

Tutorial 6

Android Store & Retrieve Data

1. Internal Data Storage

Edit layout **activity_main.xml** sehingga memiliki baris program sebagai berikut:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >
    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:layout_marginRight="20dp"
        android:layout_marginTop="24dp"
        android:ems="10" >
        <requestFocus />
    </EditText>
    <EditText
        android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignRight="@+id/editText1"
        android:layout_below="@+id/editText1"
        android:layout_marginTop="24dp"
        android:ems="10" />
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/editText1"
        android:layout_alignBottom="@+id/editText1"
        android:layout_alignParentLeft="true"
        android:text="File Name:" />
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/editText2"
        android:layout_alignBottom="@+id/editText2"
        android:layout_alignParentLeft="true"
        android:text="Data:" />
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/editText2"
        android:layout_below="@+id/editText2"
        android:layout_marginLeft="50dp"
        android:layout_marginTop="16dp"
        android:text="save" />
    <Button
        android:id="@+id/button2"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/button1"
        android:layout_alignBottom="@+id/button1"
        android:layout_toRightOf="@+id/button1"
        android:text="read" />
</RelativeLayout>
```

Selanjutnya, pada **MainActivity.java**, tambahkan baris program berikut:

```
package com.example.tutorial6;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;

public class MainActivity extends AppCompatActivity {
    EditText editTextFileName, editTextData;
    Button saveButton, readButton;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        editTextFileName=findViewById(R.id.editText1);
        editTextData=findViewById(R.id.editText2);
        saveButton=findViewById(R.id.button1);
        readButton=findViewById(R.id.button2);

        saveButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                /* Melakukan inisialisasi nilai variabel filename
                dan data ketika user menyentuh tombol */
                String filename=editTextFileName.getText().toString();
                String data=editTextData.getText().toString();
                // Membuat variabel bernama fos bertipe FileOutputStream
                FileOutputStream fos;
                try {
                    // Melakukan Inisialisasi nilai fos yang merupakan variabel objek
                    fos = openFileOutput(filename, Context.MODE_PRIVATE);
                    //default mode is PRIVATE, can be APPEND etc.
                    // Menuliskan data ke Internal Storage
                    fos.write(data.getBytes());
                    fos.close();

                    Toast.makeText(getApplicationContext(),filename + " saved",
                        Toast.LENGTH_LONG).show();

                } catch (FileNotFoundException e) {e.printStackTrace();}
                catch (IOException e) {e.printStackTrace();}
            }
        });
    }
}
```

```
});

// Ketika tombol Read disentuh
readButton.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View arg0) {
        String filename=editTextFileName.getText().toString();
        StringBuffer stringBuffer = new StringBuffer();
        try {
            /* Menambahkan BufferedReader untuk FileInputString dengan
            bantuan InputStreamReader untuk membaca nilai */
            InputStreamReader(
                openFileInput(filename));
            String inputString;
            // Membaca data per-baris dan mengirimkannya ke stringBuffer
            while ((inputString = inputReader.readLine()) != null) {
                stringBuffer.append(inputString + "\n");
            }

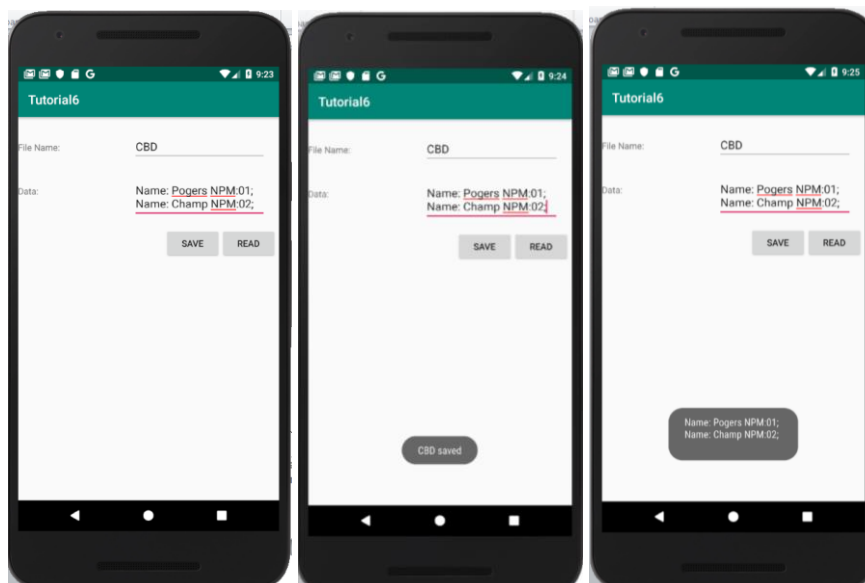
        } catch (IOException e) {
            e.printStackTrace();
        }

        Toast.makeText(getApplicationContext(), stringBuffer.toString(), Toast.LENGTH_LONG).show();

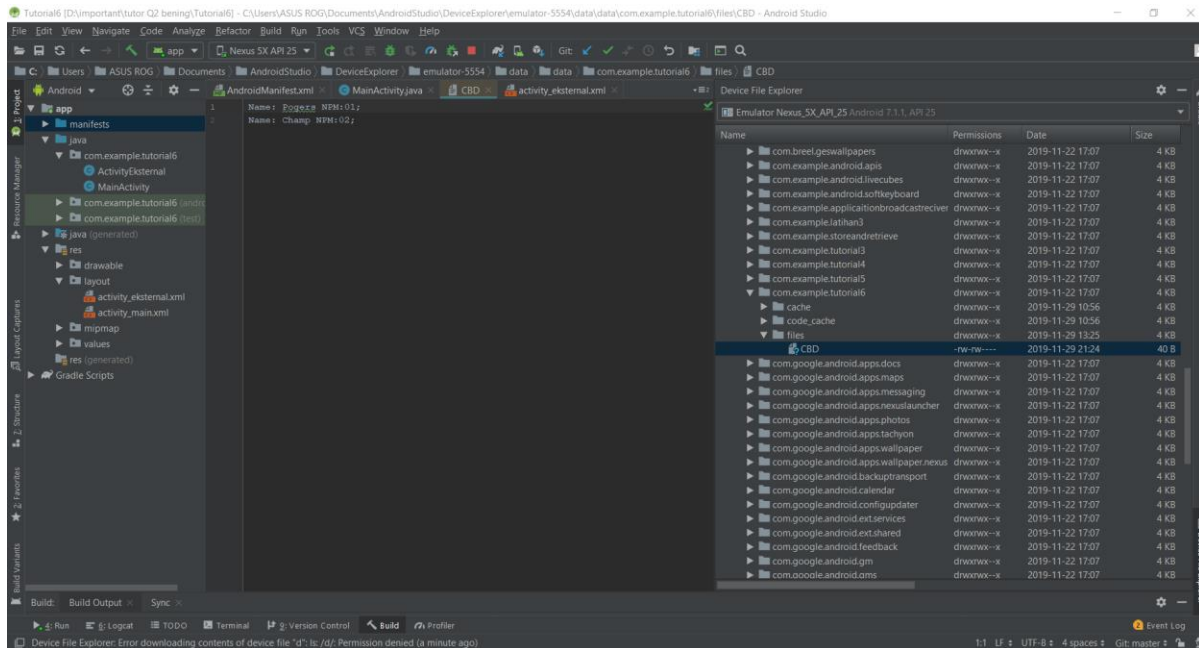
    }

});
}
```

Jalankan program!



Untuk mengecek data tersimpan di Internal Storage, silahkan buka **Device File Explorer**, pada layar IDE terdapat di sebelah kanan bawah, atau bisa dengan menekan tombol keyboard **Ctrl + Shift + A**, lalu ketikkan **Device Explorer**. Setelahnya buka **data/data/<nama_package>/file**.



2. Eksternal Data Storage

Eksternal Storage sama seperti SD card, bisa digunakan untuk menyimpan data tanpa adanya penentuan keamanan terhadap file-file yang disimpan didalamnya.

Secara umum terdapat dua tipe Eksternal Storage, yakni:

a. Primary External Storage:

Merupakan penyimpanan bawaan yang bisa diakses bersama-sama. Storage tipe ini bisa diakses oleh user dengan menancapkan sebuah kabel USB dan menjadikannya sebagai drive di komputer host. Contoh: Nexus 5 32 GB.

b. Secondary External Storage:

Merupakan storage yang dapat dipindah-pindah seperti SD card. Semua aplikasi bisa membaca dan menulis file yang diletakkan di Eksternal Storage dan user bisa menghapusnya. Kita perlu melakukan pengecekan terhadap availability SD card dan apakah kita bisa menulis file didalamnya. Saat Eksternal Storage available, barulah kita bisa menuliskan file ke dalamnya, selain dari itu tombol penyimpanan akan disabled.

Pertama jangan lupa untuk meminta izin melalui file manifest untuk melakukan Read dan Write ke Eksternal Storage.

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

Berikutnya buatlah sebuah layout baru dengan nama **activity_eksternal.xml**, isikan baris kodingan berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    android:orientation="vertical">
    <TextView android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Reading and Writing to External Storage"
        android:textSize="24sp"/>
    <EditText android:id="@+id/myInputText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10" android:lines="5"
        android:minLines="3" android:gravity="top|left"
        android:inputType="textMultiLine">
        <requestFocus />
    </EditText>
    <LinearLayout
        android:layout_width="match_parent" android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:weightSum="1.0"
        android:layout_marginTop="20dp">
        <Button android:id="@+id/saveExternalStorage"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="SAVE"
            android:layout_weight="0.5"/>
        <Button android:id="@+id/getExternalStorage"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="0.5"
            android:text="READ" />
    </LinearLayout>
    <TextView android:id="@+id/response"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:padding="5dp"
        android:text=""
        android:textAppearance="?android:attr/textAppearanceMedium" />
</LinearLayout>
```

Selanjutnya buat sebuah activity baru dengan nama **ActivityEksternal.java**, lalu isikan baris kodingan berikut:

```
public class ActivityEksternal extends AppCompatActivity {
    EditText inputText;
    TextView response;
    Button saveButton, readButton;

    // Membuat nama file
    private String filename = "SampleFile.txt";
    // Membuat nama folder penyimpanan
```

```
private String filepath = "MyFileStorage";
// Membuat objek kelas File
File myExternalFile;
// Menginisialisasi nilai awal variabel myData
String myData = "";

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_eksternal);

    inputText = (EditText) findViewById(R.id.myInputText);
    response = (TextView) findViewById(R.id.response);

    saveButton = (Button) findViewById(R.id.saveExternalStorage);
    saveButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            try {
// Membuat objek FileOutputStream untuk proses penyimpanan output file
                FileOutputStream fos = new FileOutputStream(myExternalFile);
// Menuliskan isi file berdasarkan inputan user
                fos.write(inputText.getText().toString().getBytes());
// Menutup proses tulis
                fos.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
// Mengatur nilai kosong jika user menyentuh tombol tanpa mengisi text
            inputText.setText("");
// Memberikan respon setelah user menyentuh tombol simpan
            response.setText("SampleFile.txt saved to External Storage...");
        }
    });

    readButton = (Button) findViewById(R.id.getExternalStorage);
    readButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            try {
                FileInputStream fis = new FileInputStream(myExternalFile);
// Membuat objek DataInputStream untuk mengambil objek file
                DataInputStream in = new DataInputStream(fis);
// Membuat objek BufferedReader untuk membaca isi file
                BufferedReader br = new BufferedReader(new
InputStreamReader(in));
// Membuat variabel penampung nilai text di file
                String strLine;
// Melakukan pengecekan text file kosong atau tidak
                while ((strLine = br.readLine()) != null) {
                    /* Jika file tidak kosong, variabel myData sebelumnya yang
kosong akan
                    diisi dengan nilai text di strLine */
                    myData = myData + strLine;
                }
                in.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
            inputText.setText(myData);
            response.setText("SampleFile.txt data retrieved from Internal
Storage...");
        }
    });
// Mengecek status Eksternal Storage apakah available
// Jika tidak maka tombol simpan akan disable
}
```

```

        if (!isExternalStorageAvailable() || isExternalStorageReadOnly()) {
            saveButton.setEnabled(false);
        }
        else {
            myExternalFile = new File(getExternalFilesDir(filepath), filename);
        }
    }
    // Proses pengecekan Eksternal Storage read only atau tidak
    private static boolean isExternalStorageReadOnly() {
        String extStorageState = Environment.getExternalStorageState();
        if (Environment.MEDIA_MOUNTED_READ_ONLY.equals(extStorageState)) {
            return true;
        }
        return false;
    }

    // Proses pengecekan Eksternal Storage available atau tidak
    private static boolean isExternalStorageAvailable() {
        String extStorageState = Environment.getExternalStorageState();
        if (Environment.MEDIA_MOUNTED.equals(extStorageState)) {
            return true;
        }
        return false;
    }
}

```

fungsi **getExternalStorageState()** merupakan fungsi static untuk menentukan apakah eksternal storage available atau tidak. Fungsi **Environment.getExternalStorageState()** mengembalikan path ke internal SD.

Fungsi **getExternalFilesDir()** akan mengembalikan path ke file folder didalam path **Android/data/data/<nama_package>/** didalam SD card yang digunakan untuk menyimpan semua file seperti images hasil download, saat aplikasi di uninstall, semua file dalam storage ini juga akan dihapus. Jika Eksternal Storage tidak available, tombol **Save** akan di disablekan.

