android yang menampilkan data dengan format seperti pada table, yaitu dengan terdiri dari beberapa kolom dan baris. Widget ini juga sering digunakan untuk menampilkan data dengan format gambar maupun gambar dan text. Untuk mengatur jumlah kolom pada GridView kita dapat menggunakan attribute “android:numColumns”.

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 GridView, dimana GridView bisa digunakan untuk menampilkan data baik berupa text maupun gambar.

Untuk memudahkan memahami isi dari artikel ini, kita akan membuat sebuah project latihan, dimana latihan pertama adalah kita akan menampilkan data berupa text, sehingga menghasilkan output seperti Table (contoh :Gambar kiri diatas).

Ikuti langkah-langkah dibawah ini.

1. Buat sebuah project Android dengan nama "LatihanXamarin.GridViewSample".
2. Tambahkan sebuah button pada Main.axml.

```
<Button  
    android:id="@+id/btnGridViewText"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Grid View Text" />
```

Button ini digunakan untuk menampilkan screen GridViewText yang akan kita buat

setelah ini.

3. Buat sebuah layout dengan nama GridViewTextLayout.xml. Ganti sintaks layout menjadi seperti dibawah.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#e5e5e5"
    android:orientation="vertical">
    <GridView
        android:id="@+id/gridviewHeader"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:numColumns="2"
        android:horizontalSpacing="2dp"
        android:verticalSpacing="2dp">
    </GridView>
    <GridView
        android:id="@+id/gridviewContent"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:horizontalSpacing="2dp"
        android:verticalSpacing="2dp"
        android:numColumns="2" >
    </GridView>
</LinearLayout>
```

Perhatikan pada sintaks diatas, terdapat dua buah GridView yang pertama kita gunakan untuk Header dan GridView kedua digunakan untuk Content, masing-masing GridView diatas memiliki 2 kolom (android:numColumn="2"). Agar pada GridView memiliki seperti garis/pembatas antara cell, kita gunakan atribut Horizontal dan VerticalSpacing="2dp". GridView menggunakan Adapter untuk mengatur item-item pada GridView. Sebelum masuk kedalam Adapter, kita buat terlebih dahulu class yang nantinya akan digunakan oleh Adapter.

4. Buat sebuah folder dengan nama DataAccess, selanjutnya didalam folder tersebut buat sebuah class dengan nama "ImageItem".

```
public class ImageItem
{
    public string Title { get; set; }
}
```

Dan buat sebuah class lagi didalam file yang sama seperti dibawah.

```
public static class ImageItemRepository
{
    public static List<ImageItem> Images { get; }

    private static void TambahImage()
    {
        ImageItem entity = new ImageItem
        {
            Title = "Nasi Goreng Ayam",
        };
        Images.Add(entity);
        entity = new ImageItem
        {
            Title = "Kentang Goreng",
        };
        Images.Add(entity);
        entity = new ImageItem
        {
            Title = "Sop Ayam",
        };
        Images.Add(entity);
    }
    static ImageItemRepository()
    {
        Images = new List<ImageItem>();
        TambahImage();
        TambahImage();
        TambahImage();
    }
}
```

Class ImageItem merupakan representasi entity dari sebuah objek, yang memiliki sebuah property yaitu Title. Sedangkan untuk class ImageItemRepository adalah sebuah class yang mengandung perilaku dari sebuah objek. Pada class tersebut terdapat List<T> yang digunakan untuk menampung collection dari objek-objek ImageItem. Dan untuk menambahkan kedalam List<T>, kita buat sebuah method dengan nama TambahImage. Selanjutnya method tersebut kita letakkan dalam sebuah constructor.

5. Untuk mengatur item pada GridView kita buat layout tersendiri, dan dikarenakan kita menggunakan dua buah GridView, maka custom layout pun kita buat dua buah juga. Dengan nama **grid_header_layout.xml** dan **grid_content_layout.xml**. Untuk detail sintaks xml nya dapat dilihat pada project lampiran.
6. Tambahkan sebuah folder pada project dengan nama Adapter. Tambahkan sebuah

class dengan nama GridViewHeaderAdapter. Selanjutnya class ini akan menggunakan class BaseAdapter.

```
public class GridViewHeaderAdapter : BaseAdapter
{
    readonly List<string> _header;
    private readonly Activity _context;
    public GridViewHeaderAdapter(Activity context)
    {
        _context = context;
        _header = new List<string> {"No", "Daftar Makanan"};
    }
}
```

Karena class ini melakukan inherits ke class BaseAdapter, maka ada beberapa method yang harus kita ikut sertakan kedalam class ini.

1. GetItemId(int position) : Mendapatkan Id berdasarkan dari posisi item saat ini.
2. GetView(int position, View convertView, ViewGroup parent) : Method yang paling penting didalam adapter. Dengan method ini kita akan melakukan override baris item pada GridView. Dengan menggunakan LayoutInflater, kita akan mengidentifikasi layout apa yang akan digunakan untuk mengisi baris item.
3. Count : Jumlah item yang ada pada objek.
4. this[int position] : merupakan array yang menggunakan index (posisi) untuk menampilkan data.

```
public override Object GetItem(int position)
{
    return _header[position];
}

public override long GetItemId(int position)
{
    return position;
}

public override int Count => _header.Count;

public override View GetView(int position, View convertView, ViewGroup parent)
{
    var view = convertView ?? _context.LayoutInflater.Inflate
        (Resource.Layout.grid_header_layout, parent, false);

    var lblHeader = view.FindViewById<TextView>(Resource.Id.lblHeader);

    lblHeader.Text = _header[position];

    return view;
}
```

Pada method `GetView` kita menggunakan layout `grid_header_layout`. Selanjutnya buat juga adapter untuk content dengan nama `GridViewContentAdapter`. Untuk detail sintaksnya dapat dilihat pada project lampiran.

7. Selanjutnya buat sebuah Activity dengan nama `GridViewTextActivity`. Tambahkan sebuah variable dengan scope class dan sebuah method seperti dibawah ini.

```
private List<ImageItem> _lst;

private void FillNewData()
{
    _lst=new List<ImageItem>();
    var images = ImageItemRepository.Images;
    for (int i = 0; i <= images.Count - 1; i++)
    {
        var entity = new ImageItem {Title = (i + 1).ToString()};
        _lst.Add(entity);

        entity = new ImageItem {Title = images[i].Title};
        _lst.Add(entity);
    }
}
```

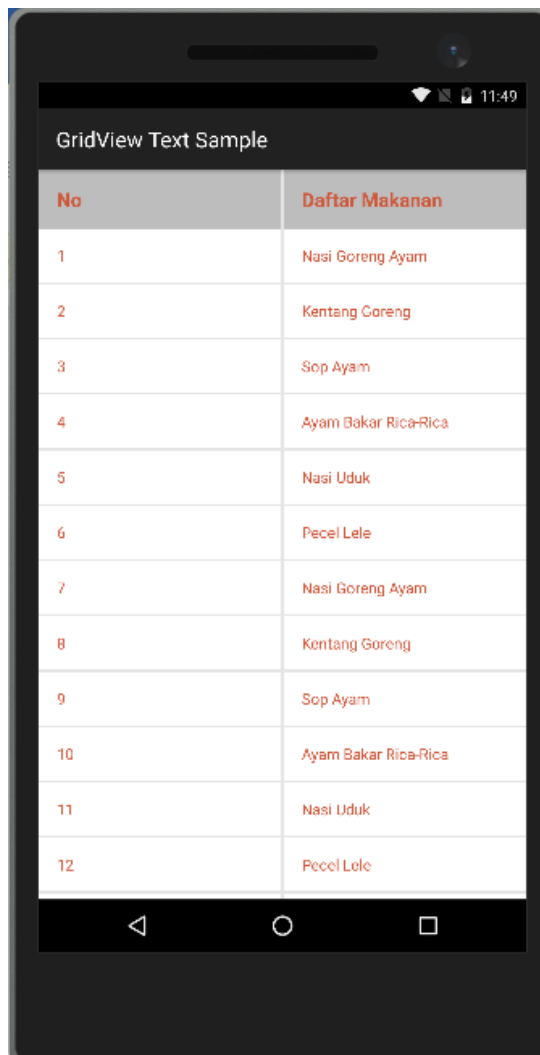
Dikarenakan kita akan membuat dua buah kolom pada `GridView` kolom pertama merupakan nomor dan kolom kedua ada `Title`. Sehingga data dari `ImageItemRepository.Images` akan kita *retrieve* kembali, dan akan kita masukkan kedalam variable yang saja kita buat sebelumnya. Dari proses looping diatas, kita gunakan `Count` untuk mendapat nilai (nomor) yang akan digunakan pada kolom

pertama pada GridView.

8. Sedangkan pada method onCreate, ketikkan sintaks dibawah ini.

```
SetContentView(Resource.Layout.GridViewTextLayout);  
FillNewData();  
var gridView = FindViewById<GridView>(Resource.Id.gridViewHeader);  
var adapter = new GridViewHeaderAdapter(this);  
gridView.Adapter = adapter;  
  
var gridViewContent = FindViewById<GridView>(Resource.Id.gridViewContent);  
var adapterContent = new GridViewContentAdapter(this, _lst);  
gridViewContent.Adapter = adapterContent;
```

Terdapat dua buah adapter yang masing-masing digunakan oleh gridView pada layout. Jalan kan program untuk melihat hasilnya.



Latihan kedua adalah dengan membuat GridView untuk menampilkan Gambar dan Text sekaligus. Sebelumnya tambahkan image-image yang diperlukan kedalam folder “drawable”.

1. Buka Class ImageItem, tambahkan sebuah property seperti berikut

```
public int ImageId { get; set; }
```

Dan pada Class ImageItemRepository, method TambahImage tambahkan ImageId untuk setiap create objek entity.

```
entity = new ImageItem  
{  
    Title = "Kentang Goreng",  
    ImageId = Resource.Drawable.kentang_goreng  
};
```

Pada property ImageId kita assign nilai Id dari image yang telah kita tambahkan sebelumnya.

2. Buat sebuah layout dengan nama GridViewImageLayout. Layout untuk menampung GridView pada screen, sedangkan untuk item buat sebuah layout dengan nama grid_item_image_layout. Kedua layout ini dapat dilihat pada project lampiran.
3. Tambah sebuah class pada folder Adapter dengan nama GridViewImageAdapter. Untuk detail sintaks lihat pada project lampiran. Pada dasarnya konsep dan ini hampir sama seperti pada Adapter-Adapter sebelumnya. Pada adapter ini menggunakan layout “grid_item_image_layout” yang memiliki dua buah widget (ImageView dan TextView.)

```
public override View GetView(int position, View convertView, ViewGroup parent)  
{  
    var view = convertView ?? _context.LayoutInflater.Inflate  
        (Resource.Layout.grid_item_image_layout, parent, false);  
  
    var lblTitle = view.FindViewById<TextView>(Resource.Id.lblTitle);  
    var imgView = view.FindViewById<ImageView>(Resource.Id.imgView);  
  
    imgView.SetImageResource(_images[position].ImageId);  
    lblTitle.Text = _images[position].Title;  
  
    return view;  
}
```

4. Tambahkan sebuah Activity dengan nama GridViewImageActivity. Lalu pada OnCreate ketikkan sintaks seperti dibawah ini.

```
protected override void OnCreate(Bundle savedInstanceState)
{
    base.OnCreate(savedInstanceState);

    SetContentView(Resource.Layout.GridViewImageLayout);
    var gridView = FindViewById<GridView>(Resource.Id.gridView);
    var adapter = new GridViewImageAdapter(this, ImageItemRepository.Images);
    gridView.Adapter = adapter;
}
```

Jalankan program untuk melihat hasilnya.

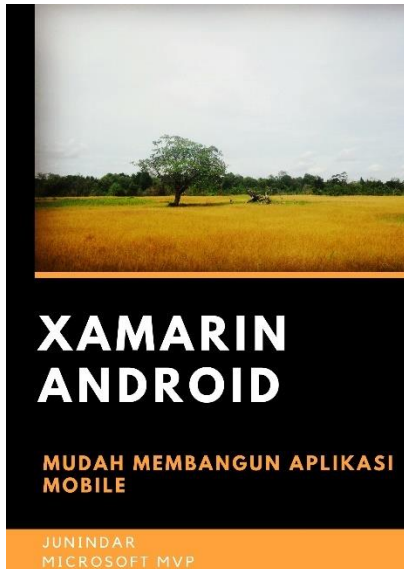


Penutup

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/2017/03/xamarin-android-membangun-aplikasi.html>

Referensi



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



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

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”**.