# Android Fundamental
## Fundamental & Navigation

bangkit

# Ground Rules

Observe the following rules to ensure a supportive, inclusive, and engaging classes

Give full attention in class

Mute your microphone when you're not talking

Keep your camera on


Turn on the CC Feature on Meet

Use raise hand or chat to ask questions

Make this room a safe place to learn and share

bangkit

# Material/Review

# Material that has been studied

- Learn how to **build your first app** with Android Studio.
- Learn about **basics** of Android Development such as Activity, Intent, Fragment, View, and ViewGroup.
- Learn how to **debug** and **resolving errors**.
- Learn how to build layout using **ConstraintLayout**.
- Learn how to design attractive applications using **Navigation** elements such as ActionBar, NavigationDrawer, BottomNavigation, and TabLayout.

# Activity

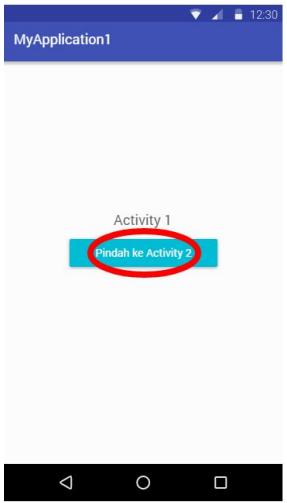
# Lifecycle Activities

onCreate
onStart
onResume
Activity Running

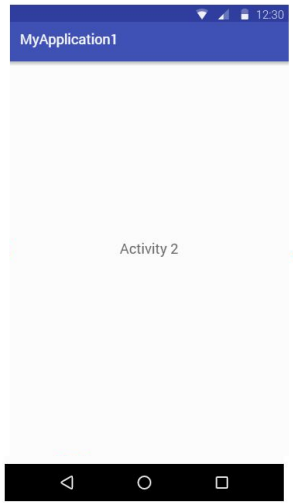
onPause
onStop
Activity Paused

onRestart
onStart
onResume
Activity Running

Activity 1




Activity 2



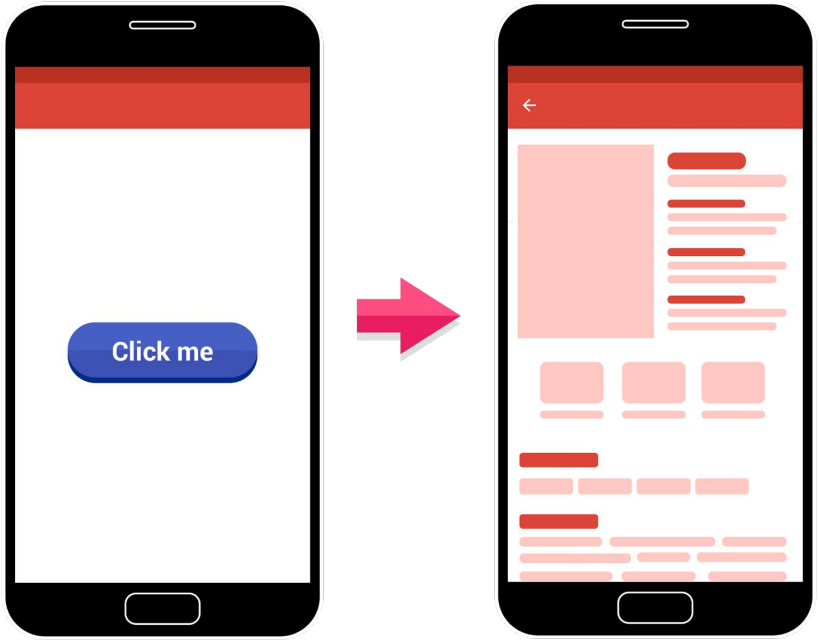

onCreate
onStart
onResume
Activity Running

onPause
onStop
onDestroy
Activity Shut Down

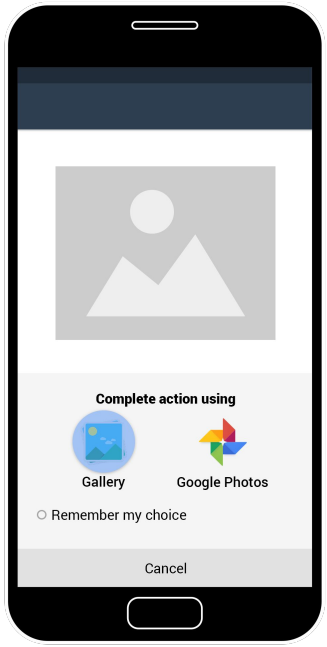
bangkit

# Intent

# Intent Implicit VS Intent Explicit

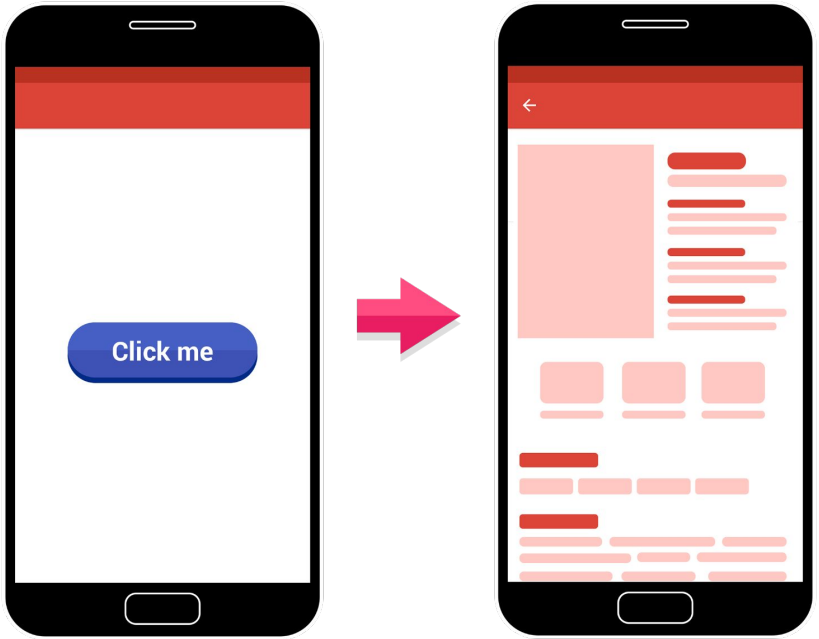



1




2

# Intent Implicit VS Intent Explicit



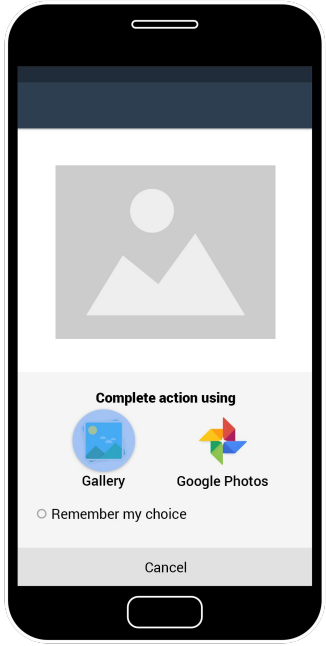

Intent Explicit

```
val moveIntent = Intent(
    this@MainActivity,
    DetailActivity::class.java
)
startActivity(moveIntent)
```

bangkit

# Intent Implicit VS Intent Explicit

```
val phoneNumber = "081000000000"
val dialPhoneIntent = Intent(
    Intent.ACTION_DIAL,
    Uri.parse("tel:$phoneNumber")
)
startActivity(dialPhoneIntent)
```

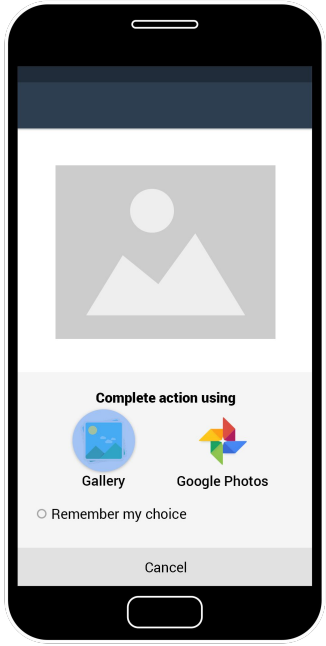



Intent Implicit

# Intent Implicit VS Intent Explicit

```
val sendIntent = Intent().apply {
    action = Intent.ACTION_SEND
    putExtra(Intent.EXTRA_TEXT, textMessage)
    type = "text/plain"
}

// Attempting to call an Intent
try{
    startActivity(sendIntent)
}catch (e: ActivityNotFoundException){
    // Handles when destination not found.
}
```




Intent Implicit

# Send Object using Parcelable

## Build.gradle (module: app)

```
plugins {
  id 'com.android.application'
  id 'kotlin-android'
  id 'kotlin-parcelize'
}
```

## Send Data

```
val intent = Intent(itemView.context,
   DetailActivity::class.java)

intent.putExtra("HERO_OBJECT", hero)

startActivity(intent)
```

## Data Class

```
@Parcelize
data class Hero(
    var name: String,
    var description: String,
    var photo: String
) : Parcelable
```

## Receive Data

```
val hero = intent.getParcelableExtra
   <Hero>("HERO_OBJECT") as Hero
```

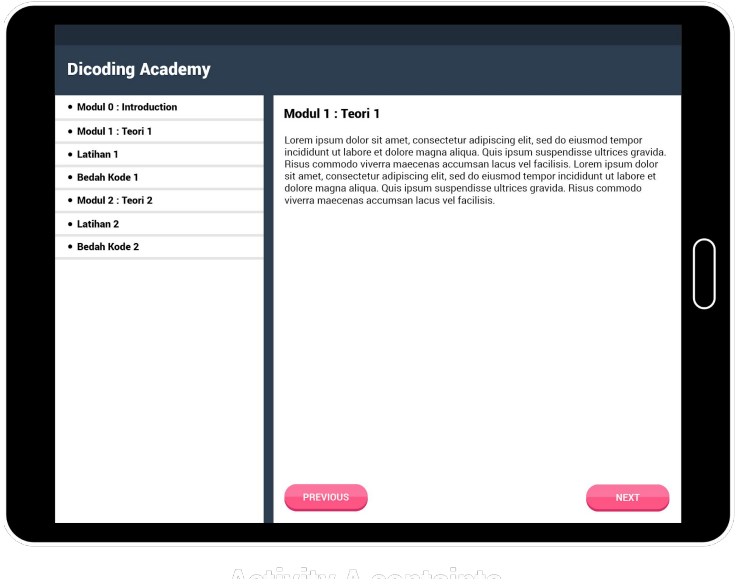
# Demo Link

https://github.com/dicodingacademy/demo-ilt-android-bangkit/tree/main/ILT2/Intent

# Fragment

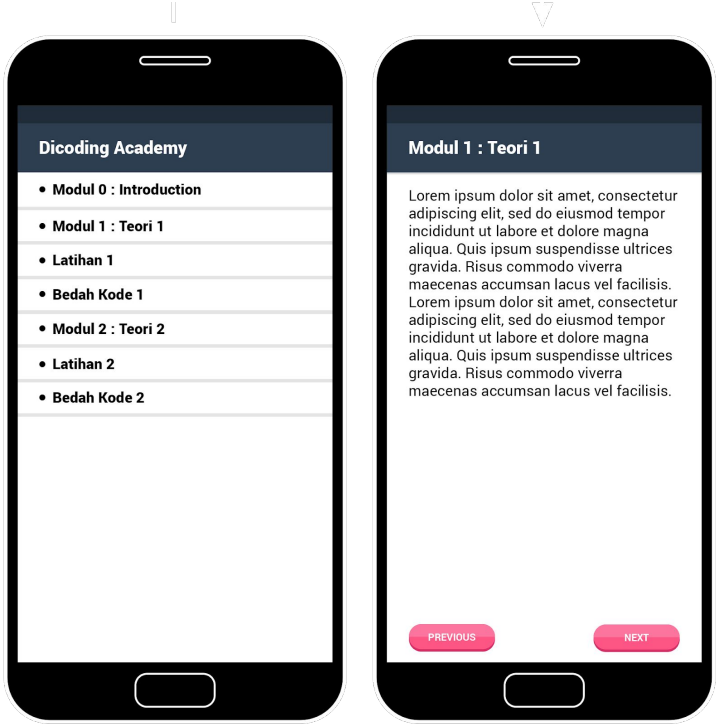
# Fragment

- Fragments are one part of the User Interface apart from Activities.
- The shape is almost the same as Activity, where it has classes for logic code and XML views.
- The difference is that the class extends (inherits) to the Fragment and does not need to register in AndroidManifest.xml



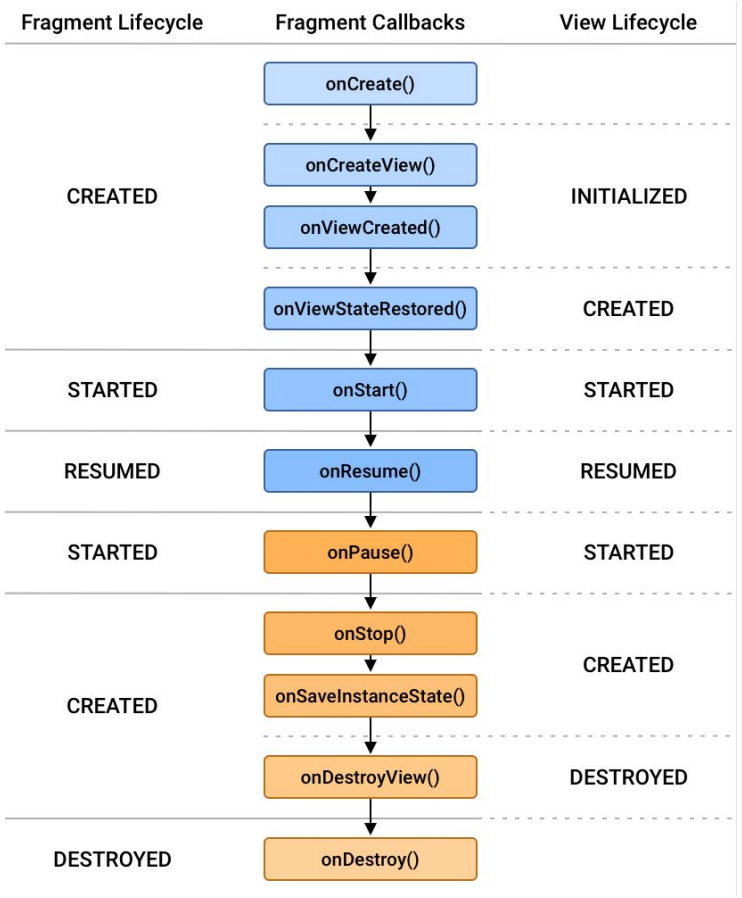
# Fragment

- Fragments are one part of the User Interface apart from Activities.
- The shape is almost the same as Activity, where it has classes for logic code and XML views.
- The difference is that the class extends (inherits) to the Fragment and does not need to register in AndroidManifest.xml



# Fragment Lifecycle

- Each Fragment instance has its own lifecycle.
- Fragment is fully dependent on the activity lifecycle in which it is embedded.



# FragmentManager Comparison

- In Activity

```
supportFragmentManager
        .beginTransaction()
        .add(R.id.frame_container, ExampleFragment())
        .commit()
```

- In Fragment

```
childFragmentManager
        .beginTransaction()
        .replace(R.id.frame_container, ExampleFragment())
        .addToBackStack(null)
        .commitNow()
```

# Fragment and Activity Comparison

```
override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    ...
    val button: Button = view.findViewById(R.id.button)
    button.setOnClickListener {
        Toast.makeText(activity, "Hello", Toast.LENGTH_SHORT).show()
    }
}
```

Same logic in Activity

```
override fun onCreate(savedInstanceState: Bundle?) {
    ...
    val button: Button = findViewById(R.id.button)
    button.setOnClickListener {
        Toast.makeText(this, "Hello", Toast.LENGTH_SHORT).show()
    }
}
```

# Send Data to Fragment

- Send Data

```
val fragment = HomeFragment()
fragment.arguments = Bundle().apply {
    putInt(HomeFragment.ARG_SECTION_NUMBER, position + 1)
}
```

- Receive Data

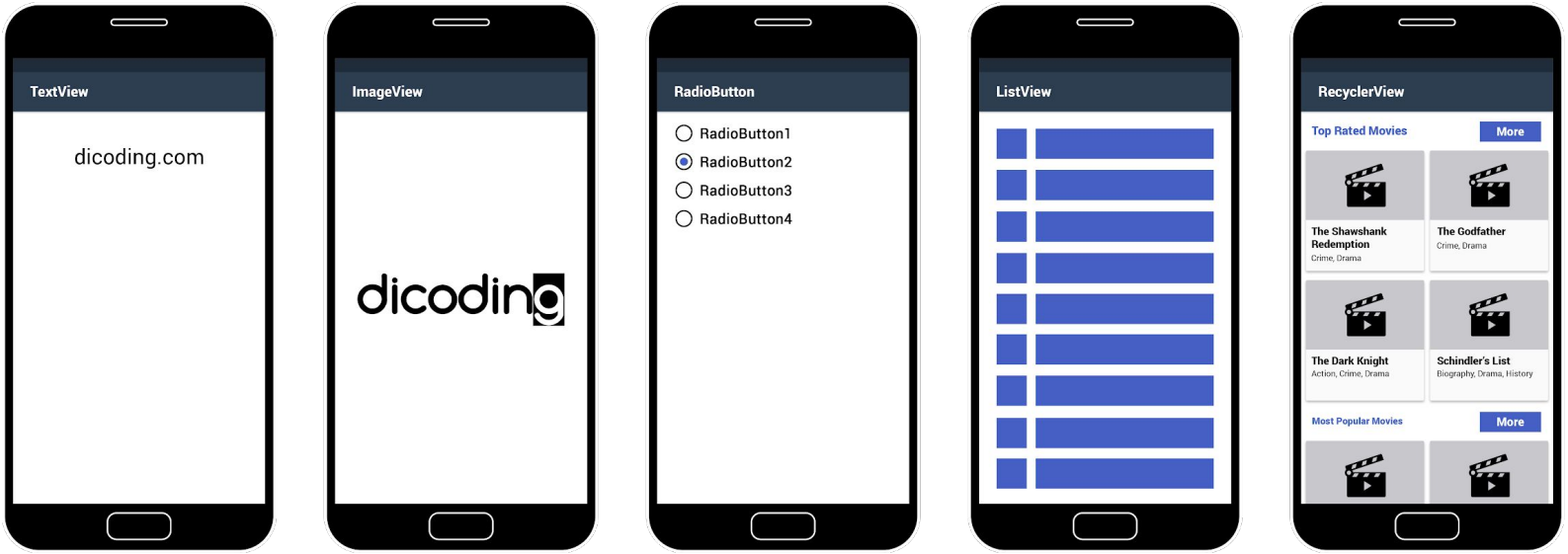
```
val index = arguments?.getInt(ARG_SECTION_NUMBER, 0)
```

# Demo Link

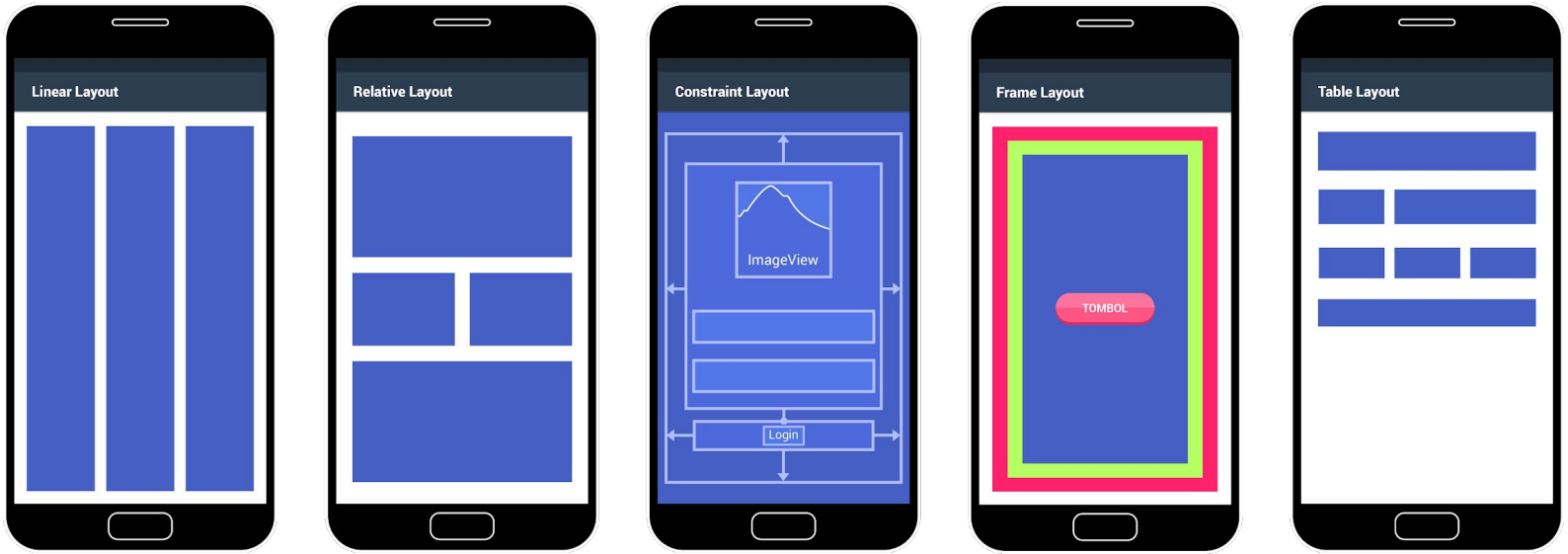
https://github.com/dicodingacademy/demo-ilt-android-bangkit/tree/main/ILT2/Fragment

# View & ViewGroup

# View and ViewGroup

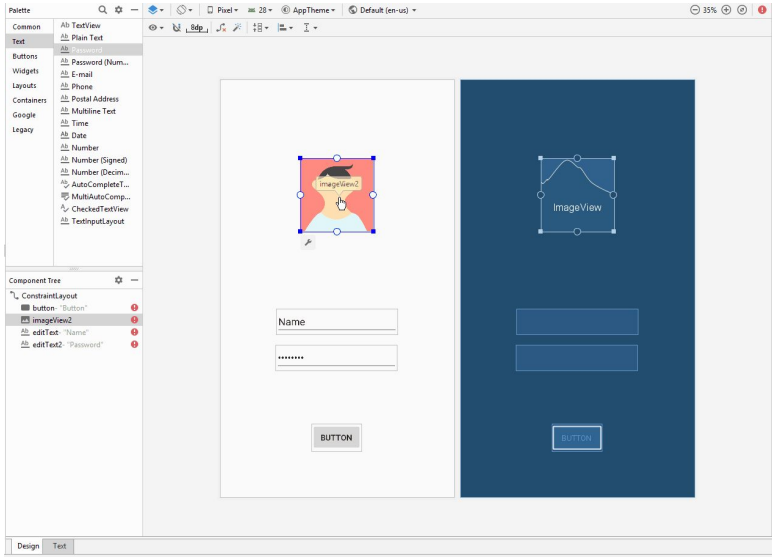


# View and ViewGroup



# ConstraintLayout

- Acts as a default layout for new Android Studio project.
- Builds complex application views without a nested layout.
- As a ViewGroup, ConstraintLayout offers flexibility for layout design.
- Provides constraints to determine positions and alignment of UI elements.
- Acts as a connection to another view, parent layout, or invisible guideline.



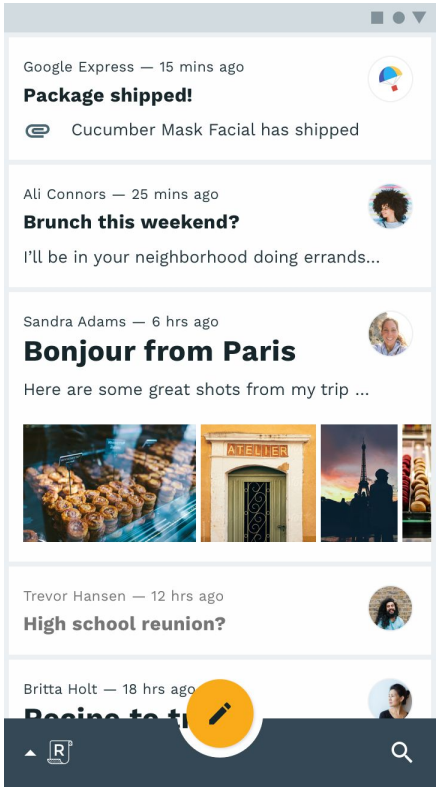
# Styles & Theme

# Using Styles

```
<Button
   android:id="@+id/btn_calculate"
   android:text="@string/calculate"
   style="@style/MyButton" />
```

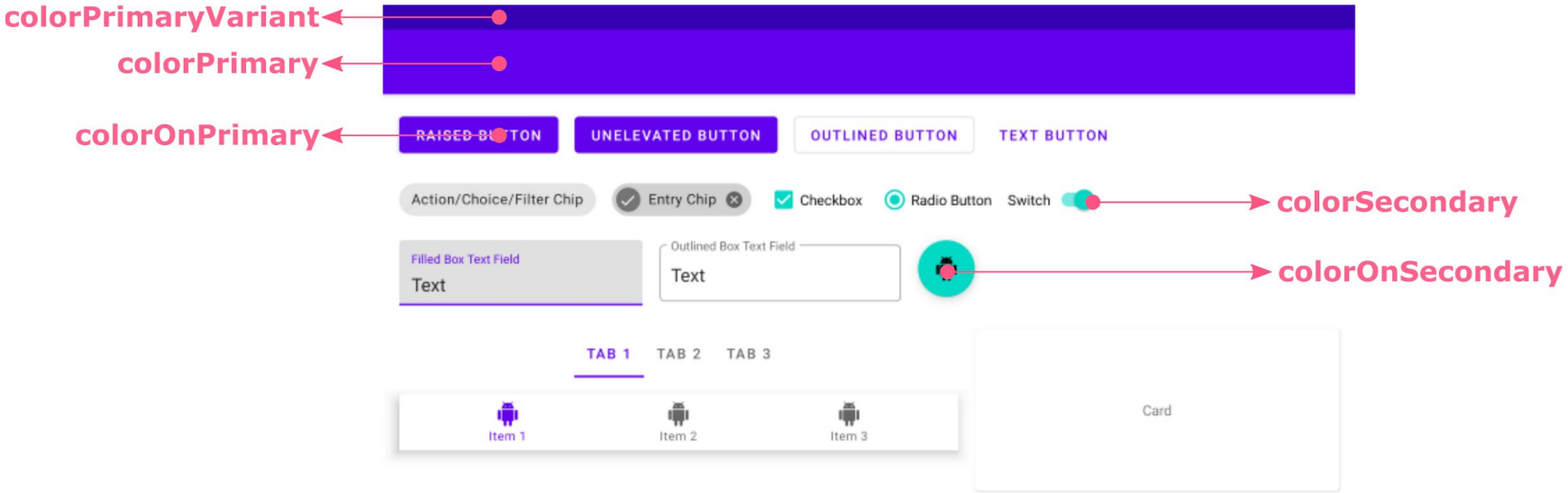
Styles in themes.xml

```
<style name="MyButton">
   <item name="android:layout_width">match_parent</item>
   <item name="android:layout_height">wrap_content</item>
   <item name="android:textColor">@color/white</item>
   <item name="android:padding">16dp</item>
   <item name="android:textSize">24sp</item>
   <item name="android:textStyle">bold</item>
</style>
```




bangkit

# Material Design Component



# Material Design Component

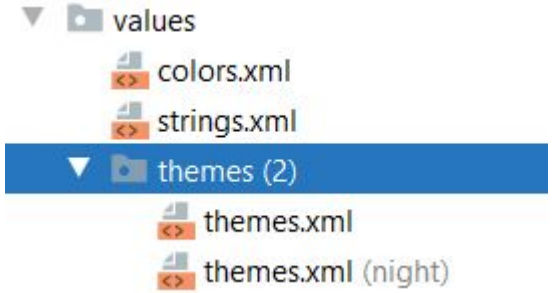
AndroidManifest.xml

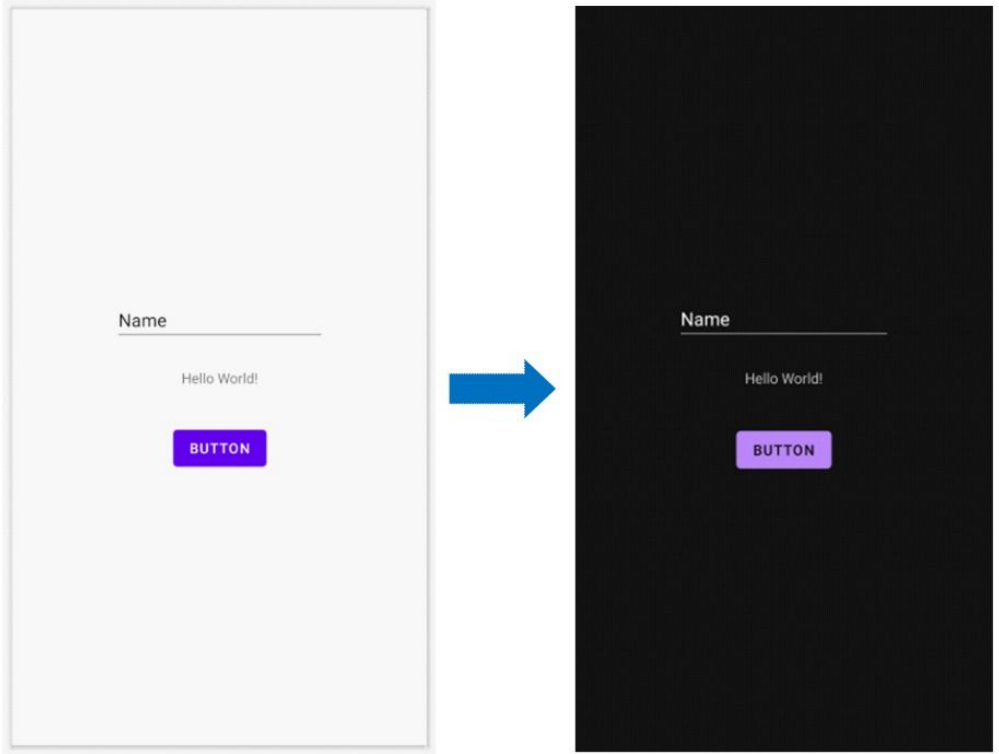
```
<application
    ...
    android:theme="@style/Theme.MyApplication">
```

Themes in themes.xml

```
<style name="Theme.MyApplication" parent="Theme.MaterialComponents.DayNight.DarkActionBar">
    <item name="colorPrimary">@color/purple_500</item>
    <item name="colorPrimaryVariant">@color/purple_700</item>
    <item name="colorOnPrimary">@color/white</item>
    <item name="colorSecondary">@color/teal_200</item>
    <item name="colorSecondaryVariant">@color/teal_700</item>
    <item name="colorOnSecondary">@color/black</item>
    <item name="android:statusBarColor" tools:targetApi="l">?attr/colorPrimaryVariant</item>
</style>
```

# Dark Theme



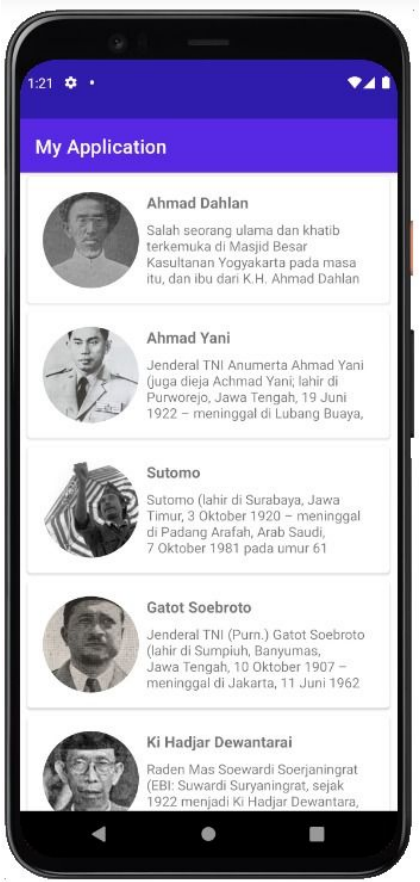




bangkit

# RecyclerView

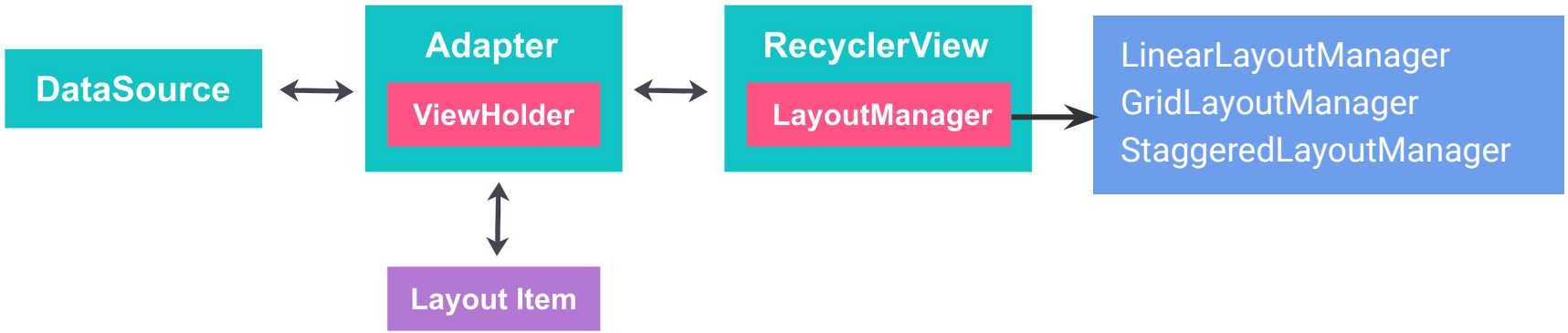
# RecyclerView

- Has a light memory and good performance.
- Contains ViewHolder by default.
- Easy animations for adding, updating, or removing items.
- Support LayoutManager to change layout.
- Support vertical and horizontal scrolling.
- Can be used together with DiffUtil.



# RecyclerView Components

- **DataSource** — ArrayList or List that contain data class
- **Adapter** — connects data to the RecyclerView
- **Layout Item** — XML file for one row of  item
- **ViewHolder** — has view information for displaying one item
- **LayoutManager** — handles the organization of UI components in a View



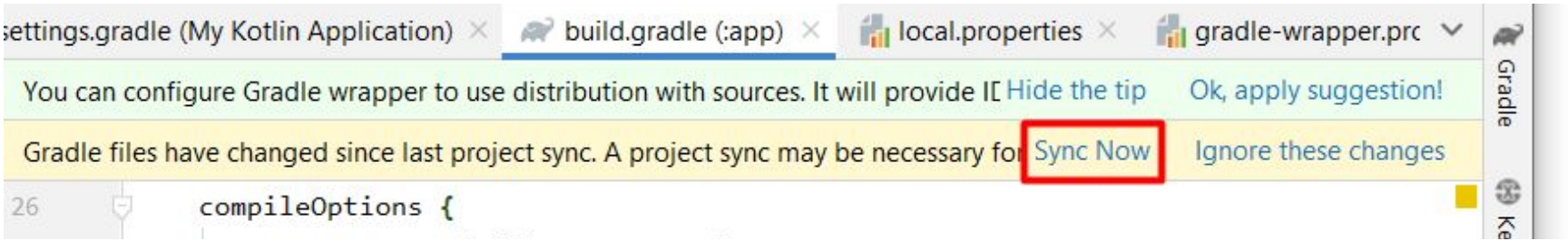
# Demo Link

https://github.com/dicodingacademy/demo-ilt-android-bangkit/tree/main/ILT2/RecyclerView

# Add Library

# Add Library

```
dependencies {
    ...
    implementation 'com.github.bumptech.glide:glide:4.13.0'
}
```

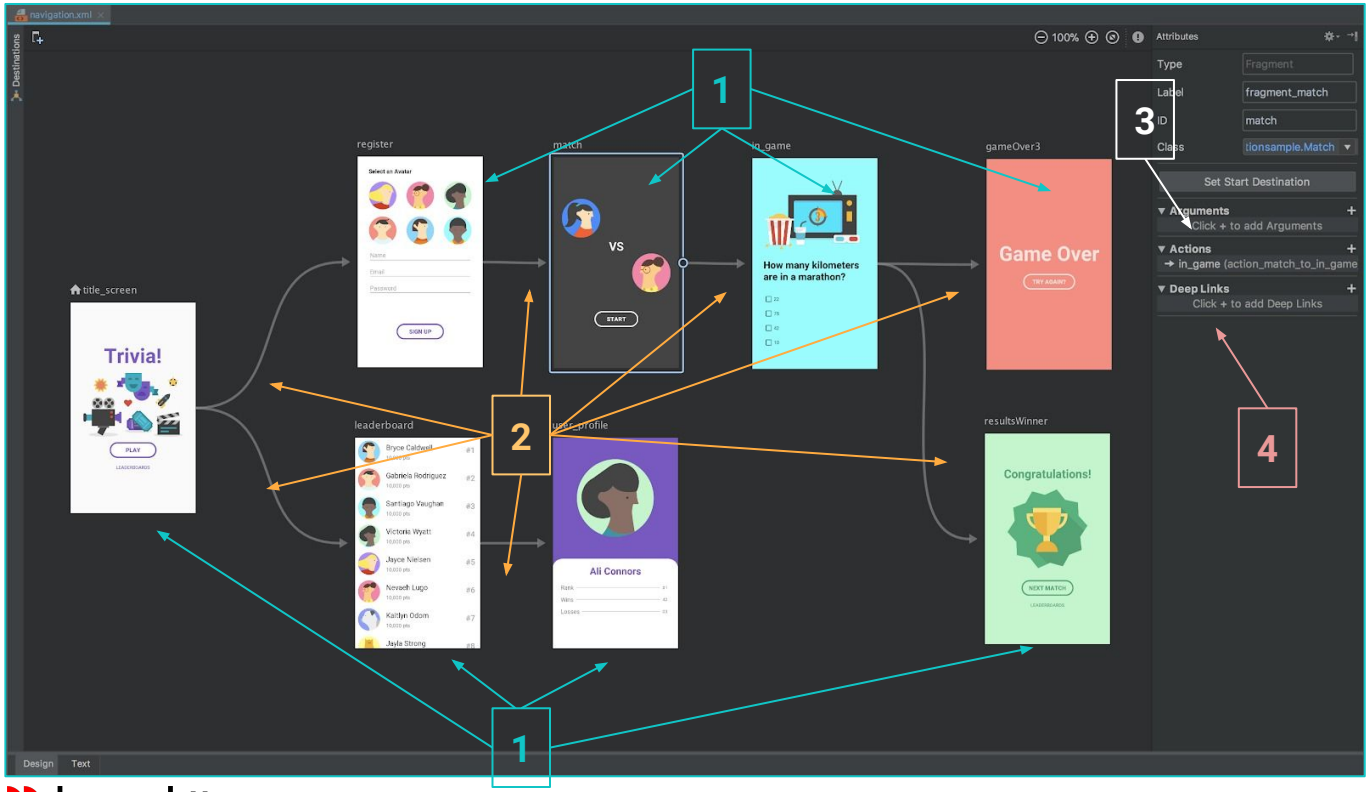


## Demo Link

- Finding out the library for something (Search Library Glide Android)
- View documentation (Glide)
- See the version (Tags)
- Implementing to Project (How Do I Use Glide)

# Navigation

# Navigation Graph




1. Destination
2. Action
3. Argument
4. DeepLink

bangkit

# NavHost & NavController

## NavHostFragment

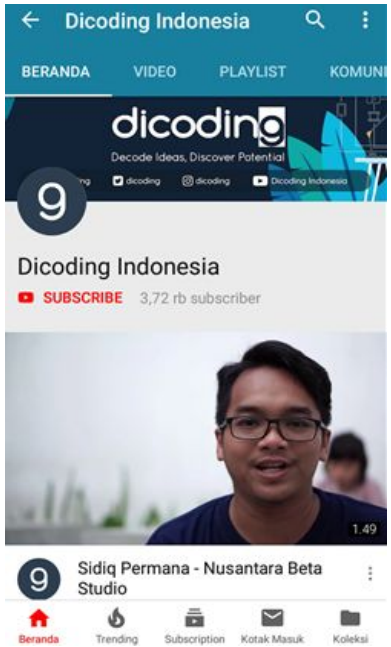
```
<fragment
   android:id="@+id/container"
   android:name="androidx.navigation.fragment.NavHostFragment"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   app:defaultNavHost="true"
   app:navGraph="@navigation/main_navigation" />
```

## NavController

```
btnCategory.setOnClickListener { view ->
    view.findNavController().navigate(
          R.id.action_homeFragment_to_categoryFragment
     )
}
```

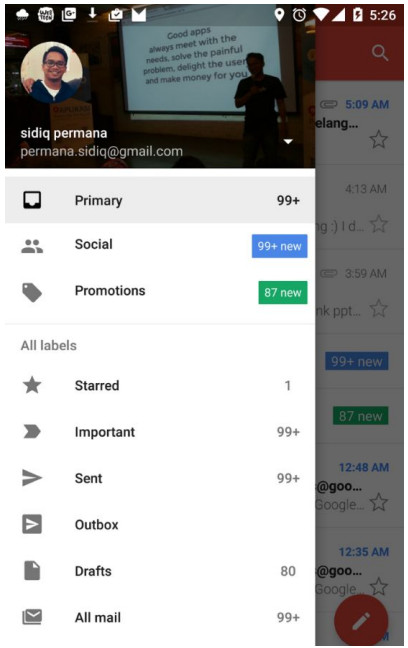
# Types of Navigation Layouts

## Bottom Navigation View



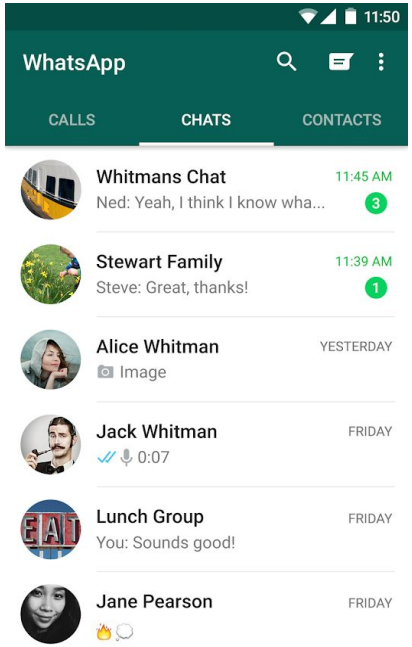

1

## Navigation Drawer



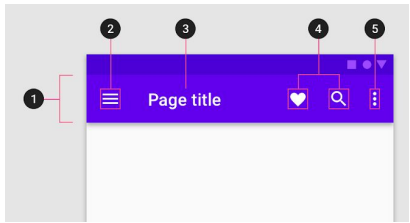

2

## Tab Layout




3

## Action Bar




4

bangkit

# View Binding

# View Binding Implementation

```
class MainActivity : AppCompatActivity() {
  private lateinit var binding: ActivityMainBinding

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    binding = ActivityMainBinding
      .inflate(layoutInflater)
    setContentView(binding.root)
    binding.tvWelcome.text = "Hello Dicoding"
  }
}
```

Activity

```
private var _binding: FragmentHomeBinding? = null
private val binding get() = _binding!!

override fun onCreateView(
  inflater: LayoutInflater, container: ViewGroup?,
  savedInstanceState: Bundle?
): View? {
  _binding = FragmentHomeBinding
    .inflate(inflater, container, false)
  return binding.root
}

override fun onDestroyView() {
   super.onDestroyView()
   _binding = null
}
```

Fragment

```
override fun onCreateViewHolder(viewGroup: ViewGroup, i: Int): ListViewHolder {
  return MyViewHolder(ItemHeroBinding.inflate(LayoutInflater.from(viewGroup.context), viewGroup, false))
}
…
inner class MyViewHolder private constructor (val binding: ItemHeroBinding): RecyclerView.ViewHolder(binding.root){
  fun bind(hero: Hero) {
    binding.txtName.setText(hero.name)
    binding.txtDescription.setText(hero.description)
    binding.imgPhoto.setImageResource(hero.photo)
  }
}
```

Adapter

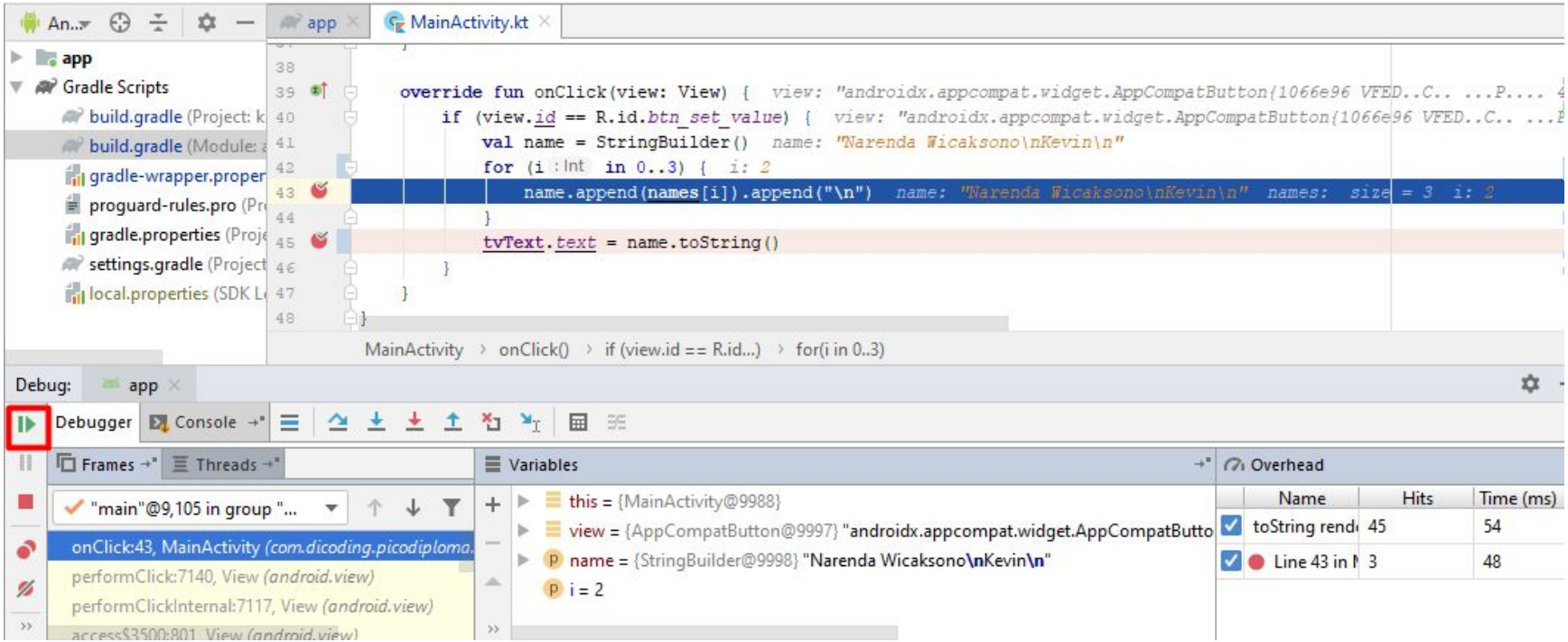
bangkit

# View Binding Comparison

| | Compile Time Safety | Elegance | Correct Type of View | Build Speed Impact | Support Kotlin & Java |
|---|---|---|---|---|---|
| findViewById | ❌ | ❌ | ❌ | ✅ | ✅ |
| ViewBinding | ✅ | ✅ | ✅ | ✅ | ✅ |

# Debugging

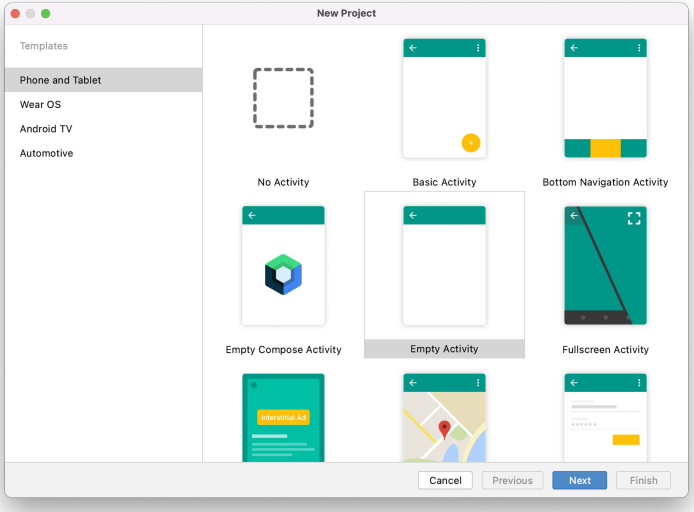
# Debugging


app
MainActivity.kt
app
Gradle Scripts
build.gradle (Project: k
build.gradle (Module:
gradle-wrapper.proper
proguard-rules.pro (Pr
gradle.properties (Proje
settings.gradle (Project
local.properties (SDK L
override fun onClick(view: View) {
if (view.id == R.id.btn_set_value) {
val name = StringBuilder()
for (i : Int in 0..3) {
name.append(names[i]).append("\n")
}
tvText.text = name.toString()
}
}
}
MainActivity › onClick() › if (view.id == R.id...) › for(i in 0..3)
Debug:
app
Debugger
Console
Frames
Threads
"main"@9,105 in group "...
onClick:43, MainActivity (com.dicoding.picodiploma.
performClick:7140, View (android.view)
performClickInternal:7117, View (android.view)
Variables
this = {MainActivity@9988}
view = {AppCompatButton@9997} "androidx.appcompat.widget.AppCompatButto
name = {StringBuilder@9998} "Narenda Wicaksono\nKevin\n"
i = 2
Overhead
Name
Hits
Time (ms)
toString rend
45
54
Line 43 in
3
48

# Tips & Tricks

# Tips & Tricks

- Code Completion
- Navigate Code
- Debugger
- Evaluate Expression in Debugger
- Code Inspection & Code Cleanup
- Android Profiler
- Project Template
- Keyboard Shortcut




bangkit

# Sharing

# Quiz

# Discussion

# Thank You