

Mobile Program

Pertemuan 4

- Komposisi Pemrograman
- Antarmuka Android

Android Studio

Komposisi Pemrograman

- Pemrograman Tampilan
- Pemrograman Fungsional

Pemrograman Tampilan

- Di bagian ini diharuskan membuat tampilan dari aplikasi yang dibuat
- Tanpa tampilan ini pembuatan fungsi aplikasi akan sulit dilakukan

Cont'd

- Pembuatan tampilan bisa dilakukan dengan cara menggunakan mode desainer, maupun secara manual menggunakan XML
- Maupun bahasa pemrograman lainnya yang memungkinkan untuk membuat tampilan

Cont'd

A screenshot of the Android Studio interface showing the code editor and the preview window.

The code editor (A) displays the XML layout file `activity_main.xml`:

```
<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" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

    <TextView android:text="Hello world!" android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

The preview window (B) shows a smartphone screen displaying the text "Hello world!".

The top bar of the interface includes tabs for `MainActivity.java` and `activity_main.xml`, and a `Preview` tab. A red circle labeled "C" is positioned above the preview window. A red circle labeled "D" is positioned at the bottom left of the interface.

Activity

- Suatu Activity adalah item tunggal dan terfokus yang dapat dilakukan pengguna.
- Hampir semua aktivitas berinteraksi dengan pengguna, sehingga kelas Activity mengatur pembuatan jendela

Activity

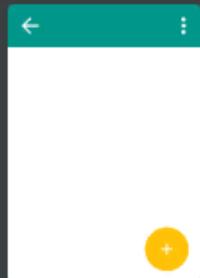
- Android menggunakan Activity sebagai kanvas membuat tampilan aplikasi. Sehingga terdapat beberapa template yang dapat digunakan.
 - Blank
 - Map
 - ListView
 - dll



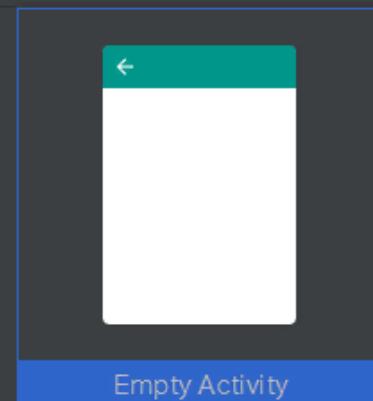
Add an activity to Mobile



Add No Activity



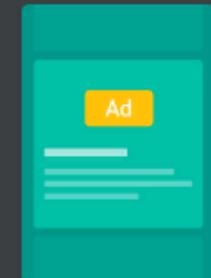
Blank Activity



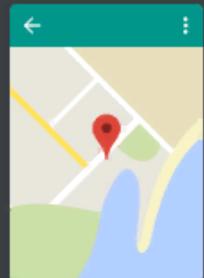
Empty Activity



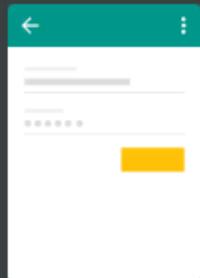
Fullscreen Activity



Google AdMob Ads Activity



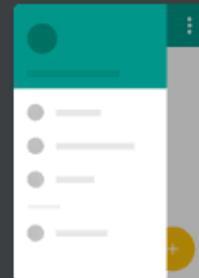
Google Maps Activity



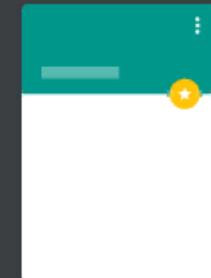
Login Activity



Master/Detail Flow



Navigation Drawer Activity



Scrolling Activity



Previous

Next

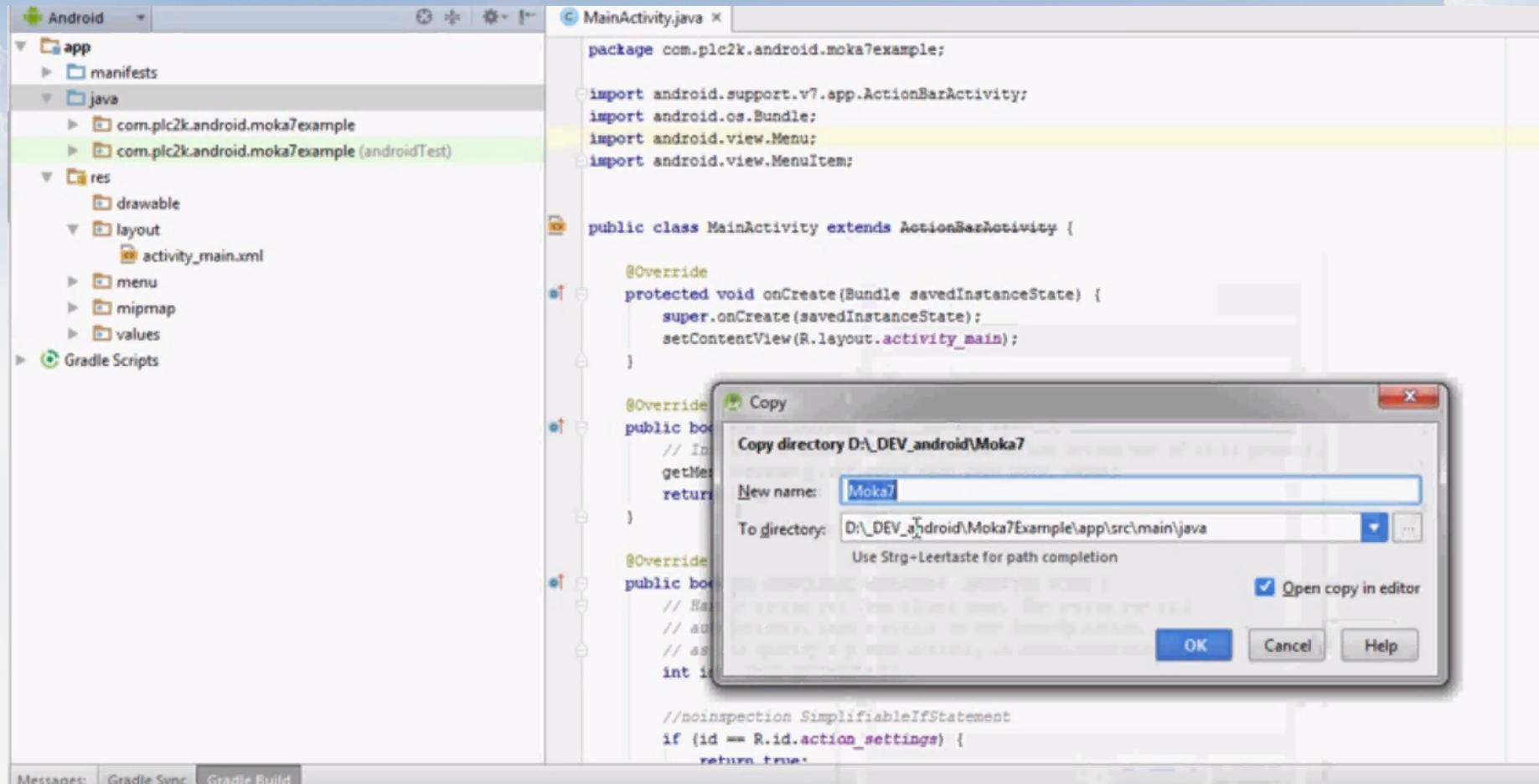
Cancel

Finish

Pemrograman Fungsional

- Di bagian ini pengembang diharuskan membuat fungsionalitas aplikasi yang ingin dibuat
- Dapat menggunakan Java maupun Kotlin sebagai dasar pemrograman

Dengan Java



Dengan Kotlin

Entry point to
Kotlin Program

KotlinFile.kt

```
fun main(args: Array<String>){  
    // main function body  
}  
  
fun otherFunction(args){  
    // function body  
}  
  
class className(){  
    // class body  
}
```

Kotlin vs Java

- Referensi kosong dikontrol oleh sistem tipe.
- Tidak ada jenis mentah
- Array di Kotlin tidak berubah
- Kotlin memiliki tipe fungsi yang tepat, berbeda dengan konversi SAM di Java
- Varians penggunaan-situs tanpa wildcard
- Kotlin tidak memiliki pengecekan pengecualian

Cont'd

Java

```
public class User {  
    private final String firstName;  
    private final String lastName;  
    private final int age;  
  
    public User(String firstName, String lastName, int age) {  
        this.firstName = firstName;  
        this.lastName = lastName;  
        this.age = age;  
    }  
  
    public String getFirstName() {  
        return firstName;  
    }  
  
    public String getLastName() {  
        return lastName;  
    }  
  
    public int getAge() {  
        return age;  
    }  
  
    public String toString() {  
        return firstName + " " + lastName + ", age " + age;  
    }  
}
```

```
class Main {  
    public static void main(String[] args) {  
        System.out.println(new User("John", "Doe", 30));  
    }  
}
```

Kotlin

```
public class User(val firstName: String,  
                 val lastName: String,  
                 val age: Int) {  
  
    fun toString() = "$firstName $lastName, age $age"  
}
```

```
fun main(args : Array<String>) {  
    println(User("John", "Doe", 30))  
}
```

Cont'd

- Agar Android dapat berjalan dengan baik, kode Fungsional harus dapat membaca Interface dari kode XML
- Data yang diambil kemudian dapat diproses oleh kode fungsional nantinya

Contoh

The screenshot shows the Android Studio interface with a project named "MyFirstAndroidApp". The left sidebar displays the project structure, including the build.gradle file, Java source code in MainActivity.java, and resource files like activity_main.xml and colors.xml. The main editor window shows the Java code for MainActivity, which contains code to set a green text color for a TextView. The bottom status bar indicates a successful compilation. On the right, the Android emulator displays the app's interface with the text "I love you" highlighted in green.

```
package com.example.myfirstandroidapp;

import ...

public class MainActivity extends Activity {

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

        TextView textElement = (TextView) findViewById(R.id.this_is);
        textElement.setText("I love you");

        textElement.setTextColor(0xFF00FF00); //this is green color
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

My First Android App

I love you

5554:Smartphone_4.0_480_800

3G 3:23

Basic Controls

Hardware Buttons

MENU

DPAD not enabled in AVD

Hardware Keyboard

Use your physical keyboard

Compilation completed successfully in 6 sec (a minute ago)

Tambahan

- Selain itu terdapat beberapa cara untuk membuat tampilan aplikasi + fungsionalitas sekaligus
- Hal ini dimaksudkan untuk mengurangi waktu pembuatan aplikasi yang intensif
 - Cth: Games

Kelebihan & Kekurangan

- Lebih cepat membuat aplikasi
- Pengembang tidak perlu membuat tampilan mereka sendiri
- Dapat dengan mengubah/kustom tampilan yang ada
- - Jika ingin merubah tampilan diharuskan mempunyai ahli di bidang UI/UX

FieldService.Android - ..\FieldService\Data\SampleAssignmentService.cs - Xamarin Studio

File Edit View Search Project Build Run Version Control Tools Window Help

Debug Default

Xamarin Studio

Press 'Control+, to search'

Solution

- AssignmentsActivity.cs
- AssignmentTabActivity.cs
- BaseActivity.cs
- BaseMapActivity.cs
- LoginActivity.cs
- MapFragmentActivity.cs
- MapViewActivity.cs
- SummaryActivity.cs
- SummaryHistoryActivity.cs
- Adapters
- Assets
- Dialogs
- FieldService
 - Data
 - Assignment.cs
 - AssignmentHistory.cs
 - AssignmentItem.cs
 - Database.cs
 - Document.cs
 - Enumerations.cs
 - Expense.cs
 - ExpensePhoto.cs
 - Item.cs
 - Labor.cs
 - Photo.cs
 - PropertyChangedBase.cs
 - SampleAssignmentService.cs
 - SampleLoginService.cs
 - Signature.cs
 - SQLite.cs
 - SQLiteAsync.cs
 - TestData.cs
 - TimerEntry.cs
 - Utilities
 - DataExtensions.cs

Document Outline

SampleAssignmentService

- SampleAssignmentService()
- List<Document> _documents
- Task<List<Assignment>> GetAssignmentsAsync
- Task<List<Item>> GetItemsAsync(CancellationToken cancellationToken)
- Task<List<AssignmentItem>> GetItemsForAssignmentAsync (Assignment assignment, CancellationToken cancellationToken)
- Task<List<Labor>> GetLaborForAssignmentAsync (Assignment assignment, CancellationToken cancellationToken)
- Task<List<Expense>> GetExpensesForAssignmentAsync (Assignment assignment, CancellationToken cancellationToken)
- Task<List<Photo>> GetPhotosForAssignmentAsync (Assignment assignment, CancellationToken cancellationToken)
- Task<int> SaveAssignmentAsync(Assignment assignment, CancellationToken cancellationToken)
- Task<Signature> GetSignatureAsync(Assignment assignment, CancellationToken cancellationToken)
- Task<int> SaveAssignmentItemAsync(AssignmentItem assignmentItem, CancellationToken cancellationToken)
- Task<int> SaveLaborAsync(Labor labor, CancellationToken cancellationToken)
- Task<int> SaveExpenseAsync(Expense expense, CancellationToken cancellationToken)
- Task<int> SavePhotoAsync(Photo photo, CancellationToken cancellationToken)
- Task<int> DeleteAssignmentAsync(Assignment assignment, CancellationToken cancellationToken)
- Task<int> DeleteAssignmentItemAsync(AssignmentItem assignmentItem, CancellationToken cancellationToken)
- Task<int> DeleteLaborAsync(Labor labor, CancellationToken cancellationToken)
- Task<int> DeleteExpenseAsync(Expense expense, CancellationToken cancellationToken)
- Task<int> DeletePhotoAsync(Photo photo, CancellationToken cancellationToken)
- Task<int> SaveTimerEntryAsync(TimerEntry timerEntry, CancellationToken cancellationToken)
- Task<int> DeleteTimerEntryAsync(TimerEntry timerEntry, CancellationToken cancellationToken)
- Task<TimerEntry> GetTimerEntryAsync(CancellationToken cancellationToken)
- Task<List<Document>> GetDocumentsAsync(CancellationToken cancellationToken)
- Task<List<AssignmentHistory>> GetAssignmentHistoriesAsync(CancellationToken cancellationToken)
- Task<List<Assignment>> GetAssignmentsFromDatabaseAsync(CancellationToken cancellationToken)
- Task<int> SaveExpensePhotoAsync(ExpensePhoto expensePhoto, CancellationToken cancellationToken)
- Task<ExpensePhoto> GetExpensePhotoAsync(ExpensePhoto expensePhoto, CancellationToken cancellationToken)

Properties

Unit Tests

Feedback Errors Deploy to Device Application Output

e X Chrome

16:08
20/02/2013

```

40
41     public Task<List<Item>> GetItemsAsync (CancellationToken cancellationToken = default(CancellationToken))
42     {
43         return Database.GetConnection (cancellationToken)
44             .Table<Item> ()
45             .OrderBy (i => i.Name)
46             .ToListAsync ();
47     }
48
49     public Task<List<AssignmentItem>> GetItemsForAssignmentAsync (Assignment assignment, CancellationToken cancellationToken)
50     {
51         return Database.GetConnection (cancellationToken)
52             .QueryAsync<AssignmentItem> (@"
53                 select AssignmentItem
54                 from Assignment
55                 where Assignment.Id == @assignmentId
56             ");
57     }
58
59     public Task<List<Labor>> GetLaborForAssignmentAsync (Assignment assignment, CancellationToken cancellationToken)
60     {
61         return Database.GetConnection (cancellationToken)
62             .Table<Labor> ()
63             .Where (l => l.AssignmentId == assignment.Id)
64             .ToListAsync ();
65     }
66
67     public Task<List<Expense>> GetExpensesForAssignmentAsync (Assignment assignment, CancellationToken cancellationToken)
68     {
69         return Database.GetConnection (cancellationToken)
70             .QueryAsync<Expense> (@"
71                 select Expense.*
72                 left outer join ExpensePhoto
73                     on ExpensePhoto.ExpenseId = Expense.Id
74                 where Expense.AssignmentId = ?",
75                     assignment.Id);
76     }
77
78

```

- Character
- AutoBlackToTitle.cs
- ClickToStart.cs
- Explosion.cs
- Explosive.cs
- Fire.cs
- FloorSection.cs
- GameControl.cs
- GameGUI.cs
- Hose.cs
- MapIcons.cs
- MessageGUI.cs
- MoveBetweenPoints
- player.cs
- Priority Particle Add.
- PriorityAlphaParticle
- SceneChanger.cs
- SmokeParticles.cs
- WaterHoseParticles
- WaterSplash.cs
- world.cs

```
50     vignette.blur = (1-health) * 2 + smokeEffect * 10 * distance;
51     vignette.blurDistance = (1-health) * 2 + smokeEffect * 10 * distance;
52     vignette.chromaticAberration = heatEffect * 10;
53 }
54
55
56 void OnTriggerStay(Collider c)
57 {
58     var fire = c.GetComponent<Fire>();
59     if (fire && fire.alive)
60     {
61         float dist = 1 - ((transform.position - fire.transform.position).magnitude);
62         NearHeat(dist);
63     }
64
65
66     var smoke = c.GetComponent<ClockstartToHealth>();
67     if (smoke && smoke.GetComponent<ParticleSystem>())
68     {
69         float dist = 1 - ((transform.position - smoke.transform.position).magnitude);
70         NearSmoke(dist);
71     }
72 }
73
74 void OnCollisionEnter(Collision c)
75 {
76     var healthBox = c.gameObject.GetComponent<HealthBox>();
77     if (healthBox)
78     {
79         if (healthBox.health > 0)
```