FILKOM | UB

CCE60220

Perangkat Bergerak (TKOM)

Fakultas Ilmu Komputer Universitas Brawijaya

MATAKULIAH	: **Perangkat Bergerak (TKOM)**
KODE/ STATUS	: CCE60220
SKS	: 2
Dosen	: Dahnial Syauqy, S.T, M.T
Email	: dahnial87@ub.ac.id
Ruang	:

Agenda Perkuliahan

1. Intro dan overview perkuliahan
2. Sejarah dan perkembangan teknologi perangkat bergerak
3. Komponen perangkat keras dan perangkat lunak
4. Pengenalan dan instalasi android studio serta aplikasi sederhana
5. Intent dan passing data pada Android Studio
6. Android Studio: Sensor reading
7. **Android Studio: Storage & shared preference**
8. **==========UTS**
9. Pengenalan dan aplikasi sederhana dengan MIT AppInventor
10. Appinventor: variable, looping, conditional, tinyDB, file
11. appInventor: sensor reading & persiapan project
12. Appinventor: Akuisisi gambar dan suara
13. Appinventor: komunikasi bluetooth
14. Appinventor: basic animation
15. Presentasi kelompok
16. **==========UAS**

Saving Data to Storage

- Shared Preference
- Internal Storage
- External Storage
- Using Database (not discussed)

Shared Preference

Shared Preferences allows activities and applications to keep preferences, in the form of **key-value pairs** that will persist even when the user closes the application.

SharedPreferences is application specific, i.e. the data is lost on performing one of the following options:

- on uninstalling the application
- on clearing the application data (through Settings)

As the name suggests, the primary purpose is to store user-specified configuration details, such as user specific settings, keeping the user logged into the application.

Android stores Shared Preferences settings as XML file in **shared_prefs** folder under DATA/data/{application package} directory.

Initialization

```
SharedPreferences pref = getApplicationContext().getSharedPreferences("MyPref", 0);
Editor editor = pref.edit();
```

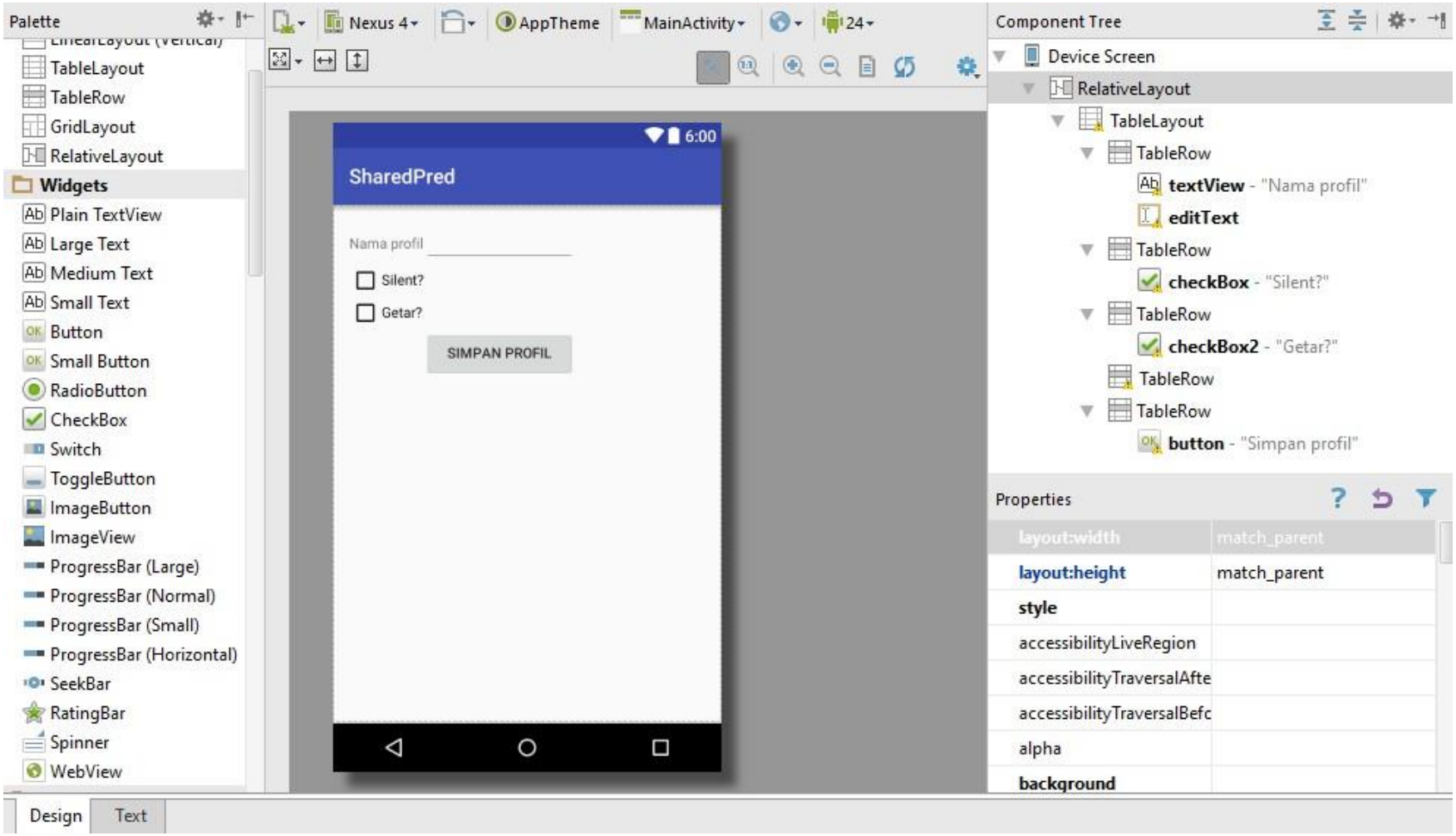
Storing Data

editor.commit() is used in order to save changes to shared preferences.

```
editor.putBoolean("key_name", true); // Storing boolean - true/false
editor.putString("key_name", "string value"); // Storing string
editor.putInt("key_name", "int value"); // Storing integer
editor.putFloat("key_name", "float value"); // Storing float
editor.putLong("key_name", "long value"); // Storing long

editor.commit(); // commit changes
```

Retrieving Data

```
pref.getString("key_name", null); // getting String
pref.getInt("key_name", null); // getting Integer
pref.getFloat("key_name", null); // getting Float
pref.getLong("key_name", null); // getting Long
pref.getBoolean("key_name", null); // getting boolean
```

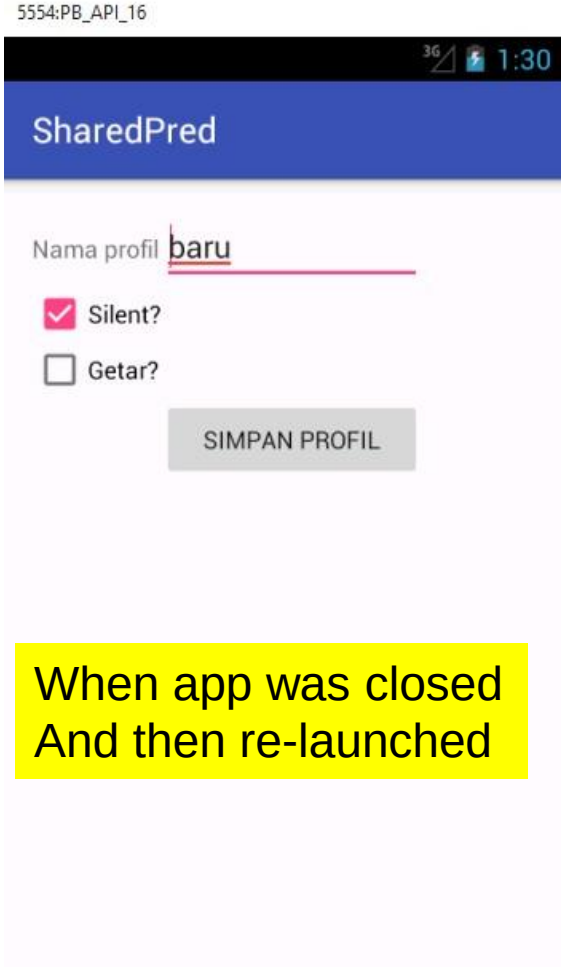

Palette
TableLayout
TableRow
GridLayout
RelativeLayout
Widgets
Plain TextView
Large Text
Medium Text
Small Text
Button
Small Button
RadioButton
CheckBox
Switch
ToggleButton
ImageButton
ImageView
ProgressBar (Large)
ProgressBar (Normal)
ProgressBar (Small)
ProgressBar (Horizontal)
SeekBar
RatingBar
Spinner
WebView
Nexus 4
AppTheme
MainActivity
24
6:00
SharedPred
Nama profil
Silent?
Getar?
SIMPAN PROFIL
Component Tree
Device Screen
RelativeLayout
TableLayout
TableRow
textView - "Nama profil"
editText
TableRow
checkBox - "Silent?"
TableRow
checkBox2 - "Getar?"
TableRow
TableRow
button - "Simpan profil"
Properties
layout:width
match_parent
layout:height
match_parent
style
accessibilityLiveRegion
accessibilityTraversalAfte
accessibilityTraversalBefc
alpha
background
Design
Text

```
package com.tekom.home.sharedpred;

import ...

public class MainActivity extends AppCompatActivity {
    private EditText myedittext;
    private CheckBox mycheckbox1;
    private CheckBox mycheckbox2;
    private Button mybutton;

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

        myedittext = (EditText)findViewById(R.id.editText);
        mycheckbox1 = (CheckBox)findViewById(R.id.checkBox);
        mycheckbox2 = (CheckBox)findViewById(R.id.checkBox2);
        mybutton = (Button)findViewById(R.id.button);

        mybutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

            }
        });

    }
}
```

```
package com.tekom.home.sharedpred;

import ...

public class MainActivity extends AppCompatActivity {
    private EditText myedittext;
    private CheckBox mycheckbox1;
    private CheckBox mycheckbox2;
    private Button mybutton;

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

        myedittext = (EditText)findViewById(R.id.editText);
        mycheckbox1 = (CheckBox)findViewById(R.id.checkBox);
        mycheckbox2 = (CheckBox)findViewById(R.id.checkBox2);
        mybutton = (Button)findViewById(R.id.button);

        SharedPreferences settings = getSharedPreferences("MyPREFERENCES", Context.MODE_PRIVATE);

        mybutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

            }
        });

    }
}
```

```
    private CheckBox mycheckbox1;
    private CheckBox mycheckbox2;
    private Button mybutton;

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

        myedittext = (EditText)findViewById(R.id.editText);
        mycheckbox1 = (CheckBox)findViewById(R.id.checkBox);
        mycheckbox2 = (CheckBox)findViewById(R.id.checkBox2);
        mybutton = (Button)findViewById(R.id.button);

        final SharedPreferences settings = getSharedPreferences("MyPREFERENCES", Context.MODE_PRIVATE);

        mybutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                SharedPreferences.Editor editor = settings.edit();

                editor.putString("namaprofil",myedittext.getText().toString());
                editor.putBoolean("kondisisilent",mycheckbox1.isChecked());
                editor.putBoolean("kondisigetar",mycheckbox2.isChecked());

                editor.commit();
                Toast.makeText(MainActivity.this, "Setting telah disimpan", Toast.LENGTH_SHORT).show();
            }
        });

    }
}
```

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

    myedittext = (EditText)findViewById(R.id.editText);
    mycheckbox1 = (CheckBox)findViewById(R.id.checkBox);
    mycheckbox2 = (CheckBox)findViewById(R.id.checkBox2);
    mybutton = (Button)findViewById(R.id.button);

    final SharedPreferences settings = getSharedPreferences("MyPREFERENCES", Context.MODE_PRIVATE);

    // Restore preferences
    myedittext.setText(settings.getString("namaprofil",""));
    mycheckbox1.setChecked(settings.getBoolean("kondisisilent",false));
    mycheckbox2.setChecked(settings.getBoolean("kondisigetar",false));

    mybutton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            SharedPreferences.Editor editor = settings.edit();

            editor.putString("namaprofil",myedittext.getText().toString());
            editor.putBoolean("kondisisilent",mycheckbox1.isChecked());
            editor.putBoolean("kondisigetar",mycheckbox2.isChecked());

            editor.commit();
            Toast.makeText(MainActivity.this, "Setting telah disimpan", Toast.LENGTH_SHORT).show();
        }
    });

}
}
```



When app was closed
And then re-launched

Using Internal Storage

Android Internal storage is the storage of the private data on the device memory (but not relational data or some sort of key/value cache pairs).

By default, saving and loading files to the internal storage are private to the application and other applications will not have access to these files. When the user uninstalls the applications the internal stored files associated with the application are also removed.

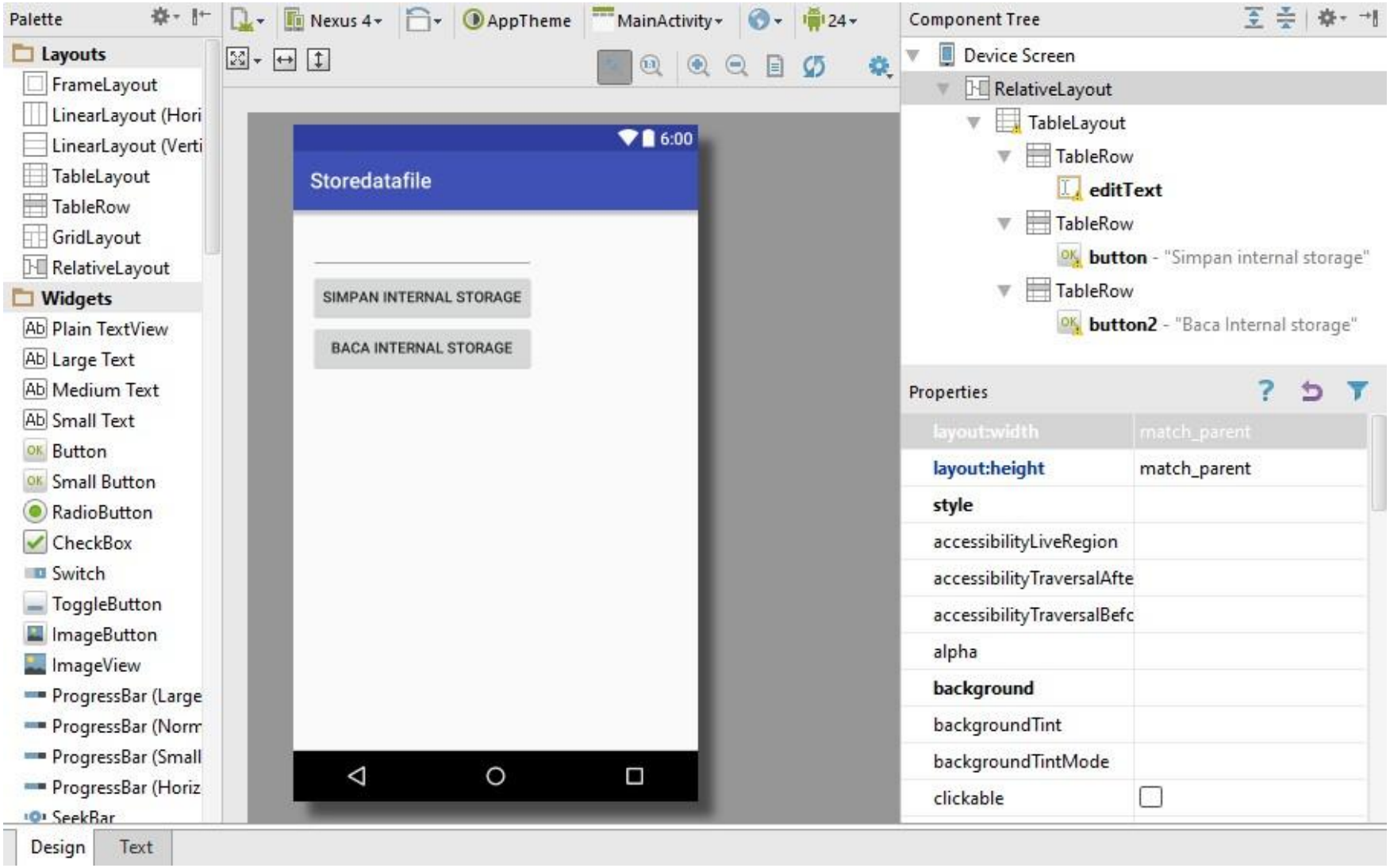
openFileOutput(): This method is used to create and save a file. It's syntax is given below:

```
FileOutputStream fOut = openFileOutput("file name",Context.MODE_PRIVATE);
```

```
String str = "test data";
fOut.write(str.getBytes());
fOut.close();
```

openFileInput(): This method is used to open a file and read it.

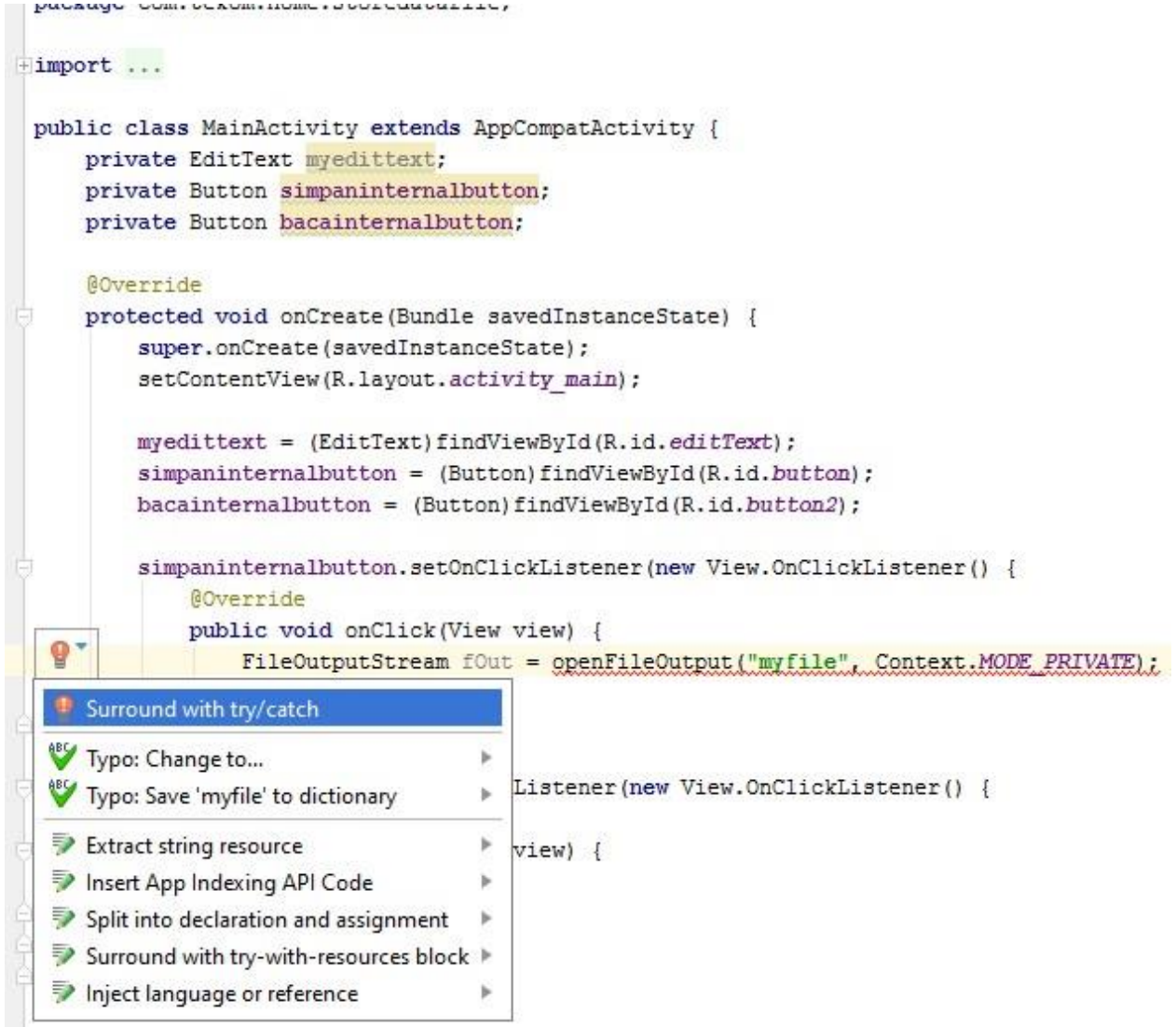
```
FileInputStream fin = openFileInput(file);
```


Palette
Layouts
FrameLayout
LinearLayout (Hori
LinearLayout (Verti
TableLayout
TableRow
GridLayout
RelativeLayout
Widgets
Plain TextView
Large Text
Medium Text
Small Text
Button
Small Button
RadioButton
CheckBox
Switch
ToggleButton
ImageButton
ImageView
ProgressBar (Large
ProgressBar (Norm
ProgressBar (Small
ProgressBar (Horiz
SeekBar
Nexus 4
AppTheme
MainActivity
24
6:00
Storedatafile
SIMPAN INTERNAL STORAGE
BACA INTERNAL STORAGE
Component Tree
Device Screen
RelativeLayout
TableLayout
TableRow
editText
TableRow
button - "Simpan internal storage"
TableRow
button2 - "Baca Internal storage"
Properties
layout:width
match_parent
layout:height
match_parent
style
accessibilityLiveRegion
accessibilityTraversalAfte
accessibilityTraversalBefc
alpha
background
backgroundTint
backgroundTintMode
clickable
Design
Text

```
import ...

public class MainActivity extends AppCompatActivity {
    private EditText myedittext;
    private Button simpaninternalbutton;
    private Button bacainternalbutton;

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

        myedittext = (EditText)findViewById(R.id.editText);
        simpaninternalbutton = (Button)findViewById(R.id.button);
        bacainternalbutton = (Button)findViewById(R.id.button2);

        simpaninternalbutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

            }
        });

        bacainternalbutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

            }
        });
    }
}
```

```
import ...

public class MainActivity extends AppCompatActivity {
    private EditText myedittext;
    private Button simpaninternalbutton;
    private Button bacainternalbutton;

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

        myedittext = (EditText)findViewById(R.id.editText);
        simpaninternalbutton = (Button)findViewById(R.id.button);
        bacainternalbutton = (Button)findViewById(R.id.button2);

        simpaninternalbutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                FileOutputStream fOut = openFileOutput("myfile", Context.MODE_PRIVATE);
            }
        });

        bacainternalbutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

            }
        });
    }
}
```

File not found exception??


import ...
public class MainActivity extends AppCompatActivity {
private EditText myedittext;
private Button simpaninternalbutton;
private Button bacainternalbutton;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
myedittext = (EditText)findViewById(R.id.editText);
simpaninternalbutton = (Button)findViewById(R.id.button);
bacainternalbutton = (Button)findViewById(R.id.button2);
simpaninternalbutton.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
FileOutputStream fOut = openFileOutput("myfile", Context.MODE_PRIVATE);
Surround with try/catch
Typo: Change to...
Typo: Save 'myfile' to dictionary
Extract string resource
Insert App Indexing API Code
Split into declaration and assignment
Surround with try-with-resources block
Inject language or reference
Listener(new View.OnClickListener() {
view) {

```
import ...

public class MainActivity extends AppCompatActivity {
    private EditText myedittext;
    private Button simpaninternalbutton;
    private Button bacainternalbutton;

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

        myedittext = (EditText)findViewById(R.id.editText);
        simpaninternalbutton = (Button)findViewById(R.id.button);
        bacainternalbutton = (Button)findViewById(R.id.button2);

        simpaninternalbutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                try {
                    FileOutputStream fOut = openFileOutput("myfile", Context.MODE_PRIVATE);
                } catch (FileNotFoundException e) {
                    e.printStackTrace();
                }
            }
        });

        bacainternalbutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

            }
        });
```

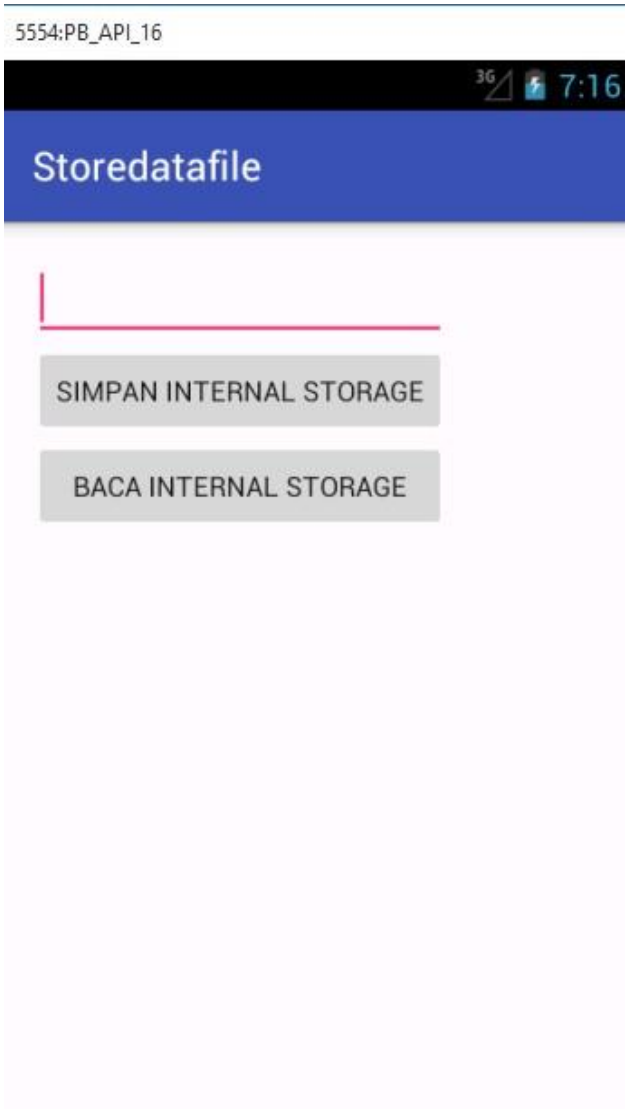
```
public class MainActivity extends AppCompatActivity {
    private EditText myedittext;
    private Button simpaninternalbutton;
    private Button bacainternalbutton;

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

        myedittext = (EditText)findViewById(R.id.editText);
        simpaninternalbutton = (Button)findViewById(R.id.button);
        bacainternalbutton = (Button)findViewById(R.id.button2);

        simpaninternalbutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                try {
                    FileOutputStream fOut = openFileOutput("myfile", Context.MODE_PRIVATE);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });

        bacainternalbutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

            }
        });
    }
}
```

```
    private Button simpaninternalbutton;
    private Button bacainternalbutton;

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

        myedittext = (EditText)findViewById(R.id.editText);
        simpaninternalbutton = (Button)findViewById(R.id.button);
        bacainternalbutton = (Button)findViewById(R.id.button2);

        simpaninternalbutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                try {
                    FileOutputStream fOut = openFileOutput("myfile", Context.MODE_PRIVATE);
                    fOut.write(myedittext.getText().toString().getBytes());
                    fOut.close();
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });

        bacainternalbutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

            }
        });
    }
}
```

```
private Button simpaninternalbutton;
private Button bacainternalbutton;

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

    myedittext = (EditText)findViewById(R.id.editText);
    simpaninternalbutton = (Button)findViewById(R.id.button);
    bacainternalbutton = (Button)findViewById(R.id.button2);

    simpaninternalbutton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            try {
                FileOutputStream fOut = openFileOutput("myfile", Context.MODE_PRIVATE);
                fOut.write(myedittext.getText().toString().getBytes());
                fOut.close();
                Toast.makeText(MainActivity.this, "data telah disimpan", Toast.LENGTH_SHORT).show();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });

    bacainternalbutton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

        }
    });
}
}
```

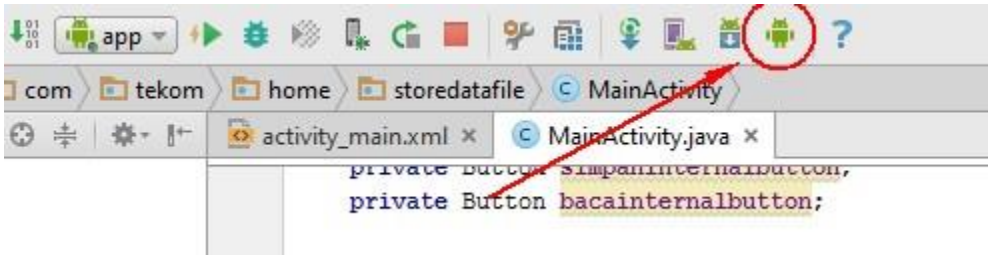
```
                } catch (Exception e) {
                    e.printStackTrace();
                }
            });

            bacainternalbutton.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    //reading text from file
                    try {
                        FileInputStream fileIn=openFileInput("myfile");
                        InputStreamReader InputRead= new InputStreamReader(fileIn);

                        char[] inputBuffer= new char[100];
                        String s="";
                        int charRead;

                        while ((charRead=InputRead.read(inputBuffer))>0) {
                            // char to string conversion
                            String readstring=String.copyValueOf(inputBuffer,0,charRead);
                            s +=readstring;
                        }
                        InputRead.close();
                        myedittext.setText(s);


                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                }
            });
        }
    }
```

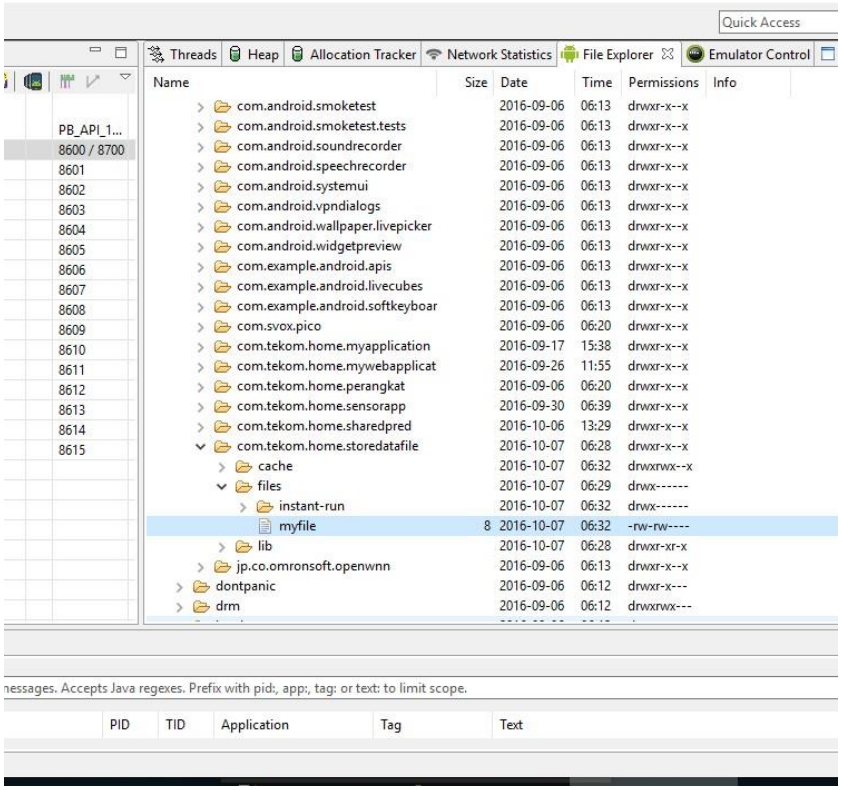
Result


5554:PB_API_16
7:16
Storedatafile
SIMPAN INTERNAL STORAGE
BACA INTERNAL STORAGE


5554:PB_API_16
7:16
Storedatafile
coba ini
SIMPAN INTERNAL STORAGE
BACA INTERNAL STORAGE

Where the file is located?





Using external storage

External storage such as SD card can also store application data

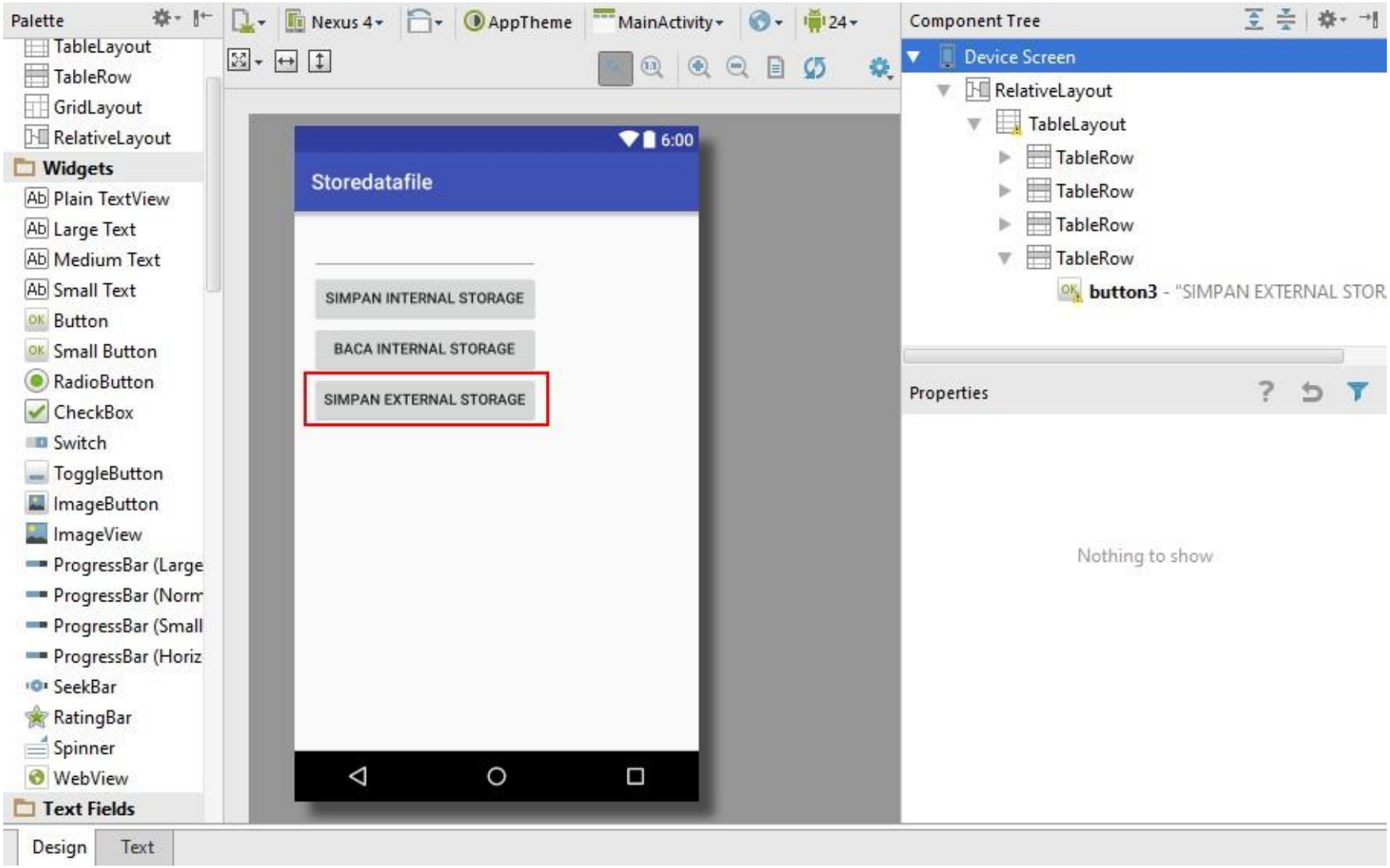
This can be a removable storage media (such as an SD card) or an internal (non-removable) storage.

All applications can read and write files placed on the external storage and the user can remove them. We need to check if the SD card is available and if we can write to it.

Permission in androidmanifest.xml

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

If you need to both read and write files, then you need to request only the WRITE_EXTERNAL_STORAGE permission, because it implicitly requires read access as well.


Palette
TableLayout
TableRow
GridLayout
RelativeLayout
Widgets
Plain TextView
Large Text
Medium Text
Small Text
Button
Small Button
RadioButton
CheckBox
Switch
ToggleButton
ImageButton
ImageView
ProgressBar (Large
ProgressBar (Norm
ProgressBar (Small
ProgressBar (Horiz
SeekBar
RatingBar
Spinner
WebView
Text Fields
Nexus 4
AppTheme
MainActivity
24
6:00
Storedatafile
SIMPAN INTERNAL STORAGE
BACA INTERNAL STORAGE
SIMPAN EXTERNAL STORAGE
Component Tree
Device Screen
RelativeLayout
TableLayout
TableRow
TableRow
TableRow
TableRow
button3 - "SIMPAN EXTERNAL STOR
Properties
Nothing to show
Design
Text

```
package com.tekom.home.storedatafile;

import ...

public class MainActivity extends AppCompatActivity {
    private EditText myedittext;
    private Button simpaninternalbutton;
    private Button bacainternalbutton;
    private Button simpanexternalbutton;

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

        myedittext = (EditText)findViewById(R.id.editText);
        simpaninternalbutton = (Button)findViewById(R.id.button);
        bacainternalbutton = (Button)findViewById(R.id.button2);
        simpanexternalbutton = (Button)findViewById(R.id.button3);

        simpaninternalbutton.setOnClickListener((view) -> {
                try {
                    FileOutputStream fOut = openFileOutput("myfile", Context.MODE_PRIVATE);
                    fOut.write(myedittext.getText().toString().getBytes());
                    fOut.close();
                    Toast.makeText(MainActivity.this, "data telah disimpan", Toast.LENGTH_SHORT).show();
                } catch (Exception e) {
                    e.printStackTrace();
                }
        });

        bacainternalbutton.setOnClickListener((view) -> {
                //reading text from file
```

```
simpanexternalbutton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        try {
        File myExternalFile = new File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOCUMENTS), "myextfile");
        FileOutputStream fos = new FileOutputStream(myExternalFile);
        fos.write(myedittext.getText().toString().getBytes());
        fos.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
});
```

If you'd like to access possible locations while also supporting Android 4.3 and lower, use the support library's static method,`ContextCompat.getExternalFilesDirs()`.

```
simpanexternalbutton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        try {
        File myExternalFile = new File(Environment.getExternalStoragePublicDirectory(Environment.D), "myextfile");
        FileOutputStream fos = new FileOutputStream(myExternalFile);
        fos.write(myedittext.getText().toString().getBytes());
        fos.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
});
if (!isExternalStorageAvailable() || isExternalStorageReadOnly()) {
    simpanexternalbutton.setEnabled(false);
}
else {
    simpanexternalbutton.setEnabled(true);
}
}
```

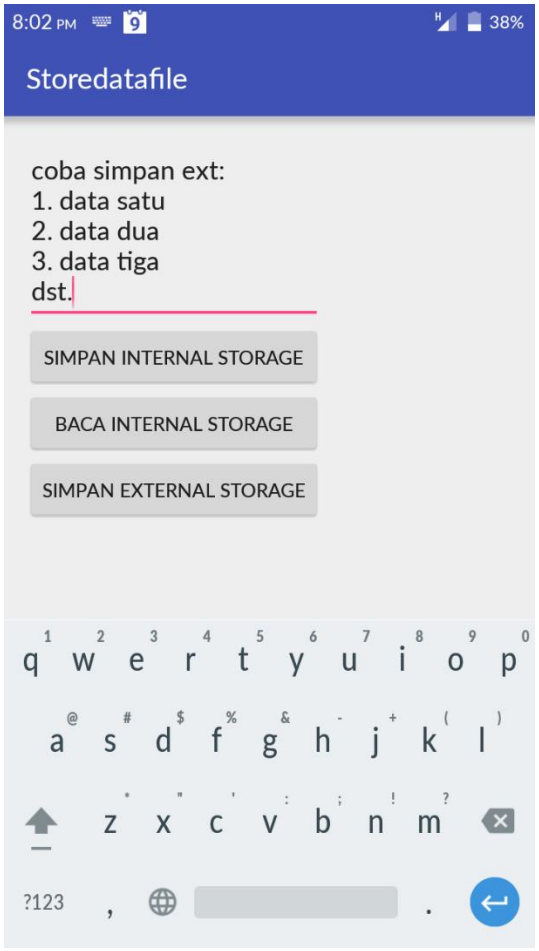
```
DIRECTORY_ALARMS          String
DIRECTORY_DCIM            String
DIRECTORY_DOCUMENTS       String
DIRECTORY_DOWNLOADS       String
DIRECTORY_MOVIES          String
DIRECTORY_MUSIC           String
DIRECTORY_NOTIFICATIONS   String
DIRECTORY_PICTURES        String
DIRECTORY_PODCASTS        String
DIRECTORY_RINGTONES       String
```

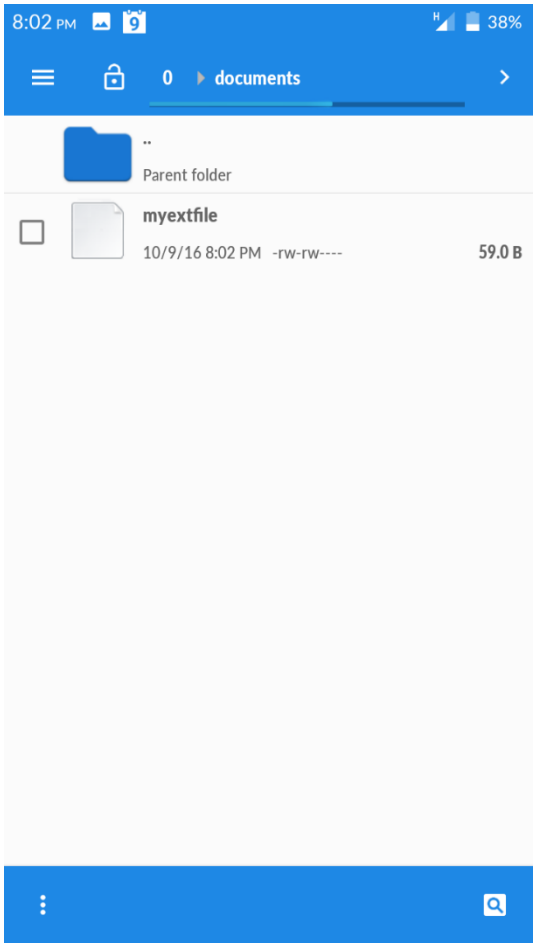

FileOutputStream fos = new FileOutputStream(myExternalFile);
fos.write(myedittext.getText().toString().getBytes());
fos.close();
} catch (IOException e) {
e.printStackTrace();
}
}
});
if (!isExternalStorageAvailable() || isExternalStorageReadOnly()) {
simpanexternalbutton.setEnabled(false);
}
else {
simpanexternalbutton.setEnabled(true);
}
}
Add this INSIDE
void onCreate(Bundle savedInstanceState) {
private static boolean isExternalStorageReadOnly() {
String extStorageState = Environment.getExternalStorageState();
if (Environment.MEDIA_MOUNTED_READ_ONLY.equals(extStorageState)) {
return true;
}
return false;
}
But Add this OUTSIDE
void onCreate(Bundle savedInstanceState) {
private static boolean isExternalStorageAvailable() {
String extStorageState = Environment.getExternalStorageState();
if (Environment.MEDIA_MOUNTED.equals(extStorageState)) {
return true;
}
return false;
}
}

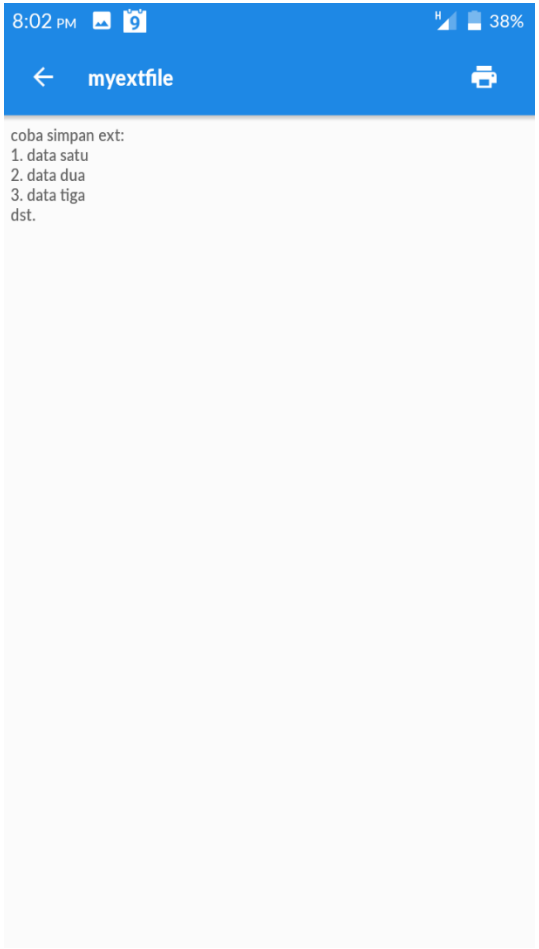
Add the required permission in androidmanifest.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.tekom.home.storedatafile">
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Storedatafile"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Result


8:02 PM
38%
Storedatafile
coba simpan ext:
1. data satu
2. data dua
3. data tiga
dst.
SIMPAN INTERNAL STORAGE
BACA INTERNAL STORAGE
SIMPAN EXTERNAL STORAGE


8:02 PM
38%
0
documents
..
Parent folder
myextfile
10/9/16 8:02 PM -rw-rw----
59.0 B


8:02 PM
38%
myextfile
coba simpan ext:
1. data satu
2. data dua
3. data tiga
dst.

Using database (not discussed)

- **You can use SQLite to keep data as records. Refer to online tutorial such as:**

http://www.androidhive.info/2011/11/android-sqlite-database-tutorial/

TERIMA KASIH