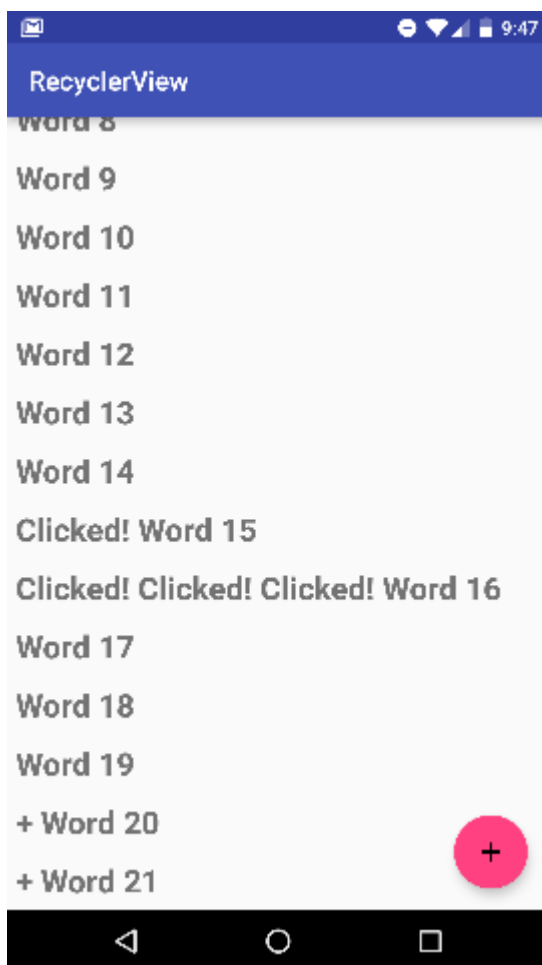


#Kemampuan Akhir Yang Direncanakan

- Mahasiswa mampu membuat aplikasi yang memiliki RecyclerView dengan berbagai macam datasource dan custom adapter.
- Mahasiswa mampu memodifikasi RecyclerView yang dibuat sebelumnya dengan memanfaatkan layout manager.

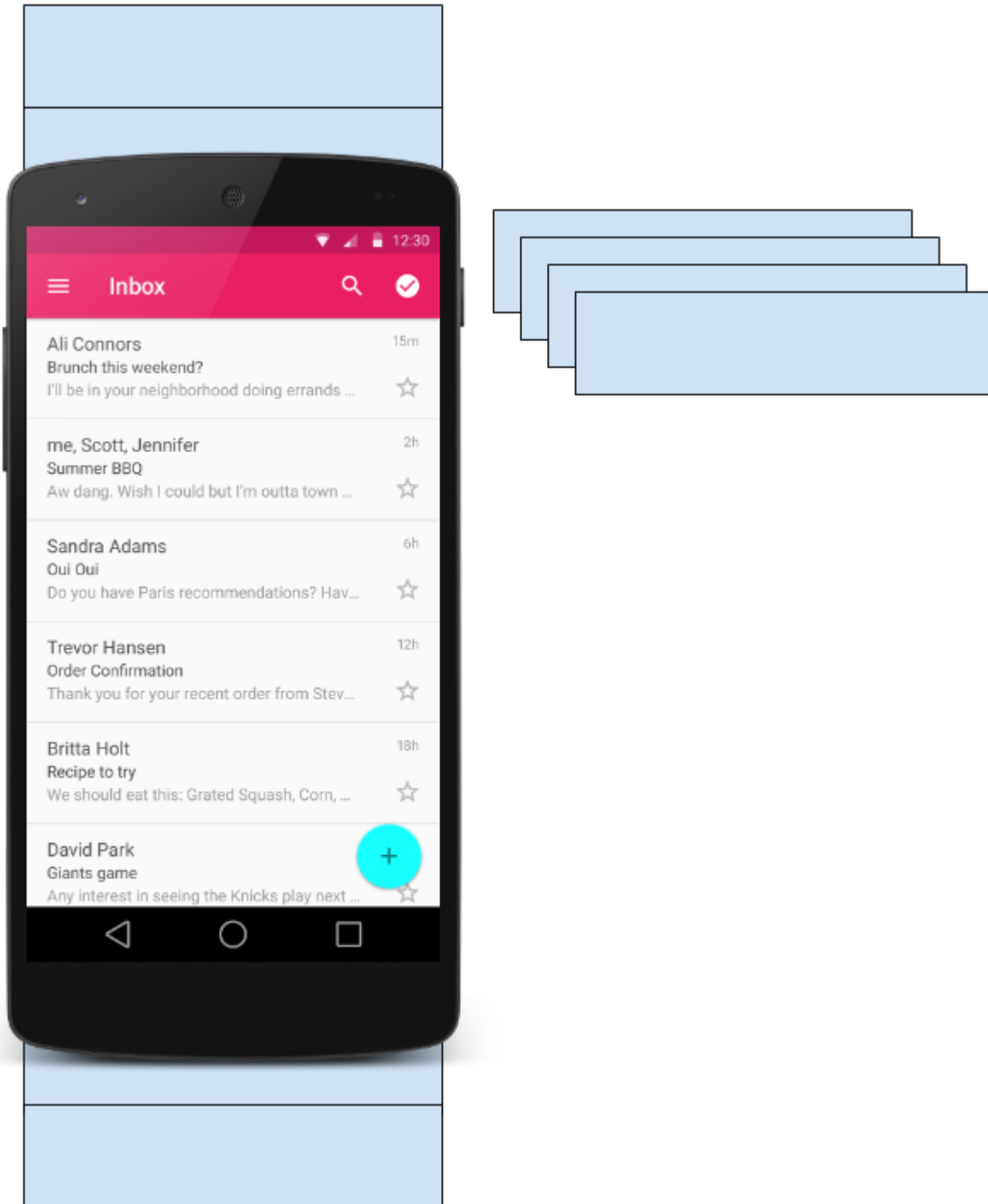
Modul

Saat Anda menampilkan banyak item dalam daftar yang bisa digulir, sebagian besar item tidak terlihat. Misalnya, dalam daftar kata yang panjang atau banyak judul berita, pengguna hanya melihat sedikit item daftar untuk setiap kalinya.



Atau, Anda bisa memiliki kumpulan data yang akan berubah saat pengguna berinteraksi dengannya. Jika Anda membuat tampilan baru setiap kali data berubah, itu juga membuat banyak tampilan, bahkan untuk kumpulan data yang kecil.

Dari perspektif kinerja, Anda bisa meminimalkan jumlah tampilan yang disimpan pada titik tertentu (Memori), dan jumlah tampilan yang harus Anda buat (Waktu). Kedua tujuan ini bisa dicapai dengan membuat agak lebih banyak tampilan daripada yang bisa dilihat pengguna pada layar, dan buat cache serta gunakan kembali tampilan yang dibuat sebelumnya dengan data berbeda saat pengguna menggulir ke dalam dan ke luar tampilan.



Kelas RecyclerView adalah versi ListView yang lebih canggih dan fleksibel. Widget ini adalah kontainer untuk menampilkan rangkaian data besar yang bisa digulir secara sangat efisien dengan mempertahankan tampilan dalam jumlah terbatas.

Gunakan widget RecyclerView bila Anda perlu menampilkan banyak data yang bisa digulir, atau kumpulan data dengan elemen yang berubah pada waktu proses berdasarkan aksi pengguna atau kejadian jaringan.

A. Komponen RecyclerView

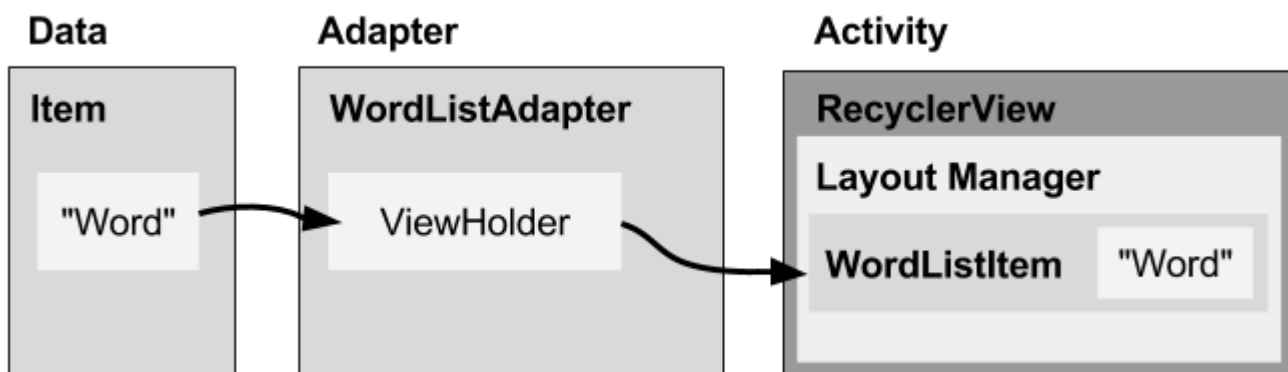
Untuk menampilkan data dalam RecyclerView, Anda memerlukan bagian berikut:

- **Data.** Tidak penting dari mana asal data. Anda bisa membuat data secara lokal, seperti yang Anda lakukan dalam latihan, mendapatkannya dari database perangkat seperti yang akan Anda lakukan

dalam praktik nanti, atau menariknya dari awan.

- **RecyclerView**. Daftar gulir yang berisi item daftar.
- **Layout untuk satu item data**. Semua item daftar tampak sama, sehingga Anda bisa menggunakan layout yang sama untuk semuanya. Layout item harus dibuat secara terpisah dari layout aktivitas, sehingga satu per satu tampilan item bisa dibuat dan diisi data.
- **Pengelola layout** Pengelola layout menangani penyusunan (layout) komponen antarmuka pengguna dalam suatu tampilan. Semua grup tampilan memiliki pengelola layout. Untuk LinearLayout, sistem Android menangani layout untuk Anda. RecyclerView memerlukan pengelola layout eksplisit untuk mengelola susunan item daftar yang terdapat di dalamnya. Layout ini bisa vertikal, horizontal, atau berupa petak. Pengelola layout adalah instance dari *RecyclerView.LayoutManager* untuk menyusun layout item dalam RecyclerView
- **Adapter**. Adapter menghubungkan data Anda dengan RecyclerView. Adapter menyiapkan data dan cara menampilkan data dalam view holder. Bila data berubah, adapter akan memperbarui materi tampilan item daftar terkait dalam RecyclerView. Adapter juga merupakan ekstensi dari *RecyclerView.Adapter*. Adapter menggunakan ViewHolder untuk menampung tampilan yang menyusun setiap item dalam RecyclerView, dan mengikat data untuk ditampilkan dalam tampilan yang menampilkannya.
- **View holder**. View holder memperluas kelas ViewHolder. View holder berisi tampilan informasi untuk menampilkan satu item dari layout item. View holder digunakan oleh adapter untuk menyediakan data, yang merupakan ekstensi dari *RecyclerView.ViewHolder*

Diagram di bawah ini menampilkan hubungan antara komponen-komponen ini.



B. Data

Semua data yang bisa ditampilkan akan ditampilkan dalam RecyclerView.

- Teks
- Gambar
- Ikon

Data bisa berasal dari sumber mana pun.

- Dibuat oleh aplikasi. Misalnya, kata acak untuk permainan.
- Dari database lokal. Misalnya, daftar kontak.
- Dari storage awan atau internet. Misalnya judul berita.

C. RecyclerView

RecyclerView adalah:

- Suatu grup Tampilan untuk kontainer yang bisa digulir
- Ideal untuk daftar item serupa yang panjang
- Hanya menggunakan tampilan dalam jumlah terbatas yang digunakan kembali saat tampilan tersebut tidak tampak di layar. Hal ini menghemat memori dan mempercepat pembaruan item daftar saat pengguna menggulir data, karena tidak perlu membuat tampilan baru untuk setiap item yang muncul.
- Secara umum, RecyclerView menyimpan sebanyak mungkin tampilan item yang muat di layar, plus sedikit tambahan pada setiap akhir daftar untuk memastikan pengguliran berjalan cepat dan lancar.

D. Layout Item

Layout adalah sebuah item daftar yang disimpan dalam file terpisah sehingga adapter bisa membuat tampilan item dan mengedit materinya secara independen dari layout aktivitas.

E. Pengelola Layout

Pengelola layout memosisikan tampilan item di dalam grup tampilan, seperti *RecyclerView* dan menentukan kapan harus menggunakan kembali tampilan item yang tidak lagi terlihat oleh pengguna. Untuk menggunakan kembali (atau mendaur ulang) tampilan, pengelola layout bisa meminta adapter untuk mengganti materi tampilan dengan elemen lain dari kumpulan data. Mendaur ulang tampilan dengan cara ini akan meningkatkan kinerja karena menghindari pembuatan tampilan yang tidak diperlukan atau melakukan pencarian *findViewById()* yang mahal. *RecyclerView* menyediakan semua pengelola layout bawaan ini:

- *LinearLayoutManager* menampilkan item dalam daftar gulir vertikal atau horizontal.
- *GridLayoutManager* menampilkan item dalam petak.
- *StaggeredGridLayoutManager* menampilkan item dalam petak zigzag Untuk membuat pengelola layout khusus, perluas kelas *RecyclerView.LayoutManager*.

F. Adapter

Adapter membantu dua antarmuka yang tidak kompatibel untuk bekerja bersama. Dalam *RecyclerView*, adapter menghubungkan data dengan tampilan. Adapter bertindak sebagai perantara antara data dan tampilan. Adapter menerima atau mengambil data, melakukan semua pekerjaan yang diperlukan agar bisa ditampilkan dalam suatu tampilan, dan menempatkan data dalam tampilan. Misalnya, adapter bisa menerima data dari database sebagai objek *Cursor*, mengekstrak kata dan definisinya, mengonversinya menjadi string, dan menempatkan string dalam suatu tampilan item yang memiliki tampilan teks, satu untuk kata dan satu untuk definisi. Anda akan mengetahui selengkapnya tentang *cursor* dalam bab berikutnya.

RecyclerView.Adapter mengimplementasikan sebuah view holder, dan harus mengganti callback berikut:

- *onCreateViewHolder()* memekarkan tampilan item dan mengembalikan view holder baru yang memuatnya. Metode ini dipanggil bila *RecyclerView* memerlukan view holder baru untuk menyatakan suatu item.

- `onBindViewHolder()` menyetel materi item pada posisi yang ditentukan dalam RecyclerView. Proses ini dipanggil oleh RecyclerView, misalnya, bila sebuah item baru bergulir ke tampilan.

G. View holder

`RecyclerView.ViewHolder` menjelaskan tampilan data dan metadata tentang tempatnya dalam RecyclerView. Setiap view holder menampung satu rangkaian data. Adapter menambahkan data ke view holder untuk ditampilkan oleh pengelola layout. Definisikan layout view holder Anda dalam file sumber daya XML. Layout ini bisa berisi (hampir) semua tipe tampilan, termasuk elemen yang bisa diklik.

H. Mengimplementasikan RecyclerView

Mengimplementasikan RecyclerView memerlukan langkah-langkah berikut:

1. Tambahkan dependensi RecyclerView ke file `app/build.gradle` aplikasi.
2. Tambahkan RecyclerView ke layout aktivitas
3. Buat file XML layout untuk satu item
4. Perluas `RecyclerView.Adapter` dan implementasikan metode `onCreateViewHolder` serta `onBindViewHolder`.
5. Perluas `RecyclerView.ViewHolder` untuk membuat view holder bagi layout item Anda. Anda bisa menambahkan perilaku klik dengan mengganti metode `onClick`.
6. Dalam aktivitas Anda, dalam metode `onCreate method`, buat RecyclerView dan inisialisasi dengan adapter serta pengelola layout.

1. Tambahkan dependensi ke `app/build.gradle`

Tambahkan pustaka RecyclerView ke file `app/build.gradle` Anda sebagai dependensi. Lihat bab mengenai pustaka dukungan atau praktik RecyclerView, jika Anda memerlukan petunjuk detail.

```
dependencies {  
    ...  
    compile 'com.android.support:recyclerview-v7:24.1.1'  
    ...  
}
```

2. Tambahkan RecyclerView ke layout aktivitas Anda

Tambahkan RecyclerView dalam file layout aktivitas Anda.

```
<android.support.v7.widget.RecyclerView  
    android:id="@+id/recyclerview"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
</android.support.v7.widget.RecyclerView>
```

Gunakan RecyclerView dari pustaka dukungan agar kompatibel dengan perangkat lama. Satu-satunya atribut yang diperlukan adalah id, beserta lebar dan tinggi. Sesuaikan item tersebut, bukan grup tampilan ini.

3. Buat layout untuk satu item

Buat file sumber daya XML dan tetapkan layout satu item. File ini akan digunakan oleh adapter untuk membuat view holder.

```
<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="6dp">

    <TextView
        android:id="@+id/word"
        style="@style/word_title" />

</LinearLayout>
```

Tampilan teks memiliki elemen `@style`. Gaya adalah kumpulan properti yang menetapkan sosok suatu tampilan. Anda bisa menggunakan gaya untuk berbagi atribut tampilan dengan beberapa tampilan. Sebuah cara mudah untuk membuat gaya adalah dengan mengekstrak gaya elemen UI yang sudah Anda buat. Misalnya, setelah menata gaya TextView, **Right-click > Refactor > Extract > Style** pada elemen tersebut dan ikuti konfirmasi dialog. Lebih detail mengenai gaya ada dalam praktik dan dalam bab berikutnya.

4. Buat adapter dengan view holder

Perluas *RecyclerView.Adapter* dan implementasikan metode *onCreateViewHolder* serta *onBindViewHolder*. Buat kelas Java baru dengan tanda tangan berikut:

```
public class WordListAdapter extends
RecyclerView.Adapter<WordListAdapter.WordViewHolder> {}
```

Dalam konstruktor, dapatkan inflater dari konteks saat ini, dan data Anda.

```
public WordListAdapter(Context context, LinkedList<String> wordList) {
    mInflater = LayoutInflater.from(context);
    this.mWordList = wordList;
}
```

Untuk adapter ini, Anda harus mengimplementasikan 3 metode.

- *onCreateViewHolder()* membuat suatu tampilan dan mengembalikannya.

```
@Override
public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType){
    // Inflate an item view.
    View itemView = inflater.inflate(R.layout.wordlist_item, parent,
false);
    return new ViewHolder(itemView, this);
}
```

- onBindViewHolder() menghubungkan data dengan view holder pada posisi yang ditentukan dalam RecyclerView.

```
@Override
public void onBindViewHolder(ViewHolder holder, int position) {
    // Retrieve the data for that position
    String mCurrent = mWordList.get(position);
    // Add the data to the view
    holder.wordItemView.setText(mCurrent);
}
```

getItemCount() kembali ke jumlah item data yang tersedia untuk ditampilkan.

```
@Override
public int getItemCount() {
    return mWordList.size();
}
```

5. Implementasikan kelas view holder

Perluas *RecyclerView.ViewHolder* untuk membuat view holder bagi layout item Anda. Anda bisa menambahkan perilaku klik dengan mengganti metode onClick. Kelas ini biasanya didefinisikan sebagai kelas dalam untuk adapter dan memperluas *RecyclerView.ViewHolder*.

```
class ViewHolder extends RecyclerView.ViewHolder {}
```

Jika Anda ingin menambahkan penanganan klik, Anda perlu mengimplementasikan listener klik. Salah satu cara untuk melakukannya adalah dengan memiliki view holder yang mengimplementasikan metode listener klik.

```
// Extend the signature of ViewHolder to implement a click listener.
class ViewHolder extends RecyclerView.ViewHolder implements
View.OnClickListener {}
```

Dalam konstruktornya, view holder harus memekarkan layoutnya, menghubungkannya dengan adapter, dan jika berlaku, menyetel listener klik.

```
public WordViewHolder(View itemView, WordListAdapter adapter) {  
    super(itemView);  
    wordItemView = (TextView) itemView.findViewById(R.id.word);  
    this.mAdapter = adapter;  
    itemView.setOnClickListener(this);  
}
```

Dan, jika Anda mengimplementasikan `onClickListener`, Anda juga harus mengimplementasikan `onClick()`.

```
@Override  
public void onClick(View v) {  
    wordItemView.setText ("Clicked! "+ wordItemView.getText());  
}
```

Perhatikan, untuk memasang listener klik ke elemen view holder lain, Anda harus melakukannya secara dinamis dalam `onBindViewHolder`. (Anda akan melakukan ini pada praktik nanti, saat Anda akan memperluas kode `RecyclerView` dari praktik.)

6. Buat RecyclerView

Terakhir, untuk mengikat semuanya, dalam metode `onCreate()` aktivitas Anda:

1. Dapatkan penanganan untuk `RecyclerView`.

```
mRecyclerView = (RecyclerView) findViewById(R.id.recyclerview);
```

2. Buat adapter dan berikan data untuk ditampilkan.

```
mAdapter = new WordListAdapter(this, mWordList);
```

3. Hubungkan adapter dengan `RecyclerView`.

```
mRecyclerView.setAdapter(mAdapter);
```

4. Buat `RecyclerView` dengan pengelola layout default.

```
mRecyclerView.setLayoutManager(new LinearLayoutManager(this));
```


RecyclerView adalah cara efisien untuk menampilkan data daftar gulir. RecyclerView menggunakan pola adapter untuk menghubungkan data dengan tampilan item daftar. Untuk mengimplementasikan RecyclerView, Anda perlu membuat adapter dan view holder, dan metode yang mengambil data serta menambahkannya ke item daftar.

Tugas

1. Membuat RecyclerView di Android

Contoh item layout yang akan dipakai adalah satu gambar di sebelah kiri dan dua baris TextView di sebelah kanan seperti gambar berikut :



Jika kalian sudah membuat sebuah project aplikasi Android baru di Android Studio, maka buka file build.gradle dan masukkan dependencies seperti berikut :

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 24
    buildToolsVersion '24.0.2'
    ...
}

dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    compile 'com.android.support:design:24.2.1'
    compile 'com.android.support:support-v4:24.2.1'
    compile 'com.android.support:appcompat-v7:24.2.1'
    compile 'com.android.support:palette-v7:24.2.1'
    compile 'com.android.support:cardview-v7:24.2.1'
    compile 'com.android.support:recyclerview-v7:24.2.1'
}
```

Langkah berikutnya adalah membuat layout untuk RecyclerView Adapter. Kita akan mendesain sehingga sebuah entry item di RecyclerView akan nampak seperti gambar di atas. Caranya buatlah sebuah file xml baru di res/layout, kemudian isikan kode berikut, file-nya bernama *view_rv_item.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="?android:attr/listPreferredItemHeight"
    android:padding="10dip" >
```

```

<ImageView
    android:id="@+id/icon"
    android:layout_width="wrap_content"
    android:layout_height="fill_parent"
    android:layout_alignParentBottom="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="6dip"
    android:contentDescription="TODO"
    android:src="@mipmap/ic_launcher" />

<TextView
    android:id="@+id/tv_subtitle"
    android:layout_width="fill_parent"
    android:layout_height="26dip"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_toRightOf="@id/icon"
    android:ellipsize="marquee"
    android:singleLine="true"
    android:textColor="@android:color/black"
    android:text="Description"
    android:textSize="12sp" />

<TextView
    android:id="@+id/tv_title"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_above="@id/tv_subtitle"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_alignWithParentIfMissing="true"
    android:layout_toRightOf="@id/icon"
    android:gravity="center_vertical"
    android:text="Example application"
    android:textColor="@android:color/black"
    android:textSize="16sp" />

</RelativeLayout>

```

Selanjutnya, kita juga akan membuat layout menambahkan elemen RecyclerView pada layout utama, biasanya bernama *activity_main.xml*, sehingga kodenya akan berubah seperti berikut :

```

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

    <android.support.v7.widget.RecyclerView
        android:id="@+id/rv_main"
        android:layout_width="match_parent"
        android:layout_height="match_parent"

```

```

        android:scrollbars="vertical" />

</LinearLayout>

```

Setelah itu, kita akan membuat sebuah Adapter untuk RecyclerView, fungsi Adapter adalah seperti yang sudah dijelaskan di atas. Caranya, buatlah sebuah file .java baru bernama *RecyclerViewAdapter.java* dan copy pastekan kode berikut :

```

import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import java.util.ArrayList;

/**
 * Created by Herdi_WORK on 15.09.16.
 */
public class RecyclerViewAdapter extends
RecyclerView.Adapter<RecyclerViewAdapter.ViewHolder> {

    private ArrayList<String> rvData;

    public RecyclerViewAdapter(ArrayList<String> inputData) {
        rvData = inputData;
    }

    public class ViewHolder extends RecyclerView.ViewHolder {

        // di tutorial ini kita hanya menggunakan data String untuk tiap
item
        public TextView tvTitle;
        public TextView tvSubtitle;

        public ViewHolder(View v) {
            super(v);
            tvTitle = (TextView) v.findViewById(R.id.tv_title);
            tvSubtitle = (TextView) v.findViewById(R.id.tv_subtitle);
        }
    }

    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        // membuat view baru
        View v =
LayoutInflater.from(parent.getContext()).inflate(R.layout.view_rv_item,
parent, false);
        // mengeset ukuran view, margin, padding, dan parameter layout
lainnya
        ViewHolder vh = new ViewHolder(v);
    }
}

```

```

        return vh;
    }

    @Override
    public void onBindViewHolder(ViewHolder holder, int position) {
        // - mengambil elemen dari dataset (ArrayList) pada posisi
        tertentu
        // - mengeset isi view dengan elemen dari dataset tersebut
        final String name = rvData.get(position);
        holder.tvTitle.setText(rvData.get(position));
        holder.tvSubtitle.setText("Frau " + position);
    }

    @Override
    public int getItemCount() {
        // menghitung ukuran dataset / jumlah data yang ditampilkan di
        RecyclerView
        return rvData.size();
    }
}

```

Yang terakhir, kita akan menambahkan kode pada MainActivity.java, yang akan mensuplai Adapter dengan inputan data yang diinginkan, copy pastekan kode berikut pada file MainActivity.java :

```

import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;

import java.util.ArrayList;

/**
 * Created by Herdi_WORK on 15.09.16.
 */
public class MainActivity extends AppCompatActivity {

    private RecyclerView rvView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;
    private ArrayList<String> dataSet;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_recyclerview);
        dataSet = new ArrayList<>();
        initDataset();

        rvView = (RecyclerView) findViewById(R.id.rv_main);
        rvView.setHasFixedSize(true);
    }
}

```

```
/**
 * Kita menggunakan LinearLayoutManager untuk list standar
 * yang hanya berisi daftar item
 * disusun dari atas ke bawah
 */
layoutManager = new LinearLayoutManager(this);
rvView.setLayoutManager(layoutManager);

adapter = new RecyclerViewAdapter(dataSet);
rvView.setAdapter(adapter);

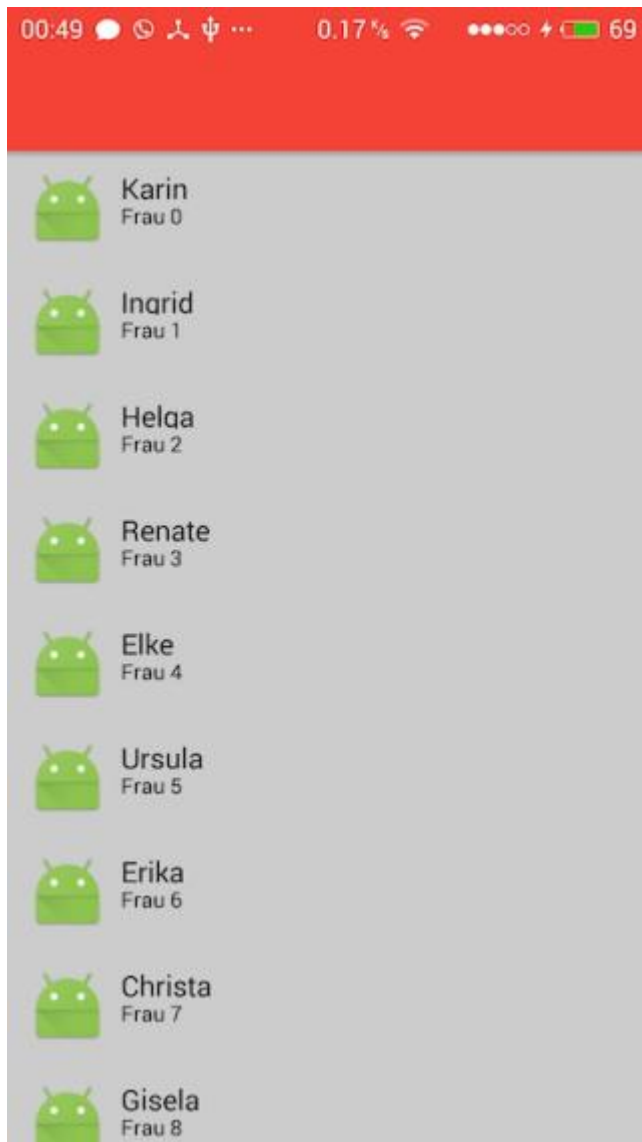
}

private void initDataset(){

    /**
     * Tambahkan item ke dataset
     * dalam prakteknya bisa bermacam2
     * tidak hanya String seperti di kasus ini
     */
    dataSet.add("Karin");
    dataSet.add("Ingrid");
    dataSet.add("Helga");
    dataSet.add("Renate");
    dataSet.add("Elke");
    dataSet.add("Ursula");
    dataSet.add("Erika");
    dataSet.add("Christa");
    dataSet.add("Gisela");
    dataSet.add("Monika");

}
}
```

Jika sudah kita akan langsung running aplikasinya dan akan muncul tampilan RecyclerView seperti di bawah ini.



2. Desain list item yang lebih bagus dan rapi menggunakan CardView, maka hasilnya akan menjadi seperti di bawah ini :

