

JUNIOR MOBILE PROGRAMMER

Merancang Database dan Data Persistence Pada Mobile Data

Deskripsi Singkat

Deskripsi Singkat mengenai Topik

Topik ini akan membahas tentang mendesain internal storage pada aplikasi berbasis mobile dan mendesain eksternal storage pada aplikasi berbasis mobile. Topik ini akan membahas tentang mendesain internal storage pada aplikasi berbasis mobile dan mendesain eksternal storage pada aplikasi berbasis mobile.

Tujuan Pelatihan

Setelah pertemuan selesai peserta pelatihan mampu:

1. menjelaskan konseptual internal dan external storage pada aplikasi berbasis mobile
2. menjelaskan internal dan external storage berdasarkan spesifikasi device mobile dan operating system berbasis mobile
3. merancang media penyimpanan internal dan external dengan proses create, read, update, delete dan data berjalan dibangun sesuai kebutuhan

Materi Yang akan disampaikan:

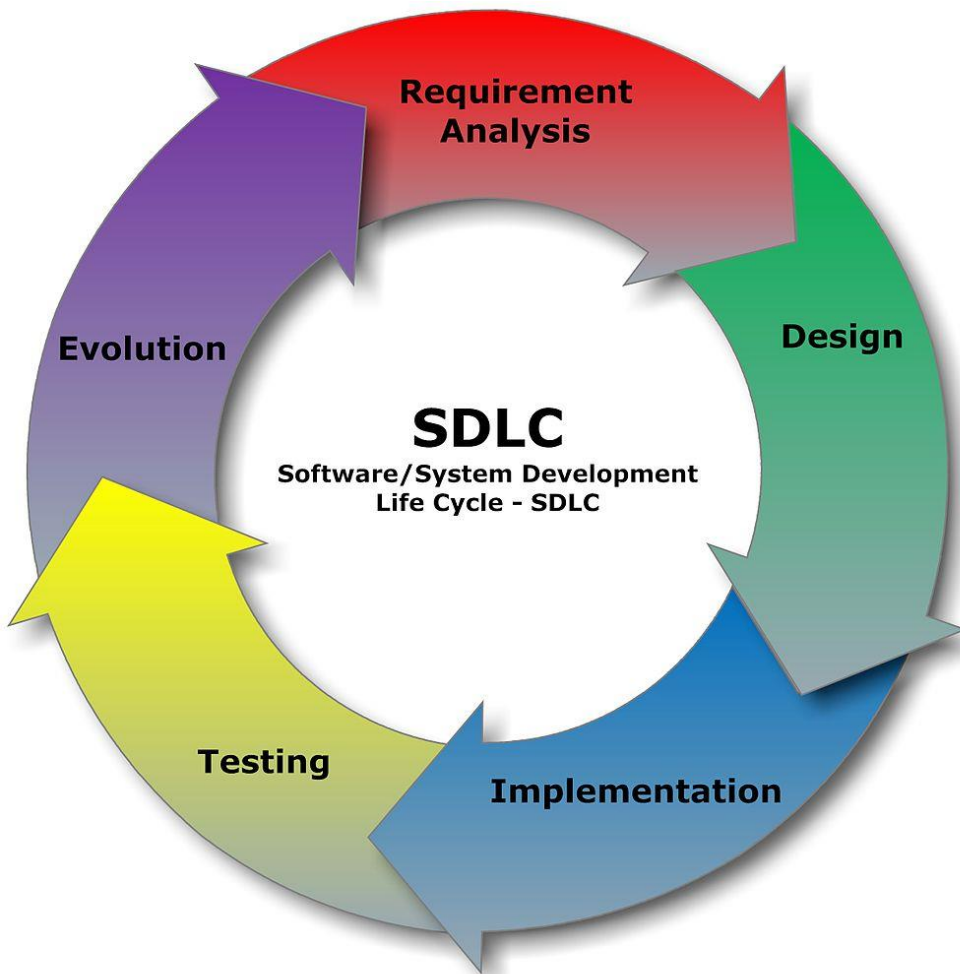
1. Data Storage
2. Internal Storage
3. External Storage
4. File

Tugas :

Contoh aplikasi persistence <http://bit.ly/2WaH2bl>

Pengenalan SDLC

Pelatihan



- **Requirement Analysis:** Penggalan kebutuhan aplikasi termasuk kebutuhan fungsional dan non fungsionalnya.
- **Design:** Perancangan aplikasi termasuk perancangan data, perancangan kelas, desain UI, dlsb.
- **Implementasi:** Dapat juga dikatakan tahap coding / pembuatan source code aplikasi
- **Testing:** Tahapan untuk menguji kesiapan aplikasi
- **Evolution:** Tahap pengembangan aplikasi yang berkelanjutan termasuk penambahan fitur, bug fixing, penyesuaian kebutuhan, dlsb.

Data Storage

Pelatihan

- ❖ Penyimpanan data pada android terdiri dari dua jenis data, yaitu :
 1. data *temporary*
 2. data *persistence*.
- ❖ Data *temporary* umumnya diproses melalui *internal storage* dalam bentuk *cache*.
- ❖ Data *persistence* disimpan secara *permanen* dalam suatu file.

Mengapa *Data Persistence* diperlukan?

Pelatihan

Android dapat di-*shutdown* dan di-*restart* kapanpun dibutuhkan, namun bagaimana ketika:

- merotasi *screen*
- mengubah Bahasa
- aplikasi berjalan secara *background process* dan disimpan dalam *short-in memory*
- menekan tombol Back

Apakah yang akan terjadi? data hilang!

Lalu bagaimana solusinya?

Pelatihan

- ❖ Android menyediakan beberapa solusi untuk menangani data *persistence*, tergantung dari sifatnya apakah *private* atau *public*.
- ❖ Beberapa solusi yang dipilih harus sesuai dengan kebutuhan spesifiknya, seperti apakah data yang disimpan akan bersifat *private* atau boleh diakses secara bebas (*public*) serta berapa banyak *space* yang dibutuhkannya.
- ❖ Android juga menyediakan cara untuk membuka data privat ke aplikasi lain (apabila diperlukan), melalui **Content Provider**.

Beberapa opsi *data storage*

Pelatihan

- ❖ *Shared Preferences*: menyimpan data primitive berupa *key-value pairs*.
- ❖ *Internal Storage*: menyimpan data *private* pada *device memory*.
- ❖ *External Storage*: menyimpan data public pada *shared external storage*.
- ❖ *Database*: menyimpan data struktur pada sebuah database
- ❖ *Network Connection*: menyimpan data pada web server
- ❖ *Bundle Class*: memetakan data string dalam bentuk map
- ❖ *File IO*: menyimpan data berupa file pada internal storage

Android File System

Pelatihan

- ***Internal Storage*** – direktori *private* yang hanya diakses pada aplikasi itu sendiri.
- ***External Storage*** – direktori *public* dapat diakses secara lebih luas pada aplikasi itu sendiri dan aplikasi lainnya.

Aplikasi dapat mengakses struktur direktori, Struktur dan operasinya mirip seperti yang digunakan pada **Linux** dan **java.io**

Karakteristik dari Internal Storage

Pelatihan

- ❖ Selalu tersedia (dapat diakses).
- ❖ Menggunakan *file system* dari *device*.
- ❖ Hanya bisa diakses pada aplikasi itu sendiri, kecuali jika secara eksplisit diatur agar dapat dibaca / ditulis.
- ❖ Ketika aplikasi di-uninstall, sistem akan menghapus semua file aplikasi dari *internal storage*.

Karakteristik dari Eksternal Storage

Pelatihan

- ❖ Tidak selalu tersedia (dapat dihapus)
- ❖ Menggunakan *file system* dari *device* atau dari penyimpanan fisik eksternal seperti SD Card.
- ❖ *World-readable*, semua aplikasi dapat membaca file tersebut.
- ❖ Ketika aplikasi di-uninstall, *system* tidak akan menghapus file *private* dari aplikasi tersebut.

Kapan kita perlu internal dan eksternal storage

Pelatihan

Internal storage paling baik digunakan jika:

- Tidak ada yang boleh mengakses file selain aplikasi itu sendiri.

Eksternal storage paling baik digunakan jika:

- File tidak memerlukan batasan akses (semua *public*).
- File dapat di-share ke aplikasi lain.
- Membolehkan pengguna untuk mengakses file melalui komputer (via *USB Connection*).

Internal Storage

Pelatihan

- File bersifat private
- Aplikasi akan selalu meminta izin untuk read/write data
- Direktori penyimpanan *persistence* – `getFilesDir()`
- Direktori penyimpanan *temporary* – `getCacheDir()`

Contoh pembuatan File

Pelatihan

```
File file = new File(context.getFilesDir(), filename);
```

Untuk fungsi-fungsi pengelolaan filenya, menggunakan standard [java.io](https://docs.oracle.com/javase/7/docs/api/java/io/package-summary.html) *file operator* atau *stream*.

Langkah menulis file *private* ke *Internal Storage*

Pelatihan

Untuk membuat dan menuliskan file *private* ke *internal storage* :

1. Panggil `openFileOutput()` dengan nama file dan mode pengoperasiannya. Proses ini mengembalikan `FileOutputStream`.
2. Tulis pada file melalui method `write()`.
3. Tutup aliran (***stream***) melalui method `close()`.

Contoh kode program untuk menulis ke file

Pelatihan

```
String filename = "myfile";  
String fileContents = "Hello world!";  
FileOutputStream outputStream;  
  
try {  
    outputStream = openFileOutput(filename, Context.MODE_PRIVATE);  
    outputStream.write(fileContents.getBytes());  
    outputStream.close();  
} catch (Exception e) {  
    e.printStackTrace();  
}
```

- `MODE_PRIVATE` akan membuat file (atau mengganti/*replace* file dengan nama yang sama) dan membuatnya *private* untuk aplikasi anda. Mode lain yang tersedia antara lain :
 - `MODE_APPEND`
 - `MODE_WORLD_READABLE` *
 - `MODE_WORLD_WRITEABLE` *

Catatan : * sudah tidak digunakan lagi sejak API level 17 (dimulai dari android N) karena penggunaannya akan berdampak pada *SecurityException*.

Langkah membaca file *private* dari *internal storage* Pelatihan

Untuk membaca file *private* dari *internal storage* :

1. Panggil `openFileInput()` dan memberikan nama file yang ingin dibaca. Proses ini mengembalikan `FileInputStream`
2. Baca byte dari file dengan method `read()`
3. Tutup aliran (*stream*) dengan method `close()`

Contoh kode program untuk membaca dari file

Pelatihan

```
String FILENAME = "hello_file.txt";  
FileInputStream fin = openFileInput(FILENAME);  
StringBuffer fileContent = new StringBuffer("");  
  
byte[] buffer = new byte[1024];  
while( (n=fin.read(buffer)) != -1) {  
    fileContent.append(new String(buffer, 0, n));  
}  
fin.close();
```

Menyimpan file *cache*

Pelatihan

- `getCacheDir()` – digunakan untuk membuka file pada direktori *internal storage*.
- `getCacheExternalDir()` – digunakan untuk membuka *cache* pada *external storage*.

Contoh kode untuk menuliskan file *cache*

Pelatihan

- `createTempFile()` akan membuat file pada direktori *cache* yang tersimpan secara privat pada aplikasi.

```
private File getTempFile(Context context, String url) {  
    File file;  
    try {  
        String fileName = Uri.parse(url).getLastPathSegment();  
        file = File.createTempFile(fileName, null,  
context.getCacheDir());  
    } catch (IOException e) {  
        // Error while creating file  
    }  
    return file;  
}
```

Beberapa Method Penting Lainnya

Pelatihan

- `getFilesDir()` – mendapatkan path absolut ke direktori file pada *internal storage*.
- `getDir()` – membuat (atau membuka yang sudah ada) direktori pada *internal storage*.
- `deleteFile()` – menghapus file yang disimpan pada *internal storage*.
- `fileList()` – mengembalikan daftar file yang saat ini telah disimpan dalam *internal storage*.

Eksternal Storage

Pelatihan

Android menyediakan metode penyimpanan file external pada ponsel atau tablet yang terbagi berdasarkan beberapa jenis, yaitu:

1. Untuk ponsel Android generasi awal biasanya hanya dibagi menjadi dua bagian, yaitu:
 - Internal storage
 - SD card

Eksternal Storage

Pelatihan

2. Untuk Smartphone yang tidak memiliki slot SD card juga dibagi menjadi dua bagian, yaitu:
 - Internal/System storage,
 - Phone/USB Storage.
3. Untuk smartphone dan tablet Android saat ini pada umumnya dibagi menjadi tiga bagian, yaitu:
 - Internal/System storage,
 - Phone/USB Storage,
 - SD card.

Perbandingan Storage Antar Platform

Pelatihan

❖ Perbandingan Storage pada Android dan iPhone

- iPhone

iPhone memiliki kapasitas internal lebih tinggi dari android dan partisi untuk OS lebih fleksibel namun tidak support untuk external storage atau tambahan memory, kapasitas nya adalah 8GB,16GB, 32GB, 64GB, 128GB, 256GB belum termasuk partisi untuk OS yang memakan sekitar 1,5GB – 2GB

Perbandingan Storage Antar Platform

Pelatihan

. Android

Android memiliki kapasitas internal yang lebih kecil dari iOS dengan partisi untuk OS biasanya memakan separuh dari kapasitas internalnya, memiliki opsi external storage, kapasitas internal nya mulai dari 4GB, 8GB, 16GB, 32GB, 64GB belum termasuk partisi OS yang memakan separuh dari internal storage.

Konfigurasi pada AndroidManifest.xml

Pelatihan

Agar bisa membaca atau menulis file pada penyimpanan eksternal, aplikasi harus memperoleh `READ_EXTERNAL_STORAGE` atau izin sistem `WRITE_EXTERNAL_STORAGE` yang di daftarkan pada `AndroidManifest.xml`. Misalnya seperti baris kode dibawah:

```
<manifest>

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

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

</manifest>
```

Fungsi Penting dalam Penyimpanan Eksternal

Pelatihan

- `getExternalStorageState()` : untuk memeriksa apakah media tersebut tersedia.
- `getExternalStorageDirectory()` : nilai dari fungsi ini menghasilkan lokasi direktori penyimpanan eksternal / eksternal utama.
- `getExternalStoragePublicDirectory(String type)` : nilai dari fungsi ini menghasilkan lokasi direktori penyimpanan eksternal / eksternal utama sesuai dari isi Argumen `type` yang ada pada fungsi tersebut

External Storage State

Pelatihan

Dibawah adalah beberapa contoh baris kode penggunaan *External Storage State*

```
/* Periksa apakah penyimpanan eksternal
tersedia untuk membaca dan menulis */

public boolean isExternalStorageWritable() {
    String state = Environment.getExternalStorageState();
    if (Environment.MEDIA_MOUNTED.equals(state)) {
        return true;
    }
    return false;
}
```

```
/* Periksa apakah penyimpanan eksternal
setidaknya tersedia untuk dibaca */

public boolean isExternalStorageReadable() {
    String state = Environment.getExternalStorageState();
    if (Environment.MEDIA_MOUNTED.equals(state) ||
        Environment.MEDIA_MOUNTED_READ_ONLY.equals(state)) {
        return true;
    }
    return false;
}
```

Jenis-Jenis Status External Storage

Pelatihan

Adapun nilai balik status dari fungsi `getExternalStorageState()` di antaranya :

- `MEDIA_UNKNOWN` : kartu SD Card tidak dapat dikenali
- `MEDIA_REMOVED` : status tidak ada kartu SD Card yang tersedia
- `MEDIA_UNMOUNTED` : Status penyimpanan jika media ada tetapi tidak dipasang.
- `MEDIA_MOUNTED` : Status penyimpanan jika media ada dan terpasang dengan akses baca / tulis.
- `MEDIA_MOUNTED_READ_ONLY` : Status penyimpanan jika media ada dan terpasang dengan akses hanya baca.

External Files Dir

Pelatihan

```
public void buatFile() {  
    String cashback = "tulisi isi file";  
    String state = Environment.getExternalStorageState();  
    //external storage availability check  
    if (!Environment.MEDIA_MOUNTED.equals(state)) {  
        return;  
    }  
    //Pembuatan file yang akan disimpan denga direktori Eksternal  
    File file = new File(Environment.getExternalStorageDirectory(), child: "DemoFile.txt");  
    FileOutputStream outputStream = null;  
    try {  
        file.createNewFile();  
        //Argumen kedua konstruktor FileOutputStream menunjukkan apakah  
        // akan menambahkan isi data atau membuat file baru jika sudah ada  
        outputStream = new FileOutputStream(file, append: true);  
        outputStream.write(cashback.getBytes());  
        outputStream.flush();  
        outputStream.close();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

External Storage Public Directory

Pelatihan

Penggunaan public directory
Hanya berlaku untuk versi OS (19)
Kitkat keatas

```
public void buatFile() {  
    String cashback = "tulisi isi file";  
    String state = Environment.getExternalStorageState();  
    //external storage availability check  
    if (!Environment.MEDIA_MOUNTED.equals(state)) {  
        return;  
    }  
    //Pembuatan file yang akan disimpan di direktori Eksternal  
    File file = new File(Environment.getExternalStoragePublicDirectory(  
        Environment.DIRECTORY_DOCUMENTS), child: "namafile.txt");  
    FileOutputStream outputStream = null;  
    try {  
        file.createNewFile();  
        //Argumen kedua konstruktor FileOutputStream menunjukkan apakah  
        // akan menambahkan isi data atau membuat file baru jika sudah ada  
  
        outputStream = new FileOutputStream(file, append: true);  
        outputStream.write(cashback.getBytes());  
        outputStream.flush();  
        outputStream.close();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

Jenis-Jenis Direktori Publik

Pelatihan

Adapun nilai-nilai static yang sering digunakan dari fungsi `getExternalStoragePublicDirectory(String type)` di antaranya:

- `DIRECTORY_DOCUMENTS` : Direktori standar tempat meletakkan dokumen yang telah dibuat oleh pengguna.
- `DIRECTORY_DOWNLOADS` : Direktori standar tempat meletakkan file yang telah diunduh oleh pengguna.
- `DIRECTORY_MOVIES` : Direktori standar untuk menempatkan film yang tersedia bagi pengguna.
- `DIRECTORY_PICTURES` : Direktori standar tempat meletakkan gambar yang tersedia bagi pengguna.

Membuat Fungsi Create,Read Update dan Delete Pelatihan

Pada penyimpanan eksternal dapat melakukan beberapa hal diantaranya :

- Create
- Read
- Update
- dan Delete

Create File

Pelatihan

```
public void buatFile() {  
    String cashback = "tulis isi file";  
    String state = Environment.getExternalStorageState();  
    //external storage availability check  
    if (!Environment.MEDIA_MOUNTED.equals(state)) {  
        return;  
    }  
    //Pembuatan file yang akan disimpan dengan direktori Eksternal  
    File file = new File(Environment.getExternalStorageDirectory(), child: "DemoFile.txt");  
    FileOutputStream outputStream = null;  
    try {  
        file.createNewFile();  
        //Argumen kedua konstruktor FileOutputStream menunjukkan apakah  
        // akan menambahkan isi data atau membuat file baru jika sudah ada  
        outputStream = new FileOutputStream(file, append: true);  
        outputStream.write(cashback.getBytes());  
        outputStream.flush();  
        outputStream.close();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

Read File

Pelatihan

```
public void bacaFile() {  
    File sdcard = Environment.getExternalStorageDirectory();  
    File file = new File(sdcard, FILENAME);  
  
    if(file.exists()) {  
  
        StringBuilder text = new StringBuilder();  
  
        try {  
            BufferedReader br = new BufferedReader(new FileReader(file));  
  
            String line = br.readLine();  
  
            while (line != null) {  
                text.append(line);  
                line = br.readLine();  
            }  
            br.close();  
        } catch (IOException e) {  
            System.out.println("Error " + e.getMessage());  
        }  
  
        textBaca.setText(text.toString());  
    }  
}
```

File diubah menjadi stream menggunakan FileReader sehingga dapat dibaca dalam BufferedReader, setelah BufferedReader mempunyai value dari FileReader, maka data tersebut dibaca baris perbaris menggunakan StringBuilder.

Delete File

Pelatihan

```
void hapusFile() {  
    File file = new File(Environment.getExternalStorageDirectory(), FILENAME);  
    if (file.exists()) {  
        file.delete();  
    }  
}
```

`file.delete()` mempunyai nilai balik berupa Boolean yang mana jika nilai true, maka data berhasil dihapus

Latihan 1

Pelatihan

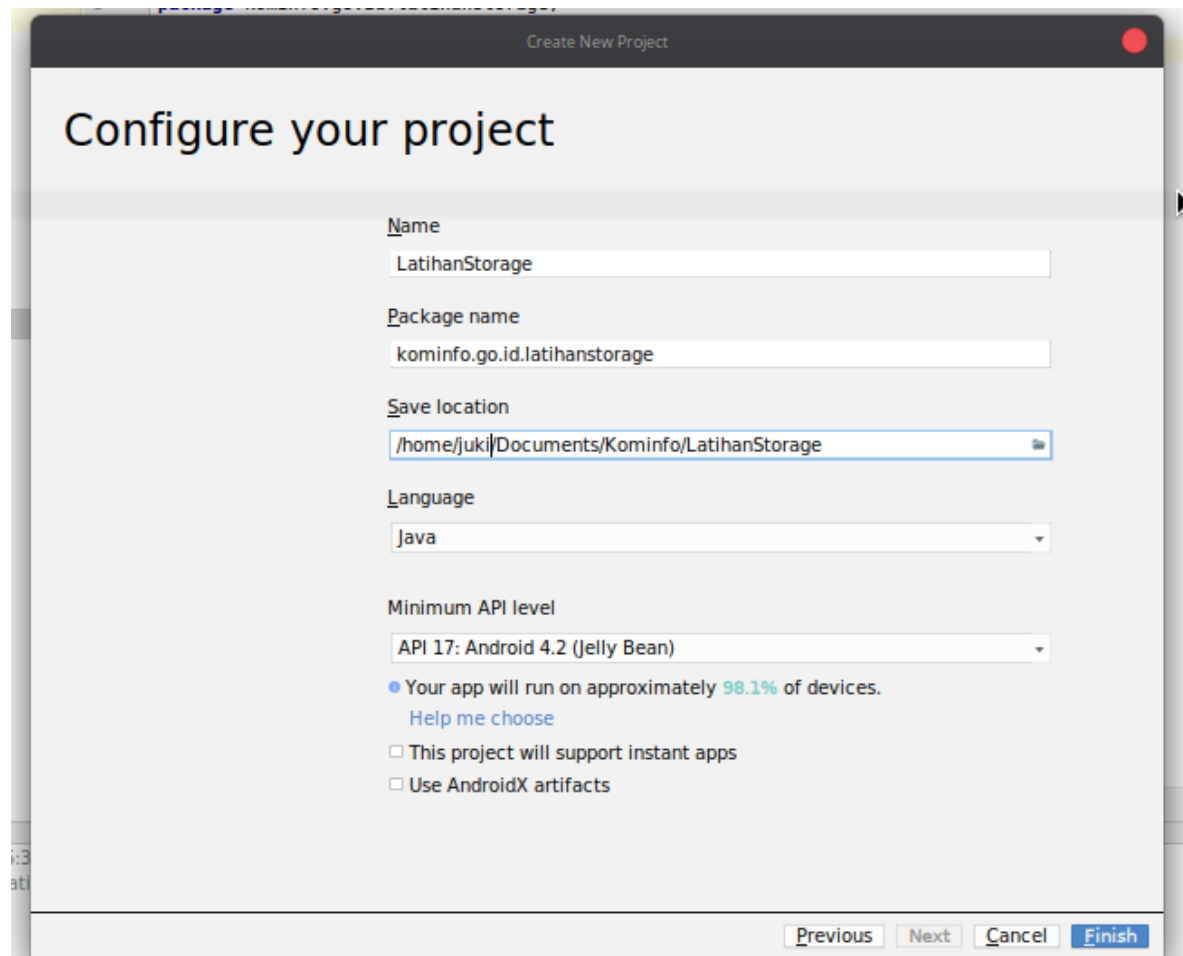
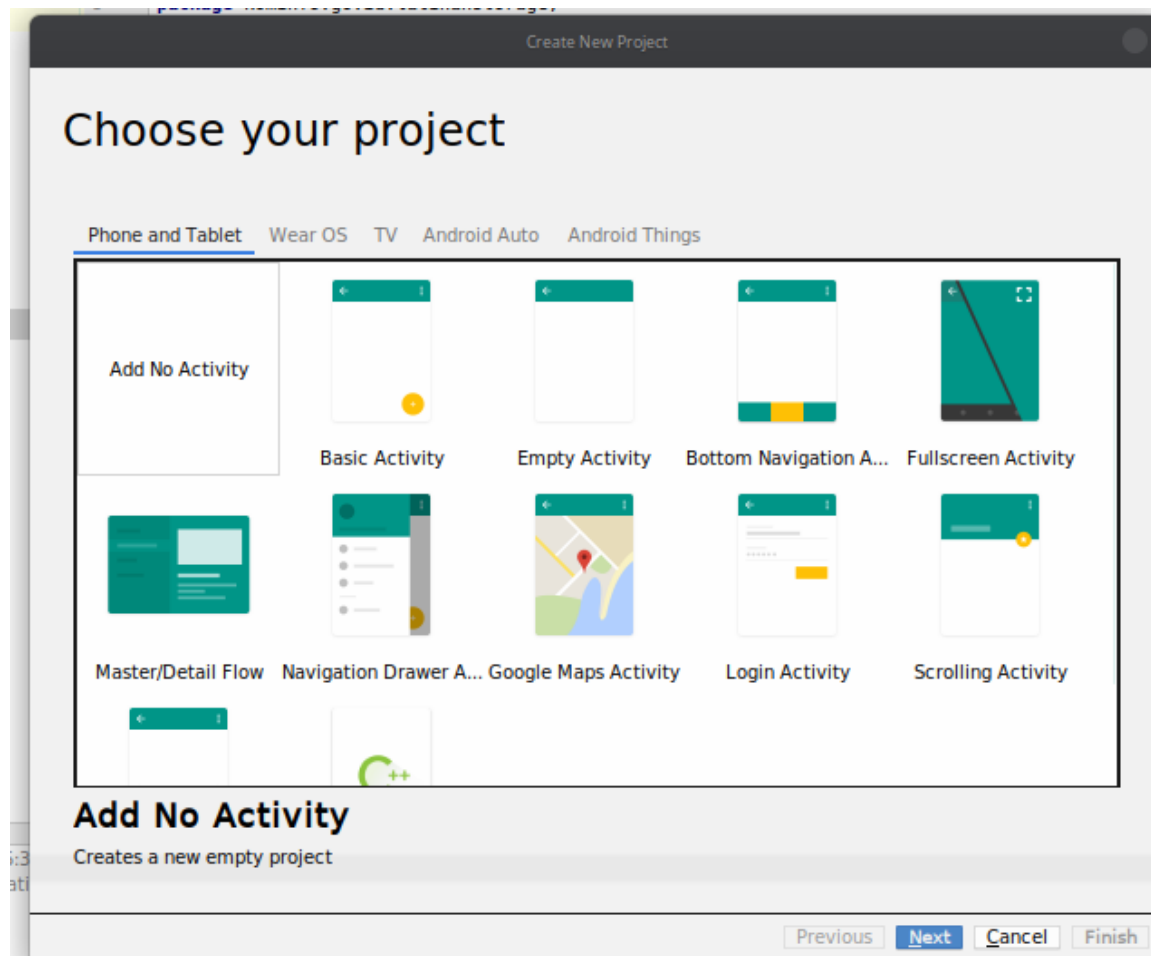
Tahapan-tahapan pembuatan Aplikasi pemanfaatan *Internal Storage* :

- **[Langkah 1]** Buatlah sebuah Project Baru Android dan beri nama Aplikasi `LatihanStorage`
- **[Langkah 2]** Buat Sebuah Java Kelas yang mewariskan Kelas Activity, beri nama Kelas `InternalStorageActivity.java`
- **[Langkah 3]** Buat Sebuah *Layout* xml dengan nama `layout_internal.xml`
- **[Langkah 4]** Pada Java Kelas `InternalStorageActivity.java` Tambahkan Baris Fungsi *Create, Read, Update* dan *Delete*.
- **[Langkah 5]** Jalankan Aplikasi.

Latihan 1 – Langkah 1

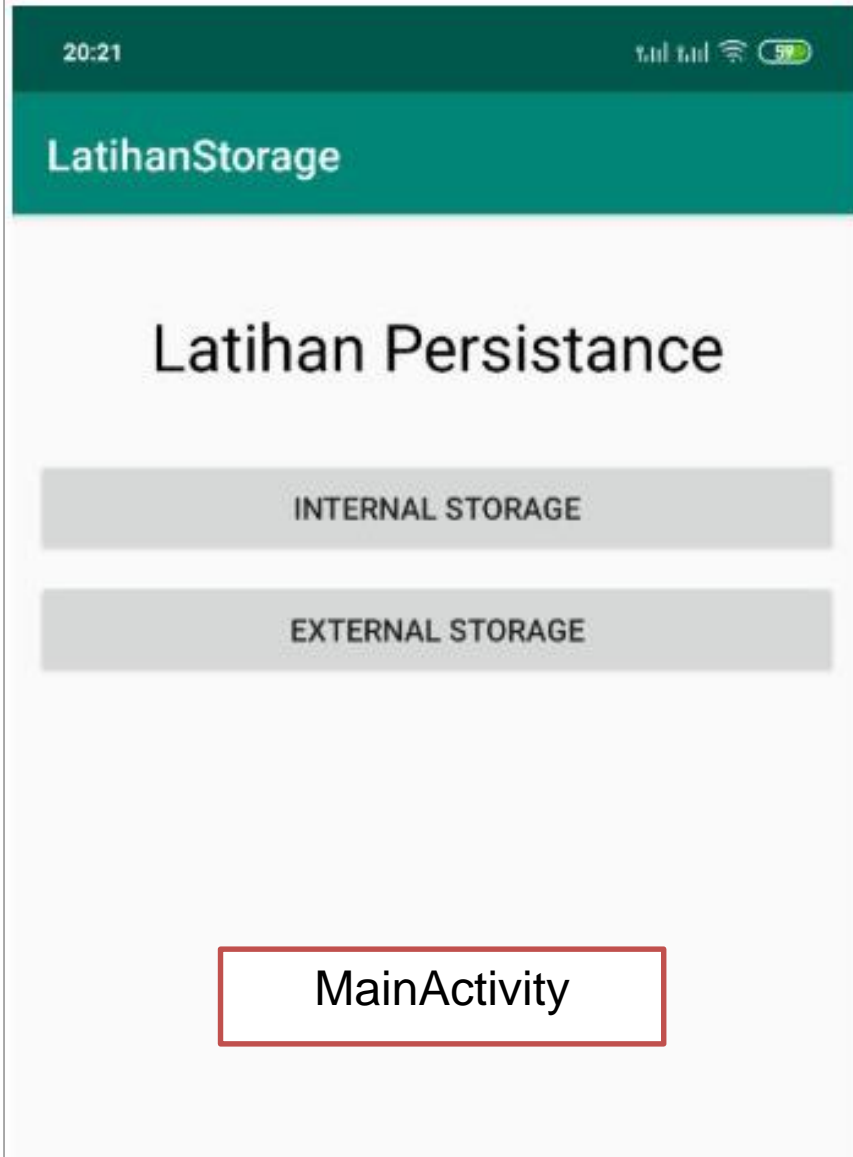
Pelatihan

[Langkah 1] Buatlah sebuah Project Baru Android dan beri nama Aplikasi LatihanStorage



Latihan 2

Pelatihan

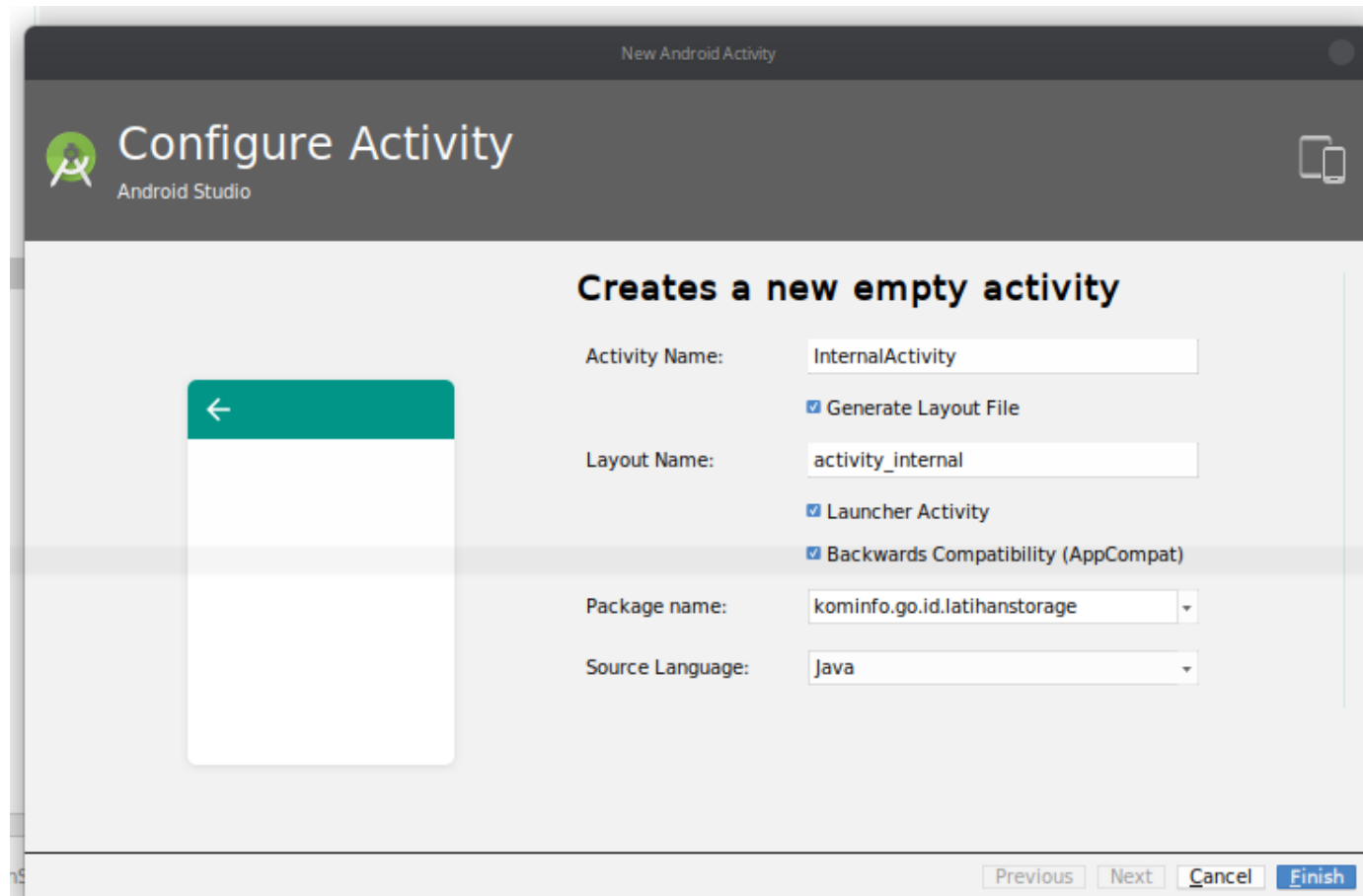


- ❖ [Langkah 1] Buat 2 *button* yang berfungsi untuk menyimpan data secara *internal* maupun *external*

Latihan 1 – Langkah 2

Pelatihan

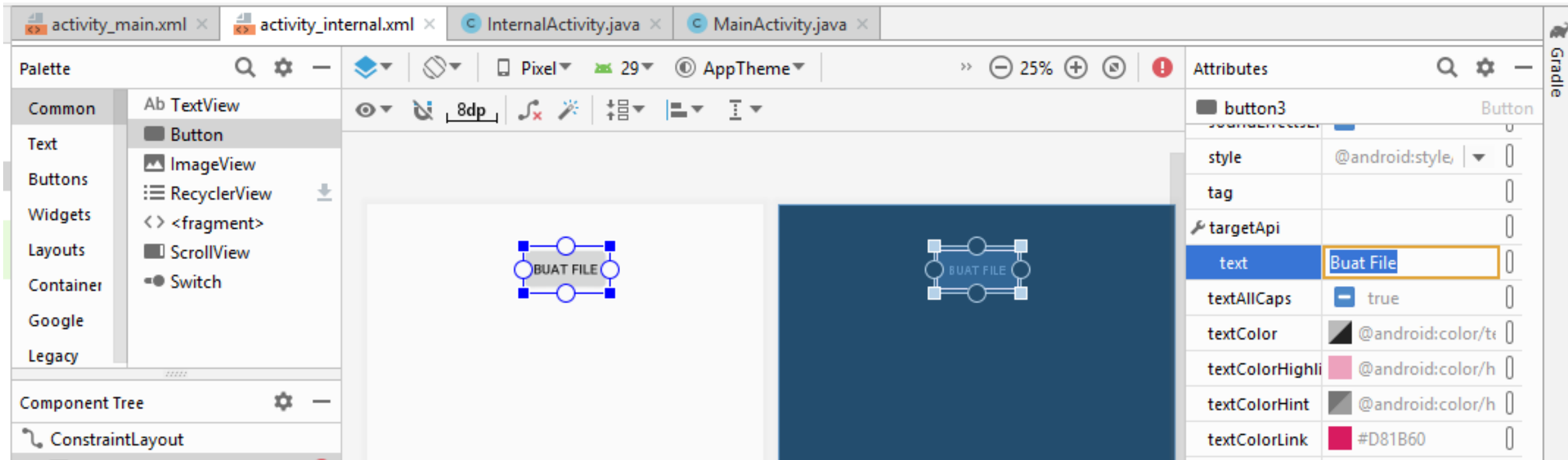
[Langkah 2] Buat Sebuah Activity, beri nama Kelas InternalActivity.java dan sebuah Layout dengan nama activity_internal.xml



Latihan 1 – Langkah 3

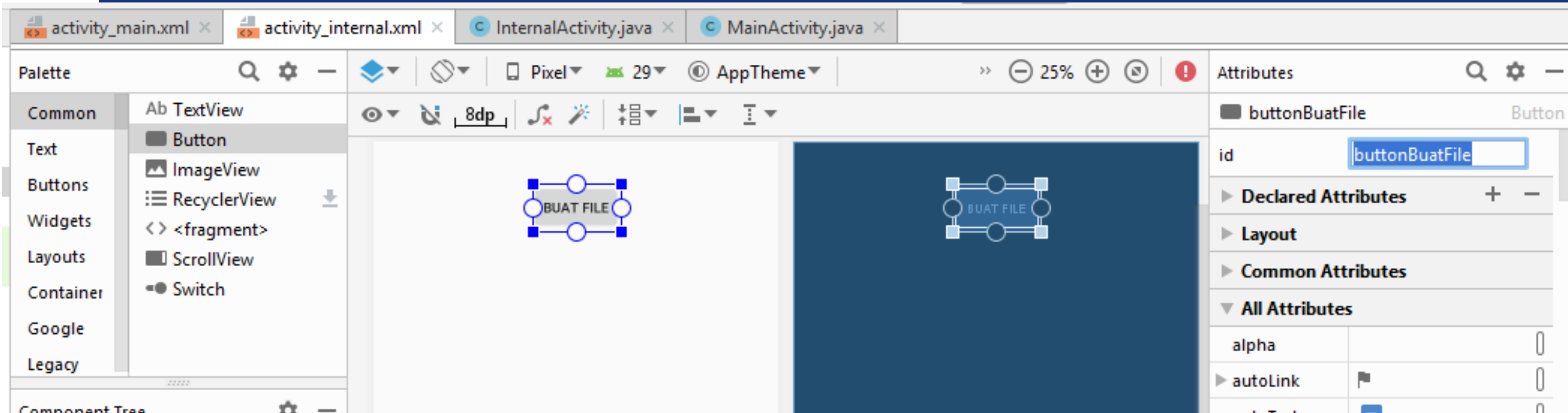
Pelatihan

[Langkah 3] Membuat sebuah view di dalam layout `activity_internal.xml`



Latihan 1 – Langkah 3

Pelatihan



Latihan 1 – Langkah 3

Pelatihan

BUAT FILE

UBAH FILE

BACA FILE

HAPUS FILE

BUAT FILE

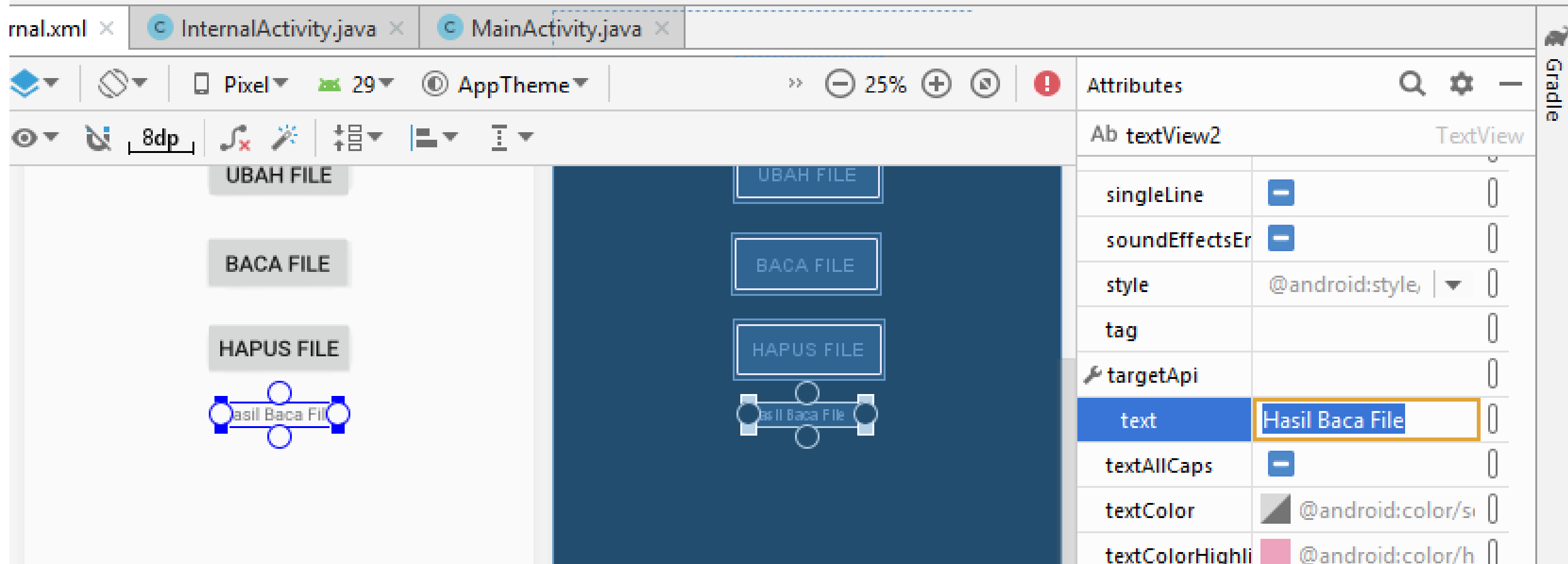
UBAH FILE

BACA FILE

HAPUS FILE

Latihan 1 – Langkah 3

Pelatihan



The screenshot displays the Android Studio IDE with the following components:

- Top Bar:** Shows the current file being edited, `MainActivity.java`.
- Toolbar:** Includes icons for navigation, zooming (25%), and other development tools.
- Main Canvas:** Displays a visual representation of the app's UI. It features three buttons: `UBAH FILE`, `BACA FILE`, and `HAPUS FILE`. Below the `BACA FILE` button is a text view containing the text `Hasil Baca File`.
- Right Sidebar:** The **Attributes** panel is open, showing the properties of the selected text view (`Ab textView2`). The `text` attribute is highlighted with a yellow border and contains the value `Hasil Baca File`.

Attribute	Value
<code>singleLine</code>	<code>0</code>
<code>soundEffectsEnabled</code>	<code>0</code>
<code>style</code>	<code>@android:style/</code>
<code>tag</code>	<code>0</code>
<code>targetApi</code>	<code>0</code>
<code>text</code>	<code>Hasil Baca File</code>
<code>textAllCaps</code>	<code>0</code>
<code>textColor</code>	<code>@android:color/s</code>
<code>textColorHighlight</code>	<code>@android:color/h</code>

Latihan 1 – Langkah 3

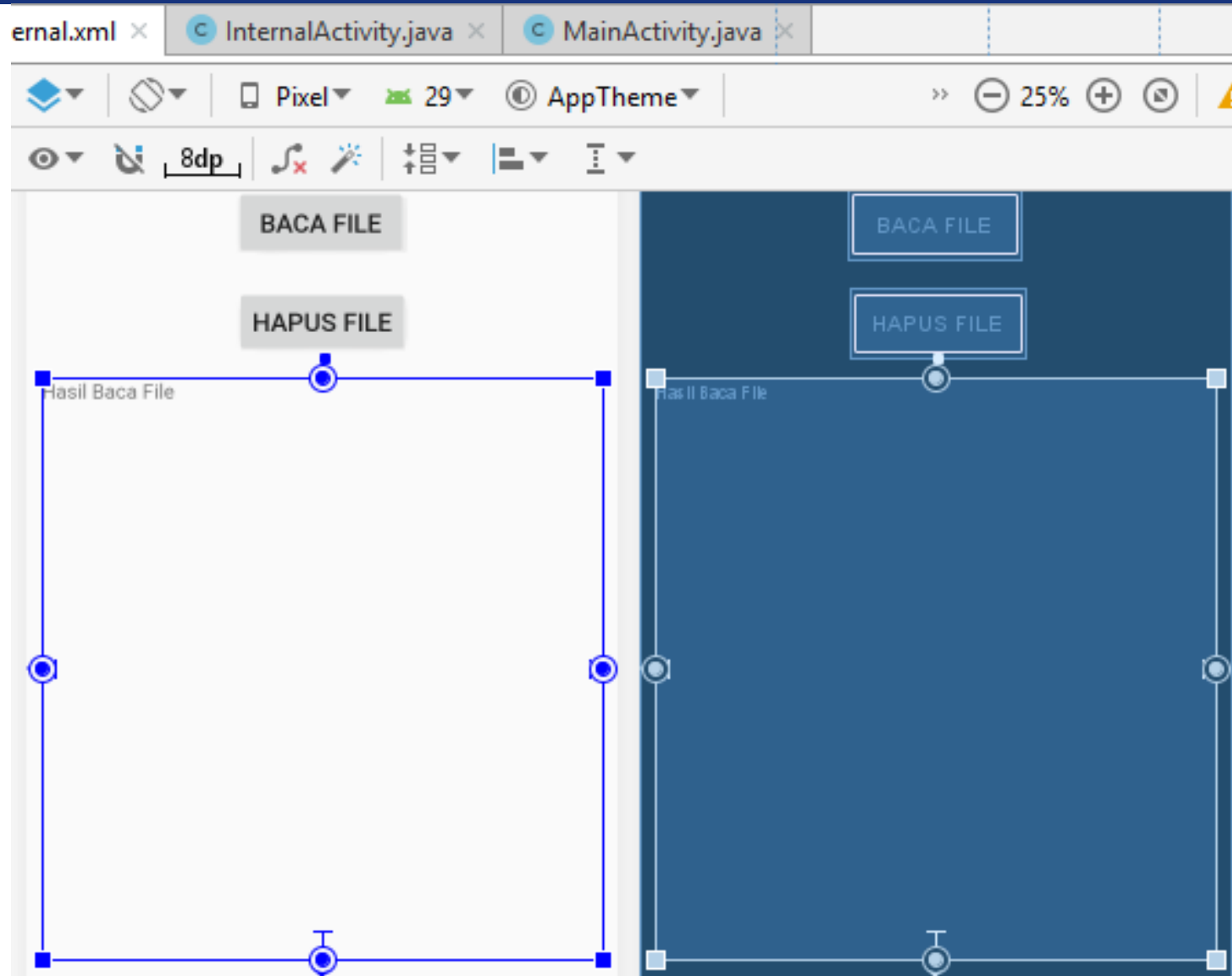
Pelatihan

The screenshot displays the Android Studio IDE. At the top, the tab bar shows 'Internal.xml', 'InternalActivity.java', and 'MainActivity.java'. The main editor area is split: the left pane shows the Java code for 'MainActivity.java', and the right pane shows the XML layout. The layout contains two buttons labeled 'BACA FILE' and 'HAPUS FILE', and a text field labeled 'Hasil Baca File'. The right sidebar shows the 'Attributes' panel for the selected 'textView2' widget. The 'id' attribute is set to 'textBaca'. The 'Declared Attributes' section is expanded, showing 'alpha' with a value of 1.0.

Attributes	
Ab textView2	TextView
id	textBaca
▶ Declared Attributes + -	
▶ Layout	
▶ Common Attributes	
▼ All Attributes	
alpha	1.0

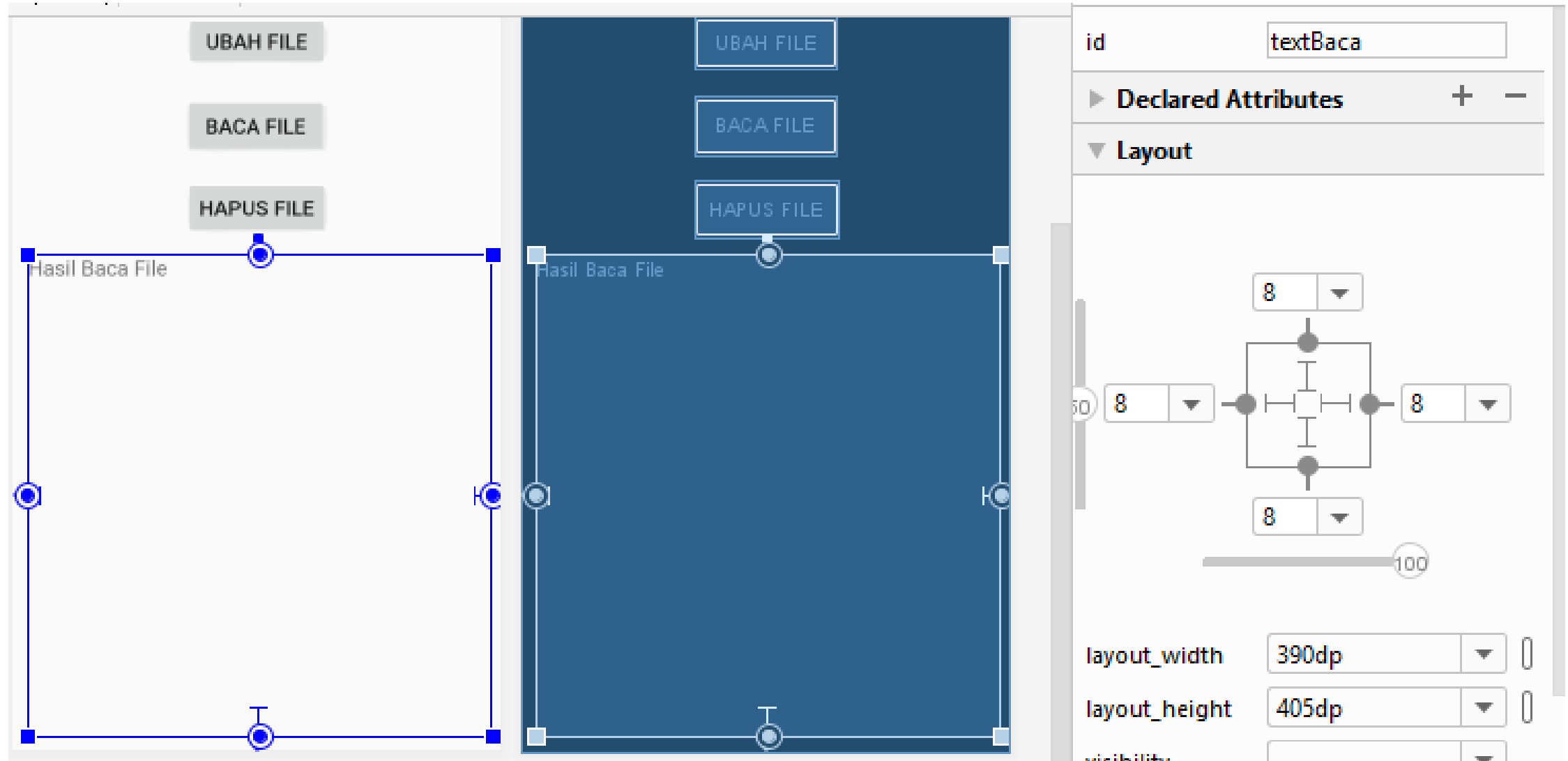
Latihan 1 – Langkah 3

Pelatihan



Latihan 1 – Langkah 3

Pelatihan



Latihan 1 – Langkah 3

Pelatihan

UBAH FILE

BACA FILE

HAPUS FILE

Hasil Baca File

UBAH FILE

BACA FILE

HAPUS FILE

Hasil Baca File

idtextBaca

Declared Attributes

Layout

8

8

8

8

100

layout_width0dp

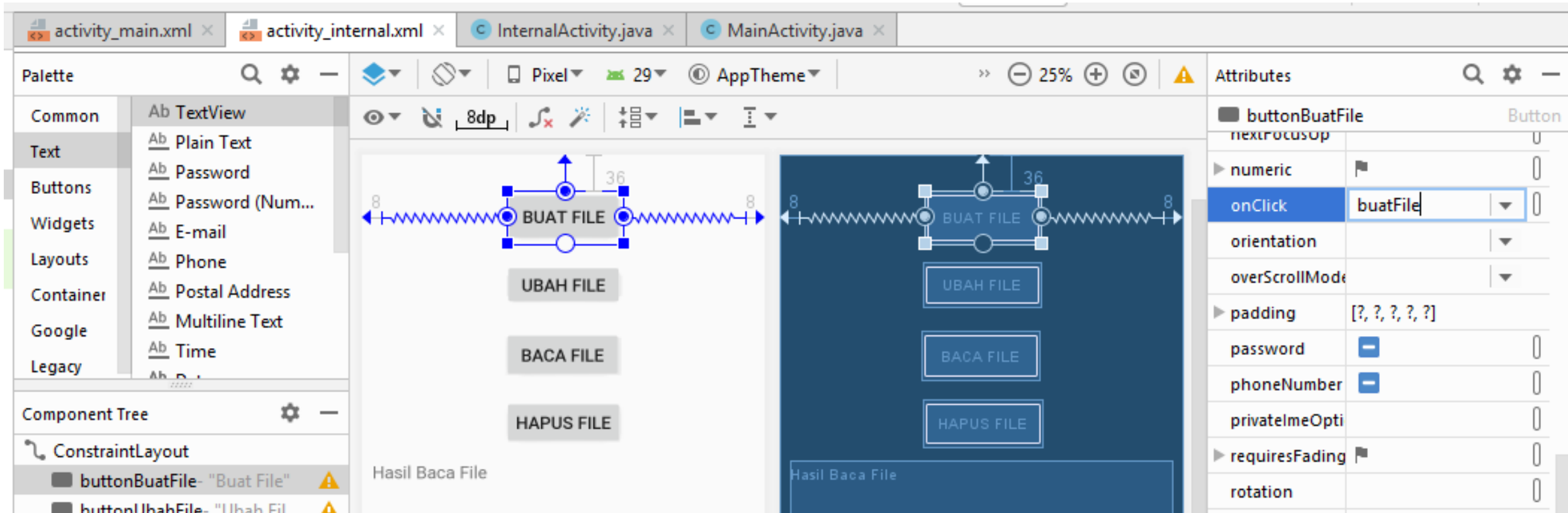
layout_height0dp

visibility

Latihan 1 – Langkah 4

Pelatihan

[Langkah 4] Pada Java Kelas `InternalActivity.java` Tambahkan baris fungsi `buatFile`, `ubahFile`, `bacaFile`, dan `hapusFile`.



Latihan 1 – Langkah 4

Pelatihan

The screenshot shows an IDE window with an Android XML layout. The XML code is as follows:

```
android:layout_marginRight="8dp"
android:onClick="buatFile"
```

A warning icon (red exclamation mark) is visible next to the `android:onClick="buatFile"` attribute. A context menu is open over the warning, listing the following options:

- Provide feedback on this warning
- Suppress: Add `tools:ignore="OnClick"` attribute
- Create 'buatFile(View)' in 'InternalActivity' (highlighted)
- Typo: Change to...
- Save 'buat' to project-level dictionary
- Create onClick event handler
- Override Resource in Other Configuration...
- Rearrange tag attributes
- Remove attribute
- Inject language or reference

On the right side of the IDE, a snippet of XML code is visible:

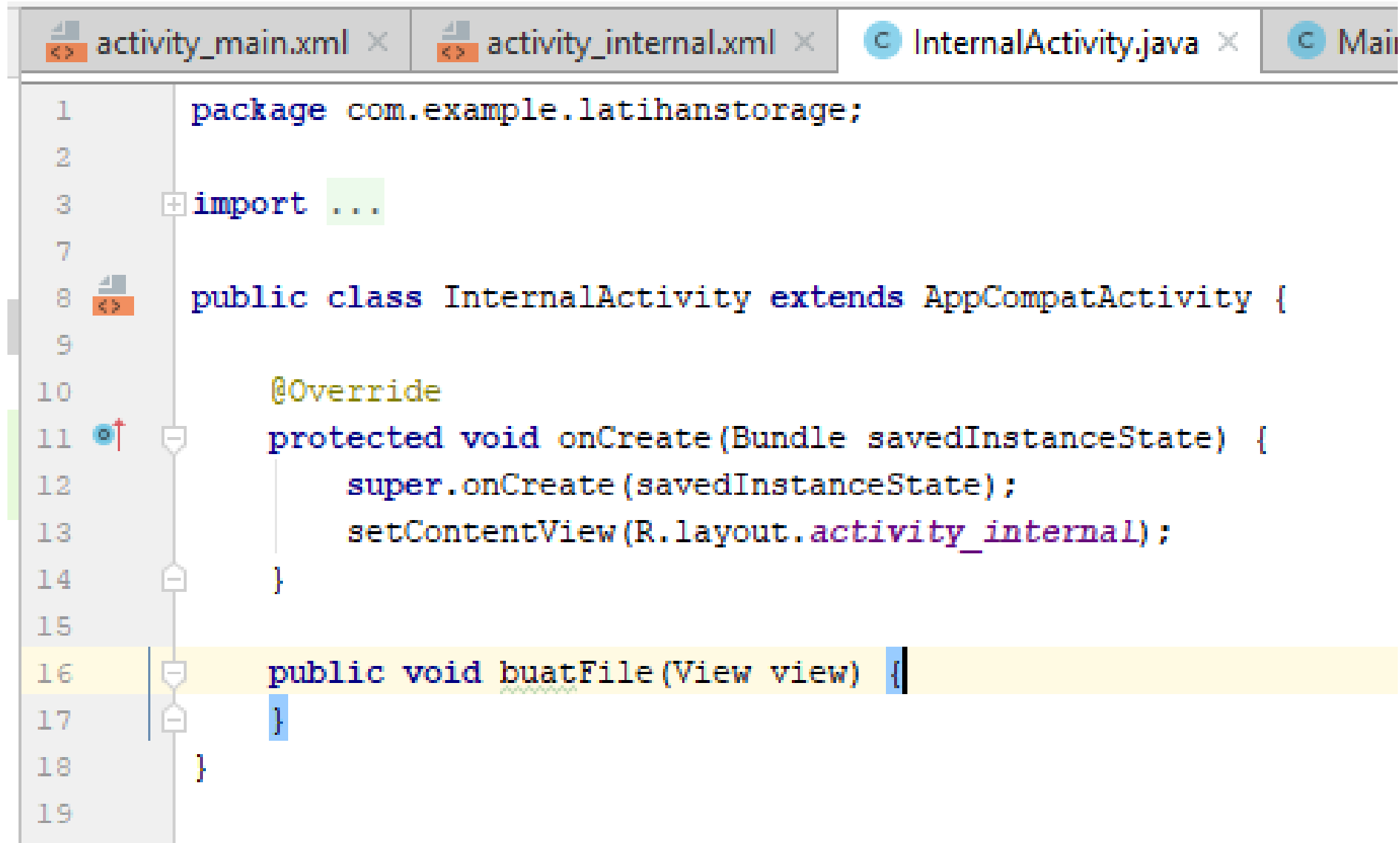
```
"parent"
ias="0.501"
tOf="parent"
"parent" />
```

Below the code snippet, a table with one row is visible:

Button

Latihan 1 – Langkah 4

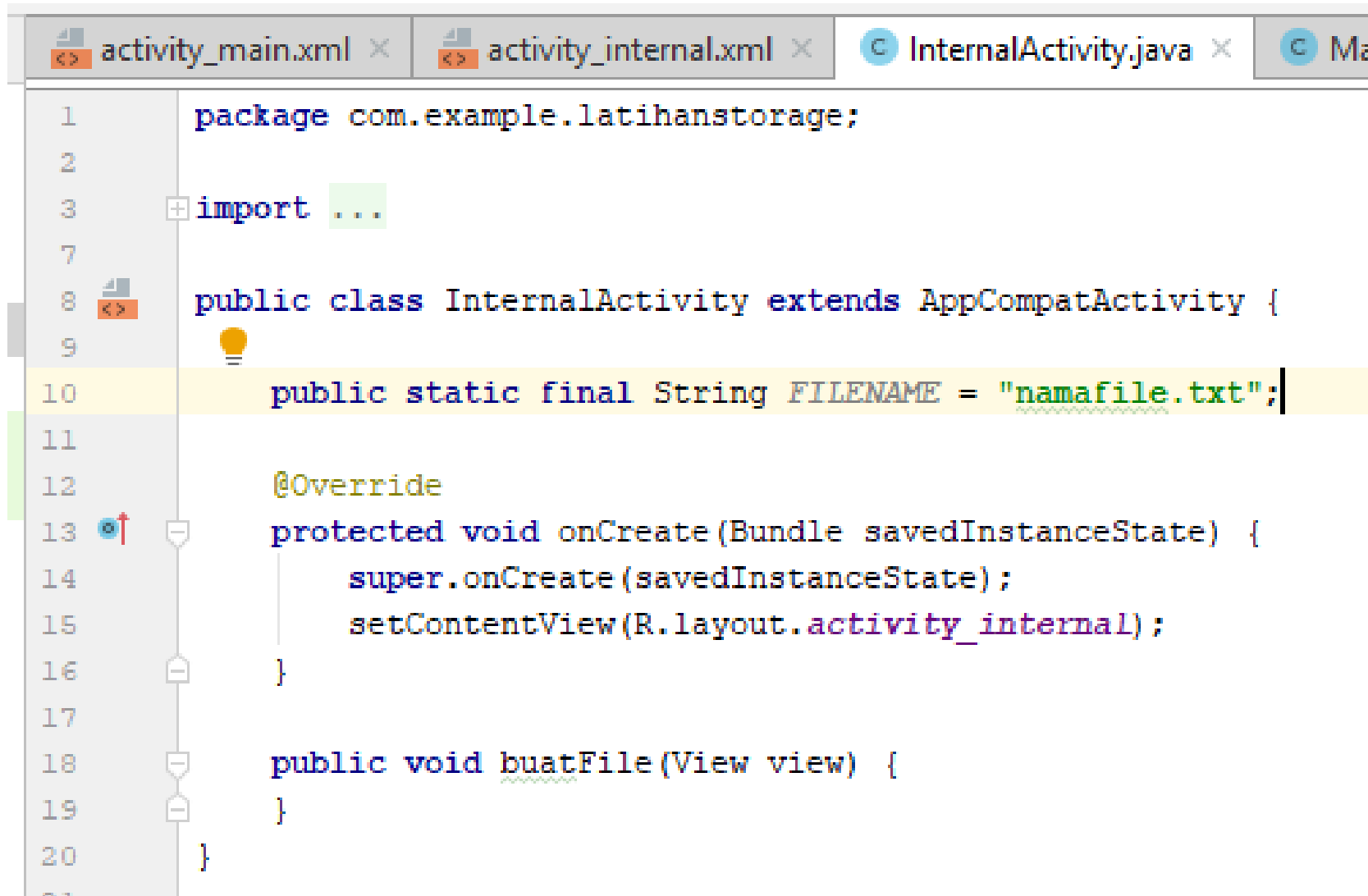
Pelatihan



```
activity_main.xml x activity_internal.xml x InternalActivity.java x Main
1 package com.example.latihanstorage;
2
3 import ...
7
8 public class InternalActivity extends AppCompatActivity {
9
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_internal);
14     }
15
16     public void buatFile(View view) {
17     }
18 }
19
```

Latihan 1 – Langkah 4

Pelatihan



```
activity_main.xml x activity_internal.xml x InternalActivity.java x Ma
1 package com.example.latihanstorage;
2
3 import ...
7
8 public class InternalActivity extends AppCompatActivity {
9
10     public static final String FILENAME = "namafile.txt";
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_internal);
16     }
17
18     public void buatFile(View view) {
19     }
20 }
```

Latihan 1 – Langkah 4

Pelatihan

```
public class InternalActivity extends AppCompatActivity {  
  
    public static final String FILENAME = "namafile.txt";  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_internal);  
    }  
  
    public void buatFile(View view) {  
        String isiFile = "Coba Isi Data File Text";  
    }  
}
```

Latihan 1 – Langkah 4

Pelatihan

```
public class InternalActivity extends AppCompatActivity {  
  
    public static final String FILENAME = "namafile.txt";  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_internal);  
    }  
  
    private void writeFile(View view) {  
        String isiFile = "Coba Isi Data File Text";  
        File  
    }  
}
```

? java.io.File? Alt+Enter

Latihan 1 – Langkah 4

Pelatihan

```
public class InternalActivity extends AppCompatActivity {  
  
    public static final String FILENAME = "namafile.txt";  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_internal);  
    }  
  
    public void buatFile(View view) {  
        String isiFile = "Coba Isi Data File Text";  
        File file = new File(getFilesDir(), FILENAME);  
    }  
}
```

Latihan 1 – Langkah 4

Pelatihan

```
10 public class InternalActivity extends AppCompatActivity {  
11  
12     public static final String FILENAME = "namafile.txt";  
13  
14     @Override  
15     protected void onCreate(Bundle savedInstanceState) {  
16         super.onCreate(savedInstanceState);  
17         setContentView(R.layout.activity_internal);  
18     }  
19  
20     public void buatFile(View view) {  
21         ? java.io.FileOutputStream? Alt+Enter "Coba Isi Data File Text";  
22         File file = new File(getFilesDir(), FILENAME);  
23         FileOutputStream  
24     }  
25 }
```


Latihan 1 – Langkah 4

Pelatihan

```
public class InternalActivity extends AppCompatActivity {  
  
    public static final String FILENAME = "namafile.txt";  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_internal);  
    }  
  
    public void buatFile(View view) {  
        String isiFile = "Coba Isi Data File Text";  
        File file = new File(getFilesDir(), FILENAME);  
        FileOutputStream outputStream = null;  
    }  
}
```

Latihan 1 – Langkah 4

Pelatihan

```
public void buatFile(View view) {  
    String isiFile = "Coba Isi Data File Text";  
    File file = new File(getFilesDir(), FILENAME);  
    FileOutputStream outputStream = null;  
    try {  
        file.createNewFile();  
        outputStream = new FileOutputStream(file, append: true);  
        outputStream.write(isiFile.getBytes());  
        outputStream.flush();  
        outputStream.close();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

Latihan 1 – Langkah 4

Pelatihan

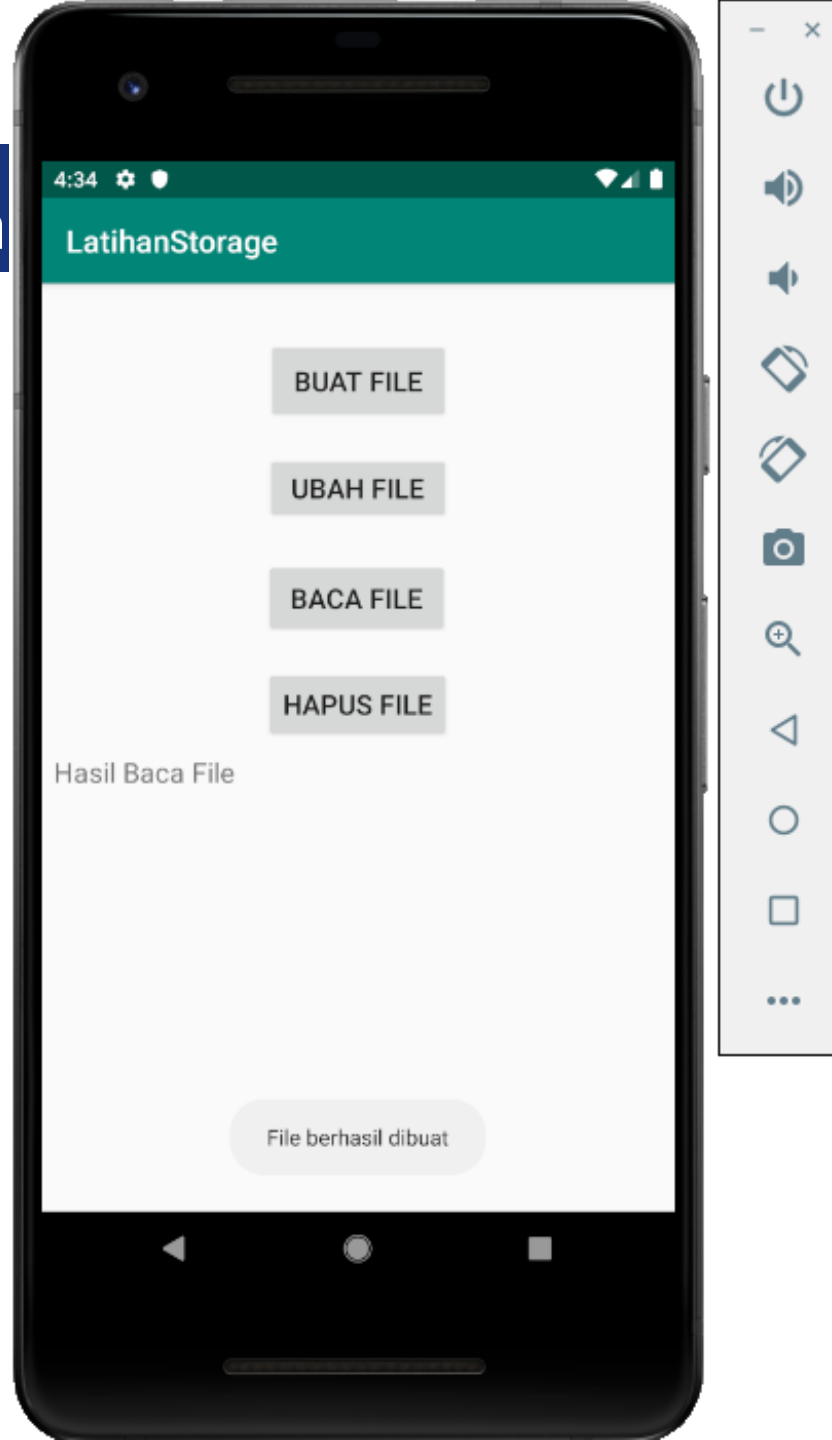
```
public void buatFile(View view) {
    String isiFile = "Coba Isi Data File Text";
    File file = new File(getFilesDir(), FILENAME);
    FileOutputStream outputStream = null;
    try {
        file.createNewFile();
        outputStream = new FileOutputStream(file, append: true);
        outputStream.write(isiFile.getBytes());
        outputStream.flush();
        android.widget.Toast.makeText(getApplicationContext(), "File Created", Toast.LENGTH_SHORT).show();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Latihan 1 – Langkah 4

Pelatihan

```
public void buatFile(View view) {  
    String isiFile = "Coba Isi Data File Text";  
    File file = new File(getFilesDir(), FILENAME);  
    FileOutputStream outputStream = null;  
    try {  
        file.createNewFile();  
        outputStream = new FileOutputStream(file, append: true);  
        outputStream.write(isiFile.getBytes());  
        outputStream.flush();  
        outputStream.close();  
  
        Toast.makeText(getApplicationContext(), text: "File berhasil dibuat", Toast.LENGTH_SHORT).show();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

Latihan 1 – Langkah



Pelatihan

Latihan 1 –

View | Navigate | Code | Analyze | Refactor | Build | Run | Tools | VCS | Window

Tool Windows

- Quick Definition Ctrl+Shift+I
- Show Siblings
- Quick Documentation Ctrl+Q
- Parameter Info Ctrl+P
- Expression Type Ctrl+Shift+P
- Context Info Alt+Q
- Recent Files Ctrl+E
- Recently Changed Files Ctrl+Shift+E
- Recent Changes Alt+Shift+C
- Compare With... Ctrl+D
- Compare with Clipboard
- Quick Switch Scheme... Ctrl+`
- Toolbar
- ✓ Tool Buttons
- ✓ Status Bar
- ✓ Navigation Bar
- Active Editor
- Bidi Text Direction
- Enter Presentation Mode
- Enter Distraction Free Mode
- Enter Full Screen

Tool Windows

- Project Alt+1
- ★ Favorites Alt+2
- ▶ Run Alt+4
- ⚙ Debug Alt+5
- ≡ Logcat Alt+6
- ▦ Structure Alt+7
- ↕ Version Control Alt+9
- 🔍 Profiler
- ⬅ Build
- 🏗 Build Variants
- 🔍 Capture Analysis
- ⚙ Capture Tool
- 🖼 Layout Captures
- 🏗 Designer
- Device File Explorer**
- 🟢 Event Log
- 🐘 Gradle
- ↕ Image Layers
- 🎨 Palette
- 👁 Preview
- 🏗 Resource Manager
- 📄 Terminal Alt+F12
- 🌈 Theme Preview

Pelatihan

Latihan 1

Device File Explorer

Emulator Pixel_2_API_29 Android 10, API 29

Name	Permissions	Date	Size
▶ folder acct	dr-xr-xr-x	2019-07-17 03:57	0 B
▶ folder apex	drwxr-xr-x	2019-07-17 03:57	320 B
▶ folder bin	lrw-r--r--	2009-01-01 07:00	11 B
▶ folder cache	drwxrwx---	2009-01-01 07:00	4 KB
▶ folder config	drwxr-xr-x	2019-07-17 03:57	0 B
▶ folder d	lrw-r--r--	2009-01-01 07:00	17 B
▼ folder data	drwxrwx--x	2019-07-03 05:27	4 KB
▶ folder app	drwxrwx--x	2019-07-03 05:27	4 KB
▶ folder data	drwxrwx--x	2019-07-03 05:27	4 KB
▶ folder local	drwxrwx--x	2019-07-03 05:27	4 KB
▶ folder debug_ramdisk	drwxr-xr-x	2009-01-01 07:00	4 KB
▶ folder dev	drwxr-xr-x	2019-07-17 03:57	1.3 KB
▶ folder etc	lrw-r--r--	2009-01-01 07:00	11 B
▶ folder lost+found	drwx-----	2009-01-01 07:00	16 KB
▶ folder mnt	drwxr-xr-x	2019-07-17 03:57	260 B
▶ folder odm	drwxr-xr-x	2009-01-01 07:00	4 KB
▶ folder oem	drwxr-xr-x	2009-01-01 07:00	4 KB
▶ folder proc	dr-xr-xr-x	2019-07-17 03:57	0 B
▶ folder product	lrw-r--r--	2009-01-01 07:00	15 B
▶ folder res	drwxr-xr-x	2009-01-01 07:00	4 KB
▶ folder sbin	drwxr-x---	2009-01-01 07:00	4 KB
▶ folder sdcard	lrw-r--r--	2009-01-01 07:00	21 B

Pelatihan

Latihan 1

Device File Explorer

Emulator Pixel_2_API_29 Android 10, API 29

Name	Permissions	Date	Size
▼ data	drwxrwx--x	2019-07-03 05:27	4 KB
▶ android	drwxrwx--x	2019-07-03 05:27	4 KB
▶ com.android.apps.tag	drwxrwx--x	2019-07-03 05:27	4 KB
▶ com.android.backupconfirm	drwxrwx--x	2019-07-03 05:27	4 KB
▶ com.android.bips	drwxrwx--x	2019-07-03 05:27	4 KB
▶ com.android.bluetooth	drwxrwx--x	2019-07-03 05:27	4 KB
▶ com.android.bluetoothmidiservic	drwxrwx--x	2019-07-03 05:27	4 KB
▶ com.android.bookmarkprovider	drwxrwx--x	2019-07-03 05:27	4 KB
▶ com.android.calllogbackup	drwxrwx--x	2019-07-03 05:27	4 KB
▶ com.android.camera2	drwxrwx--x	2019-07-03 05:27	4 KB
▶ com.android.captiveportallogin	drwxrwx--x	2019-07-03 05:27	4 KB
▶ com.android.carrierconfig	drwxrwx--x	2019-07-03 05:27	4 KB
▶ com.android.carrierdefaultapp	drwxrwx--x	2019-07-03 05:27	4 KB
▶ com.android.cellbroadcastreceiv	drwxrwx--x	2019-07-03 05:27	4 KB
▶ com.android.certinstaller	drwxrwx--x	2019-07-03 05:27	4 KB
▶ com.android.chrome	drwxrwx--x	2019-07-03 05:27	4 KB
▶ com.android.companiondevicem	drwxrwx--x	2019-07-03 05:27	4 KB
▶ com.android.contacts	drwxrwx--x	2019-07-03 05:27	4 KB
▶ com.android.cts.ctsshim	drwxrwx--x	2019-07-03 05:27	4 KB
▶ com.android.cts.priv.ctsshim	drwxrwx--x	2019-07-03 05:27	4 KB
▶ com.android.defcontainer	drwxrwx--x	2019-07-03 05:27	4 KB
▶ com.android.documentsui	drwxrwx--x	2019-07-03 05:27	4 KB

Gradle

Pelatihan

Latihan

Device File Explorer				
Emulator Pixel_2_API_29 Android 10, API 29				
Name	Permissions	Date	Size	
▶ com.android.vpndialogs	drwxrwx--x	2019-07-03 05:27	4 KB	
▶ com.android.wallpaper.livepicker	drwxrwx--x	2019-07-03 05:27	4 KB	
▶ com.android.wallpaperbackup	drwxrwx--x	2019-07-03 05:27	4 KB	
▶ com.breel.wallpapers18	drwxrwx--x	2019-07-03 05:27	4 KB	
▶ com.example.aplikasilistview	drwxrwx--x	2019-07-03 05:27	4 KB	
▶ com.example.aplikasipertamaanc	drwxrwx--x	2019-07-03 05:27	4 KB	
▶ com.example.latihan1luhurpidek	drwxrwx--x	2019-07-03 05:27	4 KB	
▼ com.example.latihanstorage	drwxrwx--x	2019-07-03 05:27	4 KB	
cache	drwxrws--x	2019-07-17 04:00	4 KB	
code_cache	drwxrws--x	2019-07-17 04:00	4 KB	
▼ files	drwxrwx--x	2019-07-17 05:25	4 KB	
namafile.txt	-rw-----	2019-07-17 06:00	48 B	
▶ com.google.android.angle	drwxrwx--x	2019-07-03 05:27	4 KB	
▶ com.google.android.apps.diagno	drwxrwx--x	2019-07-03 05:27	4 KB	
▶ com.google.android.apps.docs	drwxrwx--x	2019-07-03 05:27	4 KB	
▶ com.google.android.apps.inputtr	drwxrwx--x	2019-07-03 05:27	4 KB	
▶ com.google.android.apps.maps	drwxrwx--x	2019-07-03 05:27	4 KB	
▶ com.google.android.apps.messa	drwxrwx--x	2019-07-03 05:27	4 KB	
▶ com.google.android.apps.nexusl	drwxrwx--x	2019-07-03 05:27	4 KB	
▶ com.google.android.apps.photos	drwxrwx--x	2019-07-03 05:27	4 KB	
▶ com.google.android.apps.pixelm	drwxrwx--x	2019-07-03 05:27	4 KB	
▶ com.google.android.apps.restore	drwxrwx--x	2019-07-03 05:27	4 KB	

Pelatihan

Latihan 1 – Langkah 4

Pelatihan

InternalActivity.java x MainActivity.java x

Pixel 29 AppTheme 22%

8dp

BUAT FILE
UBAH FILE
BACA FILE
HAPUS FILE

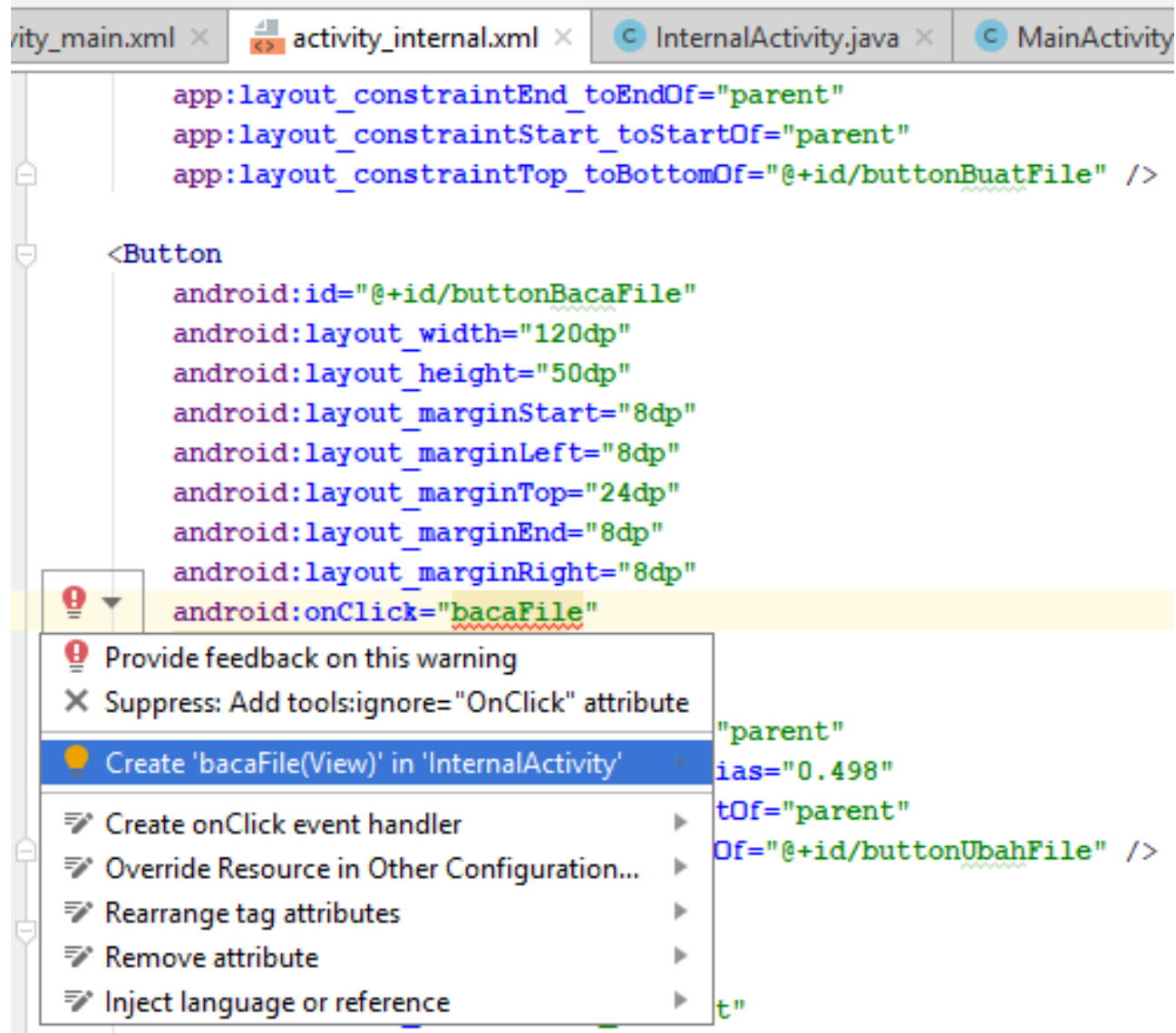
BUAT FILE
UBAH FILE
BACA FILE
HAPUS FILE

Attributes

buttonBacaFile	Button
nextFocusForward	0
nextFocusLeft	0
nextFocusRight	0
nextFocusUp	0
numeric	0
onClick	bacaFile
orientation	0
overScrollMode	0

Latihan 1 – Langkah 4

Pelatihan



The screenshot shows an IDE with four tabs: `activity_main.xml`, `activity_internal.xml`, `InternalActivity.java`, and `MainActivity`. The `activity_internal.xml` file is active, displaying the following XML code:

```
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/buttonBuatFile" />

<Button
    android:id="@+id/buttonBacaFile"
    android:layout_width="120dp"
    android:layout_height="50dp"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="24dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:onClick="bacaFile" />
```

A warning icon is visible next to the `android:onClick="bacaFile"` attribute. A context menu is open, showing the following options:

- Provide feedback on this warning
- Suppress: Add tools:ignore="OnClick" attribute
- Create 'bacaFile(View)' in 'InternalActivity'
- Create onClick event handler
- Override Resource in Other Configuration...
- Rearrange tag attributes
- Remove attribute
- Inject language or reference

Latihan 1 – Langkah 4

Pelatihan

```
public void bacaFile(View view) {  
    }  
}
```

```
public void bacaFile(View view) {  
    try{  
        FileInputStream fileInputStream = openFileInput(FILENAME);  
        InputStreamReader inputStreamReader = new InputStreamReader(fileInputStream);  
  
        BufferedReader bufferedReader = new BufferedReader(inputStreamReader);  
        StringBuffer stringBuffer = new StringBuffer();  
  
        String baris;  
        while ((baris = bufferedReader.readLine()) != null) {  
            stringBuffer.append(baris + "\n");  
        }  
        TextView hasil = findViewById(R.id.textBaca);  
        hasil.setText(stringBuffer.toString());  
  
    } catch (FileNotFoundException e) {  
        e.printStackTrace();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

Latihan 1 – Langkah 4

Pelatihan

```
public void ubahFile(View view) {  
    String ubah = "Update Isi Data File Text";  
    File file = new File(getFilesDir(), FILENAME);  
    FileOutputStream outputStream = null;  
    try {  
        file.createNewFile();  
        outputStream = new FileOutputStream(file, append: true);  
        outputStream.write(ubah.getBytes());  
        outputStream.flush();  
        outputStream.close();  
  
        Toast.makeText(getApplicationContext(), text: "File Text berhasil diubah", Toast.LENGTH_SHORT).show();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

Latihan 1 – Langkah 4

Pelatihan

```
public void hapusFile(View view) {  
    File file = new File(getFilesDir(), FILENAME);  
    if (file.exists()) {  
        file.delete();  
        Toast.makeText(getApplicationContext(), text "File berhasil dihapus", Toast.LENGTH_SHORT).show();  
        TextView hasil = findViewById(R.id.textBaca);  
        hasil.setText("");  
    }  
}
```

Latihan 2

Pelatihan

- **[Langkah 2]** Buat activity ExternalActivity yang akan memunculkan halaman baru yang memiliki tampilan dan fungsi yang sama dengan internal storage (yang membedakan hanya lokasi penyimpanan menggunakan *external storage*)

Latihan 2

Pelatihan

The screenshot displays the Android Studio IDE with the following components:

- Project View (Left):** Shows the project structure for 'app'. It includes a 'manifests' folder containing 'AndroidManifest.xml', a 'java' folder containing the package 'com.example.latihanstorage' with classes 'InternalActivity' and 'MainActivity', and other folders like 'generatedJava', 'res', and 'Gradle Scripts'.
- Toolbar (Top):** Contains standard Android Studio icons for running, debugging, and other actions.
- Code Editor (Right):** Displays the 'AndroidManifest.xml' file with the following XML code:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3       package="com.example.latihanstorage">
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="LatihanStorage"
9         android:roundIcon="@mipmap/ic_launcher_round"
10        android:supportsRtl="true"
11        android:theme="@style/AppTheme">
12         <activity android:name=".InternalActivity"></activity>
13         <activity android:name=".MainActivity">
14             <intent-filter>
15                 <action android:name="android.intent.action.MAIN" />
16
17                 <category android:name="android.intent.category.LAUNCHER" />
18             </intent-filter>
19         </activity>
20     </application>
21
22 </manifest>
```

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.latihanstorage">

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="LatihanStorage"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".ExternalActivity"></activity>
        <activity android:name=".InternalActivity" />
        <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>
    
```

Latihan 2

Pelatihan

```
public class ExternalActivity extends AppCompatActivity {  
  
    public static final String FILENAME = "namafile.txt";  
    public static final int REQUEST_CODE_STORAGE = 100;  
    public boolean mendapatIzin;  
    private String filepath = "MyFileStorage";  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_external);  
    }  
}
```

Latihan 2

Pelatihan

```
public boolean periksaIzinPenyimpanan() {  
    if (Build.VERSION.SDK_INT >= 23) {  
        if (checkSelfPermission(android.Manifest.permission.WRITE_EXTERNAL_STORAGE)  
            == PackageManager.PERMISSION_GRANTED) {  
            mendapatIzin = true;  
            return true;  
        } else {  
            ActivityCompat.requestPermissions( activity: this,  
                new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE},  
                REQUEST_CODE_STORAGE);  
            return false;  
        }  
    } else {  
        return true;  
    }  
}
```

Latihan 2

Pelatihan

```
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
                                       @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    switch (requestCode) {
        case REQUEST_CODE_STORAGE:
            if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                mendapatIzin = true;
            }
            break;
    }
}
```

Latihan 2

Pelatihan

```
public void buatFile(View view) {  
    if (periksaIzinPenyimpanan()) {  
        if (mendapatIzin == true) {  
            String isiFile = "Coba Isi Data File Text";  
            File file = new File(getExternalFilesDir(filepath), FILENAME);  
            FileOutputStream outputStream = null;  
            try {  
                file.createNewFile();  
                outputStream = new FileOutputStream(file, append: true);  
                outputStream.write(isiFile.getBytes());  
                outputStream.flush();  
                outputStream.close();  
                Toast.makeText(getApplicationContext(), text: "File berhasil dibuat", Toast.LENGTH_SHORT).show();  
            } catch (Exception e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

```
public void bacaFile(View view) {  
    if (periksaIzinPenyimpanan()) {  
        if (mendapatIzin == true) {  
            try {  
                File file = new File(getExternalFilesDir(filepath), FILENAME);  
  
                BufferedReader bufferedReader = new BufferedReader(new FileReader(file));  
                StringBuffer stringBuffer = new StringBuffer();  
  
                String baris;  
                while ((baris = bufferedReader.readLine()) != null) {  
                    stringBuffer.append(baris + "\n");  
                }  
                TextView hasil = findViewById(R.id.textBaca);  
                hasil.setText(stringBuffer.toString());  
  
            } catch (FileNotFoundException e) {  
                e.printStackTrace();  
            } catch (IOException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

Latihan 2

Name	Permissions	Date	Size
bin	lrw-r--r--	2009-01-01 07:00	11 B
cache	drwxrwx---	2009-01-01 07:00	4 KB
config	drwxr-xr-x	2019-07-18 03:40	0 B
d	lrw-r--r--	2009-01-01 07:00	17 B
data	drwxrwx--x	2019-07-03 05:27	4 KB
debug_ramdisk	drwxr-xr-x	2009-01-01 07:00	4 KB
dev	drwxr-xr-x	2019-07-18 03:40	1.3 KB
etc	lrw-r--r--	2009-01-01 07:00	11 B
lost+found	drwx-----	2009-01-01 07:00	16 KB
mnt	drwxr-xr-x	2019-07-18 03:40	260 B
odm	drwxr-xr-x	2009-01-01 07:00	4 KB
oem	drwxr-xr-x	2009-01-01 07:00	4 KB
proc	dr-xr-xr-x	2019-07-18 03:40	0 B
product	lrw-r--r--	2009-01-01 07:00	15 B
res	drwxr-xr-x	2009-01-01 07:00	4 KB
sbin	drwxr-x---	2009-01-01 07:00	4 KB
storage	drwxr-xr-x	2019-07-18 03:40	60 B
1DFA-3702	drwxrwx--x	1970-01-01 07:00	2 KB
emulated	drwx--x--x	2019-07-03 05:29	4 KB
self	drwxr-xr-x	2019-07-18 03:40	60 B
primary	lrwxrwxrwx	2019-07-18 03:40	19 B
Alarms	drwxrwx--x	2019-07-03 05:31	4 KB
Android	drwxrwx--x	2019-07-03 05:29	4 KB
data	drwxrwx--x	2019-07-18 02:44	4 KB
com.android.vending	drwxrwx--x	2019-07-05 02:19	4 KB
com.example.latihanstorage	drwxrwx--x	2019-07-18 02:44	4 KB
files	drwxrwx--x	2019-07-18 02:44	4 KB
MyFileStorage	drwxrwx--x	2019-07-18 02:44	4 KB
namafile.txt	-rw-rw----	2019-07-18 02:54	69 B
com.google.android.apps.docs	drwxrwx--x	2019-07-05 02:18	4 KB
com.google.android.apps.maps	drwxrwx--x	2019-07-05 02:16	4 KB
com.google.android.gms	drwxrwx--x	2019-07-03 05:29	4 KB

Pelatihan

Latihan 2

Pelatihan

```
public void ubahFile(View view) {  
    if (periksaIzinPenyimpanan()) {  
  
        if(mendapatIzin == true) {  
            String isiFile = "Ubah Data File Text";  
            File file = new File(getExternalFilesDir(filepath), FILENAME);  
            FileOutputStream outputStream = null;  
            try {  
                file.createNewFile();  
                outputStream = new FileOutputStream(file, append: true);  
                outputStream.write(isiFile.getBytes());  
                outputStream.flush();  
                outputStream.close();  
                Toast.makeText(getApplicationContext(), text: "File berhasil diubah", Toast.LENGTH_SHORT).show();  
            } catch (Exception e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

Latihan 2

Pelatihan

```
public void hapusFile(View view) {  
    if (periksaIzinPenyimpanan()) {  
        if (mendapatIzin == true) {  
            File file = new File(getExternalFilesDir(filepath), FILENAME);  
            if (file.exists()) {  
                file.delete();  
                Toast.makeText(getApplicationContext(), text: "File berhasil dihapus", Toast.LENGTH_SHORT).show();  
                TextView hasil = findViewById(R.id.textBaca);  
                hasil.setText("");  
            }  
        }  
    }  
}
```

Proyek aplikasi sederhana penyimpan data *persistence*

- **[Proyek 1]** : Aplikasi Catatan Harian dengan menggunakan file pada Android
- **[Proyek 2]** : Aplikasi Validasi Login yang melakukan pengecekan terhadap data user yang tersimpan pada file Teks.

REFERENSI

1. <http://bit.ly/2WaH2bI>
2. <https://developer.android.com/reference/android/os/Environment.html>
3. <https://developer.android.com/reference/java/io/File.html>
4. <https://developer.android.com/guide/topics/data/data-storage?hl=id>
5. <https://developer.android.com/training/data-storage/files#InternalVsExternalStorage>

Tim Penyusun:

- Alif Akbar Fitrawan, S.Pd, M. Kom (Politeknik Negeri Banyuwangi);
- Anwar, S.Si, MCs. (Politeknik Negeri Lhokseumawe);
- Eddo Fajar Nugroho (BPPTIK Cikarang);
- Eddy Tungadi, S.T., M.T. (Politeknik Negeri Ujung Pandang);
- Fitri Wibowo (Politeknik Negeri Pontianak);
- Ghifari Munawar (Politeknik Negeri Bandung);
- Hetty Meileni, S.Kom., M.T. (Politeknik Negeri Sriwijaya) ;
- I Wayan Candra Winetra, S.Kom., M.Kom (Politeknik Negeri Bali) ;
- Irkham Huda (Vokasi UGM) ;
- Josseano Amakora Koli Parera, S.Kom., M.T. (Politeknik Negeri Ambon) ;
- I Komang Sugiarta, S.Kom., MMSI (Universitas Gunadarma) ;
- Lucia Sri Istiyowati, M.Kom (Institut Perbanas) ;
- Maksy Sendiang, ST, MIT (Politeknik Negeri Manado) ;
- Medi Noviana (Universitas Gunadarma) ;
- Muhammad Nashrullah (Politeknik Negeri Batam) ;
- Nat. I Made Wiryana, S.Si., S.Kom., M.Sc. (Universitas Gunadarma) ;
- Rika Idmayanti, ST, M.Kom (Politeknik Negeri Padang) ;
- Rizky Yuniar Hakkun (Politeknik Elektronik Negeri Surabaya) ;
- Robinson A.Wadu, ST, MT (Politeknik Negeri Kupang) ;
- Roslina. M.IT (Politeknik Negeri Medan) ;
- Sukamto, SKom., MT. (Politeknik Negeri Semarang) ;
- Syamsi Dwi Cahya, M.Kom. (Politeknik Negeri Jakarta) ;
- Syamsul Arifin, S.Kom, M.Cs (Politeknik Negeri Jember) ;
- Usmanudin (Universitas Gunadarma) ;
- Wandy Alifha Saputra (Politeknik Negeri Banjarmasin) ;