

XAMARIN ANDROID : Bekerja Dengan RecyclerView

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/2019/02/xamarin-android-bekerja-dengan-cardview.html>

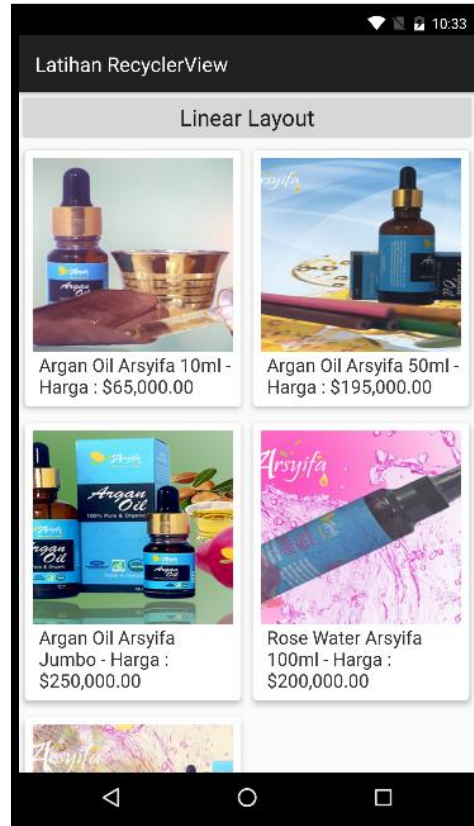
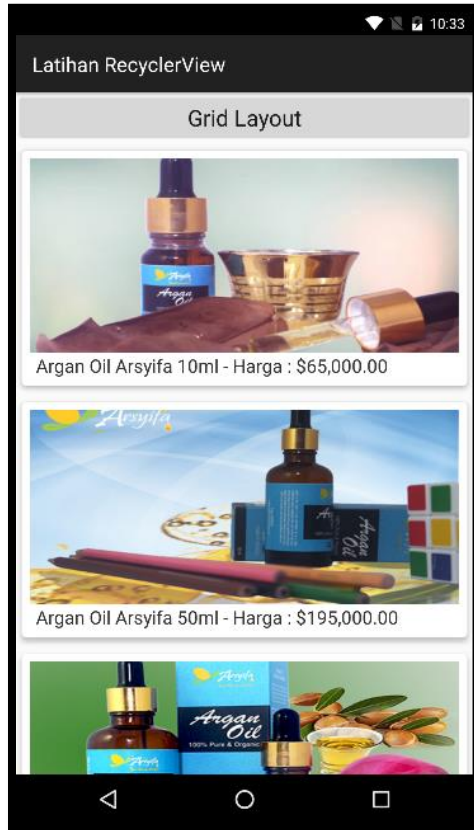
<http://junindar.blogspot.com/2018/04/xamarin-android-bekerja-dengan-listview.html> dan

<http://junindar.blogspot.com/2019/01/xamarin-android-bekerja-dengan.html>.

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

RecyclerView adalah widget yang digunakan untuk menampilkan sebuah daftar yang berisi banyak item didalam list maupun grid. Secara bentuk dan fungsi widget ini hampir sama dengan ListView, tetapi secara konsep jauh berbeda. RecyclerView digunakan untuk menampilkan data yang berjumlah lebih banyak. Widget ini diperkenalkan pada Android version 5.0 (Lollipop).

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

RecyclerView adalah widget yang digunakan untuk menampilkan sebuah daftar yang berisi banyak item didalam list maupun grid. Secara bentuk dan fungsi widget ini hampir sama dengan ListView, tetapi secara konsep jauh berbeda. RecyclerView digunakan untuk menampilkan data yang berjumlah lebih banyak. Widget ini diperkenalkan pada Android version 5.0 (Lollipop).

Terdapat 3 komponen yang penting dalam menggunakan RecyclerView.

1.Adapter

Adapter pada RecyclerView secara konsep memiliki kesamaan dengan Adapter yang telah kita bahas sebelumnya. Keduanya menggunakan inflate item row layout untuk mengikat data ke RecyclerView. Tetapi adapter pada RecyclerView melakukan inherits dari “RecyclerView.Adapter” bukan BaseAdapter. Dengan menggunakan RecyclerView.Adapter kita dapat mengakses data dari sumber data dan mengisi setiap item pada konten. Ada beberapa overridable method dan property pada RecyclerView. Adapter yang harus diimplementasikan.

- onCreateViewHolder

Untuk membuat item row layout dan nilai return dari method ini adalah ViewHolder dari proses inflate layout yang digunakan pada method onBindViewHolder.

- onBindViewHolder: method ini berfungsi untuk mengambil data pada posisi tertentu dan diberikan pada ViewHolder.

- getItemCount

Serupa dengan getItemCount pada Adapter yang biasa kita gunakan, property ini untuk mendapatkan jumlah item pada sumber data.

Adapter ini juga bertanggung jawan terhadap event ItemClick, penjelasan lebih lanjut mengenai ItemClick akan dibahas pada latihan.

2.LayoutManager

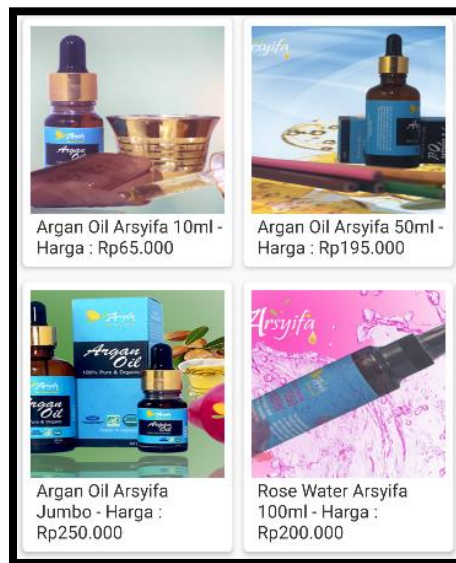
LayoutManager pada RecyclerView berfungsi untuk mengatur posisi item pada View. Pada LayoutManager kita dapat mengatur format dari tampilan pada item. Ada beberapa format pada LayoutManager.

- LinearLayoutManager

Pada format ini kita dapat mengatur orientasinya baik vertical maupun horizontal. Sesuai dengan namanya layout yang ditampilkan pada format ini berbentuk linear baik vertical maupun horizontal.

- GridLayoutManager

Seperti namanya layout ini berbentuk format grid dalam menampilkan item pada View. Kita bisa mengatur jumlah kolom yang akan ditampilkan pada screen. Pada layout ini juga mendukung untuk menampilkan item baik secara vertical maupun horizontal.



3.ViewHolder

Class ViewHolder melakukan inherits pada class “RecyclerView.ViewHolder”. Pada class ini dimana kita akan melakukan method FindViewById<T>, untuk mencari widget pada Layout. Setiap item pada RecyclerView memiliki hubungan dengan ViewHolder. Properti pada ViewHolder ini yang digunakan oleh adapter untuk mengisi data sesuai dengan posisinya.

Setelah dijelaskan bagaimana RecyclerView bekerja dan komponen-komponen yang mendukung untuk menggunakan widget ini, maka untuk memudahkan dari penjelasan diatas kita akan membuat sebuah latihan dengan menggunakan RecyclerView.

Buat sebuah layout dengan nama RecyclerViewLayout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:local="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:id="@+id/main_content">
    <android.support.v7.widget.RecyclerView
        android:id="@+id/rcView"
        android:scrollbars="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" />
</LinearLayout>
```

Pada layout diatas terdapat RecyclerView yang digunakan untuk menampilkan data-data pada screen.

*Note : Pastikan pada project sudah terdapat 3 buah nugget package seperti berikut **Android.Support.v7.AppCompat**, **Android.Support.v4** dan **Android.Support.v7.RecyclerView**.*

Pada latihan ini kita akan membuat aplikasi untuk menampilkan gambar-gambar sebuah produk kecantikan.

- Buat sebuah class dengan nama Product yang memiliki tiga buah property Id, Nama dan Harga.

```
namespace LatihanRecyclerView
{
    public class Product
    {
        public int Id { get; set; }
        public string Nama { get; set; }
        public double Harga { get; set; }

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

Property Id akan digunakan sebagai tempat menyimpan Id dari image (Resource.Drawable)

- Selanjutnya buat sebuah class baru dengan nama ProductRepository. Disini kita masih menggunakan pattern yang sama seperti pada latihan-latihan View sebelumnya. Pada

Class ini terdapat sebuah List<T> dan sebuah method untuk mengisi data pada list tersebut dan di letakkan didalam constructor.

```
namespace LatihanRecyclerView
{
    public class ProductRepository
    {
        public static List<Product> Products { get; }

        static ProductRepository()
        {
            Products = new List<Product>();
            TambahMakanan();
        }

        private static void TambahMakanan()
        {
            Products.Add(new Product()
            {
                Id = Resource.Drawable.product1,
                Nama = "Argan Oil Arsyifa 10ml",
                Harga = 65000
            });
        }
    }
}
```

Untuk detail sintaks dapat dilihat pada file latihan.

- Sedangkan untuk row Item layout kita masih menggunakan file yang sama yaitu CardViewLayout.xml
- Buat sebuah activity dengan nama “RecyclerViewLatihanActivity”.
- Lalu didalam file activity diatas, buat sebuah class ViewHolder, dengan nama ProductViewHolder

```
public class ProductViewHolder :
RecyclerView.ViewHolder
{
    public TextView ProductNameTextView { get; set; }
}

public ImageView ProductImageView { get; set; }

public ProductViewHolder(View itemView,
Action<int> listener) : base(itemView)
{
    ProductNameTextView =
itemView.FindViewById<TextView>(Resource.Id.lblTitle);
    ProductImageView =
itemView.FindViewById<ImageView>(Resource.Id.imgView);

    itemView.Click += (s, e) =>
listener(Position);
}
}
```

Pada class ini terdapat dua property dengan tipe masing-masing TextView dan ImageView sesuai dengan widget yang ada pada layout (CardViewLayout.xml). Untuk constructor pada class ini menggunakan reference dari CardView, dimana didalam constructor kita menggunakan method findViewById untuk mencari widget yang ada pada layout. Dimana property ini yang nantinya digunakan Adapter.

Yang perlu diketahui pada RecyclerView tidak ada event handler click seperti pada ListView. Jadi untuk membuat action pada saat item diklik kita harus membuatnya dari ViewHolder terlebih dahulu lalu Adapter, selanjutnya akan digunakan di activity level.

Pada parameter ItemView terdapat event handler click, dimana parameter ini merupakan reference dari CardView sehingga dapat digunakan pada saat user memilih sebuah item pada RecyclerView. Dengan menggunakan property position pada ViewHolder, kita dapat mengantarkan nilai tersebut kedalam method OnClick pada Adapter yang akan kita buat setelah ini.

- Selanjutnya kita akan membuat sebuah class adapter dengan nama ProductAdapter. Seperti yang telah dijelaskan diatas, adapter disini memiliki beberapa method yang penting. Seperti onCreateViewHolder()

```
public override RecyclerView.ViewHolder  
onCreateViewHolder(ViewGroup parent, int viewType)  
{  
    var layout =  
    LayoutInflater.From(parent.Context).Inflate(Resource.La  
yout.CardViewLayout, parent, false);  
  
    return new ProductViewHolder(layout, onItemClick);  
}
```

Pada saat RecyclerView memerlukan ViewHolder baru untuk menampilkan item, layout manager akan memanggil onCreateViewHolder(). Pada method ini kita menggunakan LayoutInflater untuk melakukan inisiasi layout (CardViewLayout) yang digunakan untuk menampilkan item. CardViewLayout merupakan representasi single row item pada RecyclerView. Jika kita lihat pada Return ViewHolder terdapat method onItemClick, sehingga kita harus membuat terlebih dahulu method tersebut.


```
public event EventHandler<int> ItemClick;
void OnItemClick(int position)
{
    ItemClick?.Invoke(this, position);
}
```

Selanjutnya adalah membuat method OnBindViewHolder().

```
public override void
OnBindViewHolder(RecyclerView.ViewHolder holder, int
position)
{
    var productViewHolder = (ProductViewHolder)holder;
    productViewHolder.ProductNameTextView.Text =
_products[position].Nama + " - Harga : " +
_products[position].Harga.ToString("C");
    productViewHolder.ProductImageView.SetImageResource(_pr
oducts[position].Id);
}
```

Pada OnBindViewHolder() ini menggunakan return ViewHolder dari method OnCreateViewHolder. Pada saat layout manager sudah ready untuk menampilkan view tertentu didalam RecyclerView, maka method OnBindViewHolder() akan dipanggil, dimana method ini akan mengisi data dari sumber data berdasarkan dari posisi yang didapat. Seperti pada latihan ini akan mengisi data dari Nama Product, harga dan Gambar yang menggunakan ImageResource Id.

```
public override int getItemCount => _products.Count;
```

Method getItemCount merupakan method yang return valuenya adalah jumlah item pada sumber data. Untuk sintaks lengkapnya dapat dilihat pada file latihan.

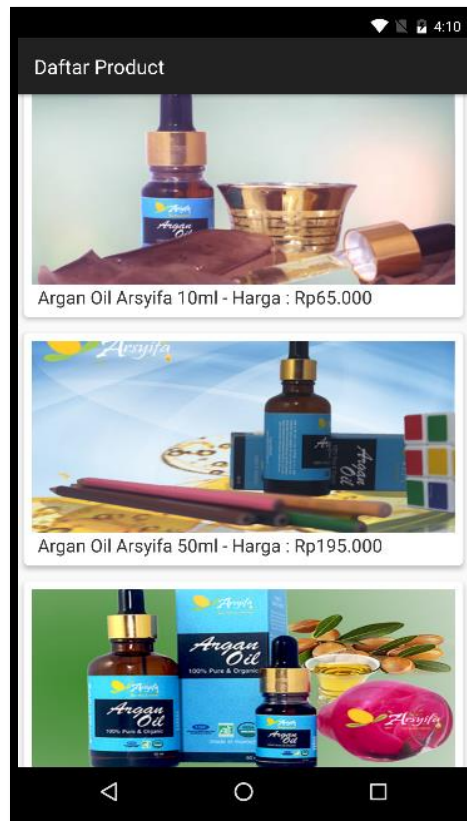
- Buka activity yang telah kita buat sebelumnya, selanjutnya pada method OnCreate(), ketikkan sintaks dibawah ini.

```
SetContentView(Resource.Layout.RecyclerViewLayout);
_recyclerView =
FindViewById<RecyclerView>(Resource.Id.rcView);
_layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.Vertical, false);
_recyclerView.SetLayoutManager(_layoutManager);
var productAdapter = new
ProductAdapter(ProductRepository.Products);
_recyclerView.SetAdapter(productAdapter);
productAdapter.ItemClick += MoviesAdapter_ItemClick;
```

Dapat dilihat pada sintaks diatas, kita menggunakan RecyclerViewLayout untuk activity ini. Menggunakan FindViewById untuk mencari widget RecyclerView pada layout dengan Id "rcView". Format yang digunakan untuk menampilkan item

menggunakan LayoutManager dengan format LinearLayoutManager. Lalu dengan menggunakan SetAdapter pada RecyclerView kita gunakan ProductAdapter. Dan terakhir untuk event click kita buat sebuah event handler seperti dibawah.

```
private void ProductAdapter_ItemClick(object sender, int e)
{
    Toast.MakeText(this, ProductRepository.Products[e].ToString(),
    ToastLength.Short).Show();
}
```



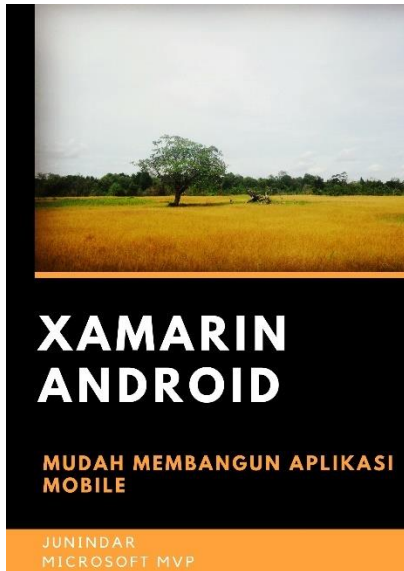
Jika kita perhatikan ItemClick menggunakan method pada ProductAdapter tidak seperti pada ListView yang menggunakan method yang telah disediakan pada widget itu sendiri. Jalankan program dan pastikan mendapatkan hasil seperti pada gambar diatas.

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/2019/03/xamarin-android-bekerja-dengan.html>

Referensi



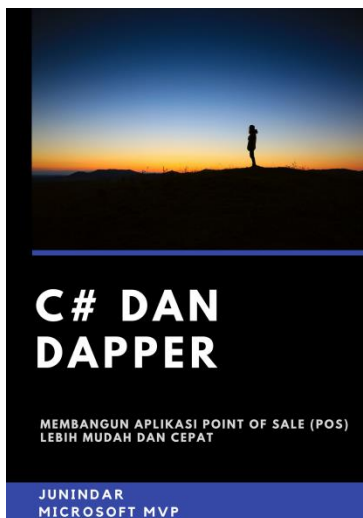
<https://play.google.com/store/books/details?id=G4tFDgAAQBAJ>



<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_C dan Dapper Membangun Aplikasi POS P
oint?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

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”**.