

# MODUL 7

## (MULTIMEDIA dan KAMERA)

### DESKRIPSI TEMA

- Multimedia
- Kamera

### CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Mahasiswa mampu membuat App Android yang dapat menampilkan dan memutar video, serta mengintegrasikan penggunaan kamera untuk mengambil gambar dan video.

### PENUNJANG PRAKTIKUM

1. Software Android Studio

### LANGKAH-LANGKAH PRAKTIKUM

#### Tutorial: Menggunakan Kamera

App yang akan dibuat pada tutorial ini adalah sebuah app yang dapat menggunakan kamera untuk meng-capture foto maupun video. Berikut langkah-langkah tutorial ini:

1. Buat Android Studio Project yang baru dengan nama: **Weeko7A\_[NIM]**, domain: **umn.ac.id**, minimum Android: **API 21: Android 5.0 (lollipop)**, Activity Template: **Empty Activity**, Activity Name: **MainActivity**, dan Layout Name: **activity\_main**.
2. pemanfaatan kamera, dengan cara mengubah file **AndroidManifest.xml** pada folder **app > manifest** dengan menambahkan skrip XML berikut :

```
<uses-feature android:name="android.hardware.camera"
            android:required="true" />
```

3. Selanjutnya buka dan layout file **activity\_main.xml** dengan mengganti obyek **textView** bawaan dengan menyiapkan **ImageView** dan **VideoView** untuk menampilkan hasil foto dan/atau video serta dua buah tombol untuk mengaktifkan kamera dengan memanggil app bawaan android yang menangani kamera sehingga file **activity\_main.xml** seperti terlihat pada skrip XML berikut:

```

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

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="200dp" />

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Ambil Foto" />

    <VideoView
        android:id="@+id/videoView"
        android:layout_width="match_parent"
        android:layout_height="200dp" />

    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Ambil Video" />
</LinearLayout>

```

4. Tambahkan penanganan kamera pada Java code pada **MainActivity.java** dengan menghubungkan ImageView, VideoView, dan dua tombol tersebut dengan membuat objek-objek dan memanggil findViewById seperti kode berikut:

```

public class MainActivity extends AppCompatActivity {
    private Button foto;
    private Button video;
    private ImageView kotakFoto;
    private VideoView kotakVideo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        foto = findViewById(R.id.button);
        video = findViewById(R.id.button2);
        kotakFoto = findViewById(R.id.imageView);
        kotakVideo = findViewById(R.id.videoView);
    }
}

```

5. Pemanggilan app bawaan Kamera dilakukan dengan penggunaan Intent Implicit **MediaStore.ACTION\_IMAGE\_CAPTURE** untuk ambil foto dan **MediaStore.ACTION\_VIDEO\_CAPTURE** untuk ambil video. Dalam hal ini MainActivity sebagai pemanggil Intent ini harus dapat menerima kembali hasil foto dan video yang dihasilkan oleh Intent tersebut, maka digunakan method **startActivityForResult(...)**. Berikut adalah penambahan kode untuk MainActivity dalam menangani hal ini:

Variable konstant

```
static final int REQUEST_IMAGE_CAPTURE = 1;
static final int REQUEST_VIDEO_CAPTURE = 2;
```

Menghubungkan VideoView dengan MediaController yang akan memutar atau memainkan hasil video, serta penanganan jika tombol Foto dan tombol Video diklik yang diletakkan pada method **onCreate(..)**:

```
MediaController controller = new MediaController(this);
controller.setMediaPlayer(kotakVideo);
kotakVideo.setMediaController(controller);
foto.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent takePictureIntent = new
            Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        if (takePictureIntent.resolveActivity(getPackageManager())
            != null) {
            startActivityForResult(takePictureIntent,
                REQUEST_IMAGE_CAPTURE);
        }
    }
});

video.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent takeVideoIntent = new
            Intent(MediaStore.ACTION_VIDEO_CAPTURE);
        if (takeVideoIntent.resolveActivity(getPackageManager())
            != null) {
            startActivityForResult(takeVideoIntent,
                REQUEST_VIDEO_CAPTURE);
        }
    }
});
```

6. Selanjutnya adalah mengimplementasi **onActivityResult(..)** method callback untuk menerima hasil dari Foto maupun Video, berikut kode dari method ini:

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
        Bundle extras = data.getExtras();
        Bitmap imageBitmap = (Bitmap) extras.get("data");
        kotakFoto.setImageBitmap(imageBitmap);
    } else if (requestCode == REQUEST_VIDEO_CAPTURE &&
        resultCode == RESULT_OK) {
        Uri videoUri = data.getData();
        kotakVideo.setVideoURI(videoUri);
        kotakVideo.seekTo(100);
        kotakVideo.start();
    }
}

```

7. Jalankan (run) project anda dengan menggunakan Smartphone sebagai target devicenya agar kamera dapat digunakan langsung dan perhatikan apa yang terjadi jika anda klik atau sentuh layar pada tombol Foto dan tombol Video pada app tersebut.

## Tutorial: Galeri Video

App yang akan dibuat pada tutorial ini adalah sebuah app yang akan menampilkan daftar Video yang disimpan pada **raw** folder. Daftar Video yang ada akan ditampilkan berupa **RecyclerView** dengan Custom layout. Berikut langkah-langkah tutorial ini:

1. Buat Android Studio Project yang baru dengan nama: **Weeko7B\_[NIM]**, domain: **umn.ac.id**, minimum Android: **API 21: Android 5.0 (lollipop)**, Activity Template: **Empty Activity**, Activity Name: **MainActivity**, dan Layout Name: **activity\_main**.
2. Buat sebuah class bantu yang dapat menjadi perantara pada layout sebuah Video dengan keterangannya. Beri nama class tersebut **SumberVideo** pada folder **app > java > id.ac.umn.weeko7b\_[NIM]**. Class ini memiliki member variable Judul, Keterangan, dan Uri dari Video. Lengkapi dengan Konstruktor, setter method, getter method dan **toString** method. Agar object class ini dapat dikirim antar Activity maka class ini harus **implements Serializable**. Berikut java program untuk class ini:

```

import android.net.Uri;
public class SumberVideo implements Serializable {
    private String judul;
    private String keterangan;
    private String videoURI;
    public SumberVideo(String judul, String keterangan,
                        String videoURI){
        this.judul = judul;
        this.keterangan = keterangan;
        this.videoURI = videoURI;
    }

    public String getJudul()          { return this.judul;          }
    public String getKeterangan()      { return this.keterangan;    }
    public String getVideoURI()        { return this.videoURI;      }

    public void setJudul(String judul){ this.judul = judul;      }
    public void setKeterangan(String keterangan) {
        this.keterangan = keterangan;
    }
    public void setVideoURI(String videoURI) {
        this.videoURI = videoURI;
    }
    public String toString() { return this.getJudul() + " => "
                               + this.getKeterangan(); }
}

```

3. Buat Layout untuk menampilkan masing-masing item pada daftar Video yang selanjutnya ditampilkan pada RecyclerView. Beri nama layout ini dengan nama **video\_item\_layout.xml** pada folder **app > res > layout**. Layout ini terdiri dari **VideoView** (dengan ukuran 100 x 100 dp) , dua **TextView** untuk Judul dan keterangan dari Video tersebut. Root Layout menggunakan **LinearLayout Horizontal**. Berikut skrip xml untuk layout ini:

```

<?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="100dp">
    <VideoView
        android:id="@+id/kotakVideo"
        android:layout_width="100dp"
        android:layout_height="100dp" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:orientation="vertical">
        <TextView
            android:id="@+id/tvJudul"
            android:layout_width="match_parent"
            android:layout_height="25dp"
            android:text="Judul"
            android:textAppearance=
                "@style/TextAppearance.AppCompat.Medium" />

```

```

        <TextView
            android:id="@+id/tvKeterangan"
            android:layout_width="match_parent"
            android:layout_height="55dp"
            android:text="keterangan" />
        <TextView
            android:id="@+id/tvUri"
            android:layout_width="match_parent"
            android:layout_height="20dp"
            android:text="Uri" />
    </LinearLayout>
</LinearLayout>

```

4. Berikutnya adalah memasukkan **RecyclerView** object pada layout **activity\_main.xml** yang menggantikan **TextView** object bawaan dengan cara drag-and-drop pada mode Design atau tulis skrip xml pada mode Text sebagai berikut:

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

5. Selanjutnya menghubungkan layout ini dengan **MainActivity.java** seperti pada tutorial sebelumnya. Berikut java code untuk keperluan ini:

```

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.RecyclerView;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    RecyclerView rvDaftarVideo;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        rvDaftarVideo = (RecyclerView) findViewById(R.id.recyclerView);
    }
}

```

6. Untuk dapat mengimplementasikan **RecyclerView** diperlukan sebuah **Adapter** yang dapat menghubungkan antara **View** pada masing-masing Item Video dan class **SumberVideo** serta **LayoutManager**. Buatlah sebuah class pada folder **app > java > id.ac.umn.weeko7b\_[NIM]** dengan nama **DaftarVideoAdapter**. Class ini merupakan sub-class dari **RecyclerView.Adapter<T>**. Dimana **T** adalah class yang akan menangani tampilan View pada masing-masing item di dalam daftar Video yang ada pada RecyclerView. Class **T** ini merupakan sub-class dari **RecyclerView.ViewHolder**. Agar mempermudah sharing member antar Adapter dengan ViewHolder sebaiknya kita buat class **ViewHolder** ini dengan nama **ItemVideoViewHolder** sebagai inner class dari Adapter. Class Adapter ini membutuhkan member variable: sebuah **LinkedList<SumberVideo>** yang menyimpan daftar Video yang diisi oleh MainActivity, **LayoutInflater** agar dapat menampilkan layout yang telah dirancang untuk masing-masing item, **Context** yang merupakan Activity yang memiliki object RecyclerView. Member variable ini harus diinisialisasi dengan menyediakan Konstruktor. Method **onCreateViewHolder(...)** yang menampilkan view berdasarkan layout file yang dipakai. Method **onBindViewHolder(...)** akan dijalankan untuk menghubungkan object-object yang ada pada layout dengan data yang ada pada daftar Video dengan mengisi content pada object layout tersebut. Method **getItemCount()** akan memberikan jumlah element pada daftar video. Berikut kode dari class outer **DatarVideoAdapter**:

```
public class DaftarVideoAdapter extends
    RecyclerView.Adapter<DaftarVideoAdapter.ItemVideoViewHolder> {
    private LinkedList<SumberVideo> mDaftarVideo;
    private LayoutInflater mInflater;
    private Context mContext;

    public DaftarVideoAdapter(Context context,
                               LinkedList<SumberVideo> daftarVideo) {
        this.mContext = context;
        this.mDaftarVideo = daftarVideo;
        this.mInflater = LayoutInflater.from(context);
    }

    @NonNull
    @Override
    public ItemVideoViewHolder onCreateViewHolder(
        @NonNull ViewGroup parent, int viewType) {
        View view = mInflater.inflate(R.layout.video_item_layout ,
                                     parent , false);
        return new ItemVideoViewHolder(view, this);
    }
}
```

```

@Override
public void onBindViewHolder(@NonNull ItemVideoViewHolder holder,
    int position) {
    SumberVideo mSumberVideo = mDaftarVideo.get(position);
    holder.tvJudul.setText(mSumberVideo.getJudul());
    holder.tvKeterangan.setText(mSumberVideo.getKeterangan());
    holder.tvUri.setText(mSumberVideo.getVideoURI());
    holder.kotakVideo.setVideoURI(
        Uri.parse(mSumberVideo.getVideoURI()));
    holder.kotakVideo.seekTo(100);
}
@Override
public int getItemCount() {
    return mDaftarVideo.size();
}

```

7. Inner class **ItemVideoViewHolder** agar dapat terintegrasi dengan outer class **DaftarVideoAdapter**, dan object-object pada layout file **video\_item\_layout.xml**, dengan membuat member variable untuk Adapter dan untuk object-object tersebut. Setelah itu tambah Konstruktor yang dapat menginisialisasi member variable di atas. Selain itu kita perlu untuk menambah penanganan jika user klik pada view tersebut yang berupa interface **OnClickListener** dengan meng-implement pada Class header dan implement method **onClick(...)**. Pada method **onClick(...)** akan membuka Activity lain yang akan menampilkan detail dari item Video yang kita pilih, misalkan dengan nama **DetailVideoActivity** serta mengirimkan data berupa object **SumberVideo** yang di-klik. Berikut Java code dari class **ItemVideoViewHolder** yang berada di dalam class outernya:

```

class ItemVideoViewHolder extends RecyclerView.ViewHolder
    implements View.OnClickListener {
    private VideoView kotakVideo;
    private TextView tvJudul;
    private TextView tvKeterangan;
    private TextView tvUri;
    private DaftarVideoAdapter mAdapter;
    private int mPosisi;
    private SumberVideo mSumberVideo;
    public ItemVideoViewHolder(@NonNull View itemView,
        DaftarVideoAdapter adapter) {
        super(itemView);
        mAdapter = adapter;
        kotakVideo = (VideoView) itemView.findViewById(R.id.kotakVideo);
        tvJudul = (TextView) itemView.findViewById(R.id.tvJudul);
        tvKeterangan = (TextView) itemView.findViewById(
            R.id.tvKeterangan);
        tvUri = (TextView) itemView.findViewById(R.id.tvUri);
        itemView.setOnClickListener(this);
    }
}

```



```

@Override
public void onClick(View v) {
    mPosisi = getLayoutPosition();
    mSumberVideo = mDaftarVideo.get(mPosisi);
    Intent detailInten = new Intent(mContext,
                                   DetilVideoActivity.class);
    Bundle bundle = new Bundle();
    bundle.putSerializable("DetilVideo", mSumberVideo);
    detailInten.putExtras(bundle);
    mContext.startActivity(detailInten);
}
}

```

8. Berikutnya adalah menyiapkan layout dan juga kode java untuk dapat menerima object **SumberVideo** yang dikirim oleh class **ItemVideoViewHolder** pada saat user klik salah satu item dari daftar Video pada **recyclerView**. Layout yang disiapkan terdiri dari sebuah **VideoView**, beberapa **TextView** untuk label, **EditText** untuk judul, keterangan dan video Uri, serta sebuah tombol untuk kembali ke pemanggil (**MainActivity**). Berikut adalah skrip xml untuk membuat layout tersebut dengan nama **activity\_detil\_video.xml**:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".DetilVideoActivity">
    <VideoView
        android:id="@+id/vvDetil"
        android:layout_width="match_parent"
        android:layout_height="374dp" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <TextView
            android:id="@+id/textView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Judul" />
        <EditText
            android:id="@+id/etJudul"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:inputType="text" />
    </LinearLayout>

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="96dp"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/textView2"
        android:layout_width="31dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Keterangan" />
    <EditText
        android:id="@+id/etKeterangan"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:ems="10"
        android:inputType="textMultiLine" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="48dp"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/textView3"
        android:layout_width="29dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Uri" />
    <EditText
        android:id="@+id/etUri"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:ems="10"
        android:inputType="text" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">
    <Button
        android:id="@+id/btnBack"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Kembali" />
</LinearLayout>
</LinearLayout>

```

9. Langkah selanjutnya adalah melengkapi code yang ada pada Activity **DetilVideoActivity.java** yang menerima kiriman object **SumberVideo** sesuai dengan pilihan user, kiriman onject ini selanjutnya

didistribusikan ke object-object **EditText** dan **VideoView** yang sebelumnya dideklarasikan dan dihubungkan dengan object-object pada layout file di atas. Berikut kode Java untuk memenuhi keperluan di atas:

```
public class DetilVideoActivity extends AppCompatActivity {
    private VideoView vvDetil;
    private EditText etJudul;
    private EditText etKeterangan;
    private EditText etUri;
    private Button btnKembali;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detil_video);
        vvDetil = (VideoView) findViewById(R.id.vvDetil);
        etJudul = (EditText) findViewById(R.id.etJudul);
        etKeterangan = (EditText) findViewById(R.id.etKeterangan);
        etUri = (EditText) findViewById(R.id.etUri);
        btnKembali = (Button) findViewById(R.id.btnBack);

        Intent intent = getIntent();
        Bundle bundle = intent.getExtras();
        SumberVideo sv = (SumberVideo) bundle.getSerializable(
            "DetilVideo");
        etJudul.setText(sv.getJudul());
        etKeterangan.setText(sv.getKeterangan());
        etUri.setText(sv.getVideoURI());
        vvDetil.setVideoURI(Uri.parse(sv.getVideoURI()));
        vvDetil.seekTo(100);
        vvDetil.start();
        MediaController controller = new MediaController(this);
        controller.setMediaPlayer(vvDetil);
        vvDetil.setMediaController(controller);
        btnKembali.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });
    }
}
```

**Keterangan:** method **seekTo(...)** digunakan untuk membuat Thumbnail dari sebuah **VideoView** yang menampilkan frame (image) pada posisi video di millisecond sesuai parameter yang diberikan. Pada class ini juga dipadukan dengan class **MediaController** yang berfungsi untuk menangani playback dari isi video pada **VideoView**. Saat user klik pada obyek **VideoView** maka jendela atau dialog **MediaController** akan ditampilkan sehingga user dapat mengendalikan video tersebut seperti play, pause, jump forward dan jump backward.

10. Mengisi dan menyiapkan daftar video serta menambahkannya ke dalam `LinkedList<SumberVideo>` dilakukan pada `MainActivity.java`. Begitu juga menghubungkan `RecyclerView` object dengan `Adapter` object dan mengkonfigurasi Layout Manager dilakukan di class ini. Berikut kode lengkap dari `MainActivity.java`:

```
public class MainActivity extends AppCompatActivity {
    RecyclerView rvDaftarVideo;
    DaftarVideoAdapter mAdapter;
    LinkedList<SumberVideo> daftarVideo = new LinkedList<>();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        isiDaftarVideo();
        rvDaftarVideo = (RecyclerView) findViewById(R.id.recyclerView);
        mAdapter = new DaftarVideoAdapter(this, daftarVideo);
        rvDaftarVideo.setAdapter(mAdapter);
        rvDaftarVideo.setLayoutManager(new LinearLayoutManager(this));
    }

    public void isiDaftarVideo() {
        daftarVideo.add(new SumberVideo("Mini Cooper Drag",
            "Drag Race Mini Cooper dengan Honda Civic hatchback",
            "android.resource://" + getPackageName() + "/" +
            R.raw.mini01));
        daftarVideo.add(new SumberVideo("Porsche 918",
            "Menikmati mobil sport Porsche 918 Spyder",
            "android.resource://" + getPackageName() + "/" +
            R.raw.porsche918));
        daftarVideo.add(new SumberVideo("Drift H2H",
            "Head to Head Drifting hasil cuplikan dari film The
            Fast and The Furious",
            "android.resource://" + getPackageName() + "/" +
            R.raw.drift01));
        daftarVideo.add(new SumberVideo("Kejar-kejaran",
            "Kejar-kejaran di jalanan hasil cuplikan dari film The
            Fast and The Furious",
            "android.resource://" + getPackageName() + "/" +
            R.raw.drift02));
        daftarVideo.add(new SumberVideo("Mini Cooper Drag Race",
            "Drag Race Mini Cooper dengan Ford Fiesta hatchback",
            "android.resource://" + getPackageName() + "/" +
            R.raw.mini02));
    }
}
```

**Keterangan:** Pada tutorial ini file-file video yang digunakan disimpan atau diletakkan di dalam folder `app > res > raw`. File-file video ini dapat diunduh pada tautan Google Drive : <https://drive.google.com/open?id=13WTolDbRf1kWTFolZc7sGz-1knNGa1YH> [Gunakan akun email / google akun domain `umn.ac.id` karena adanya kebijakan pembatasan di domainnya]. Setelah unduh file `video.zip`,

ekstrak file tersebut yang menghasilkan beberapa file video .mp4. Copy paste file-file tersebut ke dalam folder **app > res > raw**. Sementara App ini belum dapat menambahkan dan mengedit daftar video yang sudah disiapkan sebelumnya secara langsung pada saat App dijalankan. Penambahan hanya dapat dilakukan secara manual dengan menambahkan file-file .mp4 ke dalam folder **app > res > raw** dengan ketentuan nama file harus menggunakan semua huruf kecil (**lower case**) dan/atau angka, lalu tambahkan kode penambahan **daftarVideo.add(...)** pada method **isiDaftarVideo()** mengikuti contoh yang sudah ada.

11. Yakinkan tidak adanya error pada semua file yang dibuat. Build project dan jalankan (run) project dengan target emulator dan/atau smartphone yang terhubung melalui kabel USB. Lakukan pengujian terhadap App tersebut dengan klik pada salah satu item di RecyclerView dan setelah jendela Detil Video muncul klik pada Video tersebut dan klik tombol kembali. Agar RecyclerView terlihat menarik coba tambahkan beberapa video seperti pada keterangan di atas.

### **Latihan, Tugas, dan Eksperimen:**

1. Ikuti langkah-langkah di atas sampai ke project berjalan dengan baik. Zip (kompres) file-file yang anda buat dan atau yang anda lakukan perubahan dan submit ke isian tugas pada eLearning:
2. Lakukan pengujian terhadap app anda dengan mengklik tombol Foto dan Video (pada Tutorial Kamera). Serta untuk project tutorial Galeri Video, klik item pada RecyclerView pada Main Activity, Video, tombol Kembali dan EditText pada DetilVideo Activity . Beri catatan dan kesimpulan dari hasil pengujian app anda dan submit ke eLearning portal.