

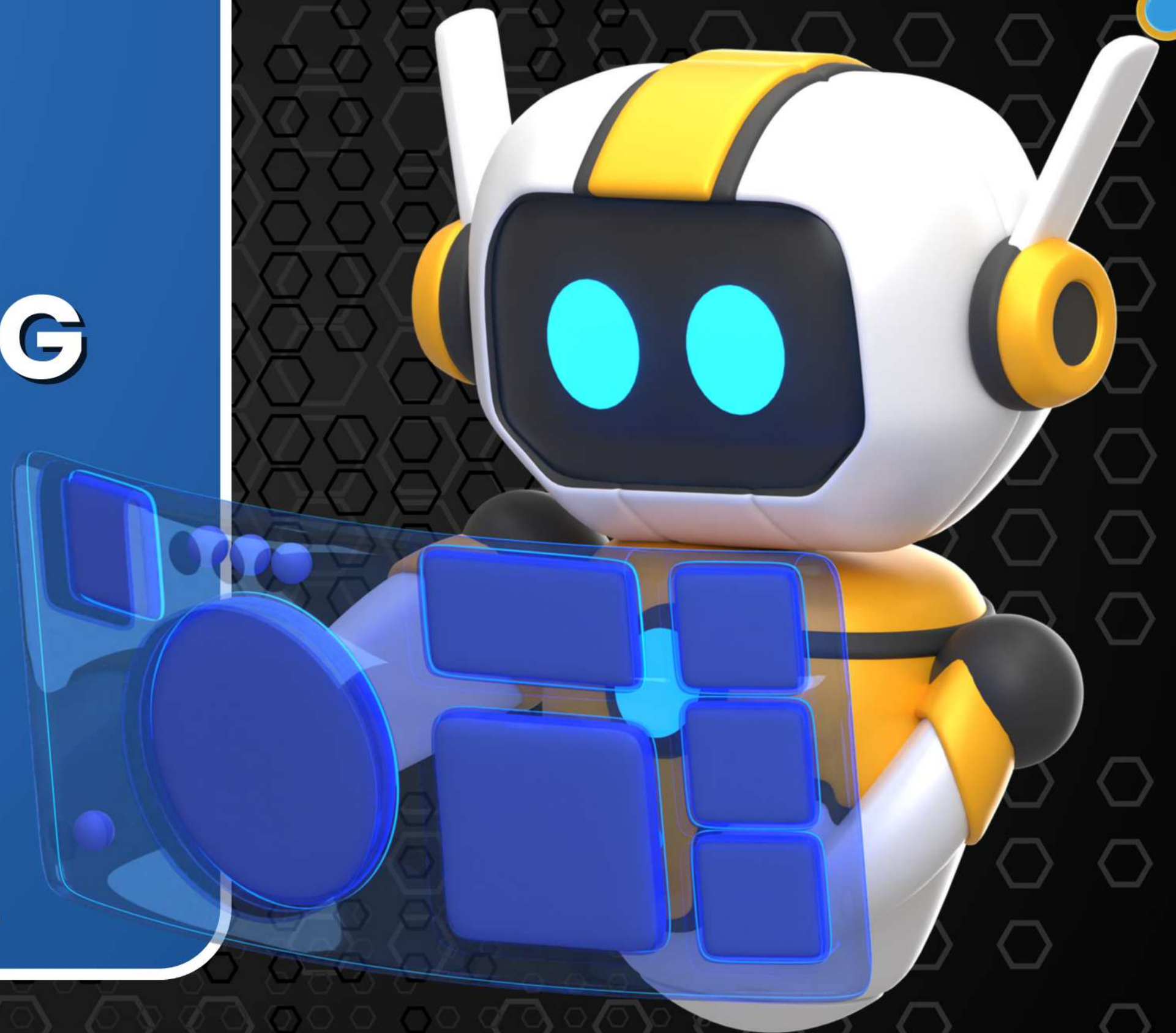


Universitas Pamulang
Sistem Informasi

MOBILE PROGRAMMING

PERTEMUAN 10

Yudisti Prayigo Permana, S.Kom., M.Kom.





TOPIK PEMBAHASAN

- Pengenalan TabLayout
- Pengenalan ViewPager
- Contoh dan Komponen Utama
- Latihan
- Tugas

01

02

03

04

05





Apa itu TabLayout?

TabLayout merupakan komponen antarmuka pengguna (UI) yang digunakan untuk menampilkan tab horizontal. ini sangat berguna ketika ingin menyajikan berbagai konten atau kategori dalam satu aktivitas (activity) yang bisa diakses pengguna dengan menggeser atau mengetuk tab tertentu. TabLayout biasanya digunakan bersama dengan ViewPager2 untuk navigasi antar halaman konten





Komponen TabLayout

- TabLayout - Menampilkan tab dibagian atas
- ViewPager2 - Mengelola dan menampilkan isi dari tiap tab
- TabLayoutMediator - Menyinkronkan TabLayout dengan ViewPager2



Implementasi TabLayout



```
1 <com.google.android.material.tabs.TabLayout
2     android:id="@+id/tab_layout"
3     android:layout_width="match_parent"
4     android:layout_height="wrap_content"/>
```



```
1 <TextView xmlns:android="http://schemas.android.com/apk/res/android"
2     android:layout_width="match_parent"
3     android:layout_height="match_parent"
4     android:text="Ini Tab 1"
5     android:gravity="center"
6     android:textSize="24sp"/>
```



```
1 class Tab1Fragment : Fragment() {
2     override fun onCreateView(
3         inflater: LayoutInflater, container: ViewGroup?,
4         savedInstanceState: Bundle?
5     ): View? = inflater.inflate(R.layout.fragment_tab1, container, false)
6 }
```




Apa itu ViewPager?

ViewPager merupakan komponen Android yang memungkinkan pengguna mengeser antar tampilan (biasanya fragment) dengan gerakan swipe (geser kiri atau geser kanan). Biasanya digunakan bersama dengan TabLayout untuk navigasi berbasis tab.





Implementasi ViewPager

```
1 class Tab1Fragment : Fragment() {  
2     override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View? {  
3         return inflater.inflate(R.layout.fragment_tab1, container, false)  
4     }  
5 }
```

```
1 <LinearLayout  
2     xmlns:android="http://schemas.android.com/apk/res/android"  
3     xmlns:app="http://schemas.android.com/apk/res-auto"  
4     android:orientation="vertical"  
5     android:layout_width="match_parent"  
6     android:layout_height="match_parent">  
7  
8     <com.google.android.material.tabs.TabLayout  
9         android:id="@+id/tab_layout"  
10        android:layout_width="match_parent"  
11        android:layout_height="wrap_content"/>  
12  
13    <androidx.viewpager2.widget.ViewPager2  
14        android:id="@+id/view_pager"  
15        android:layout_width="match_parent"  
16        android:layout_height="match_parent"/>  
17 </LinearLayout>
```




Flutter – TabBar

Dalam Flutter, TabBar berfungsi seperti TabLayout di Android yang digunakan untuk membuat navigasi horizontal antar tab. TabBar bekerja bersama TabBarView untuk menampilkan isi dari tiap tab.

Komponen TabBar

1. **TabBar** - Menampilkan tab yang dapat diklik
2. **TabBarView** - Menampilkan konten berdasarkan tab yang aktif
3. **TabController** - Mengatur logika tab (opsional jika jumlah tab tetap)
4. **DefaultTabController** - Mempermudah penggunaan **TabBar** dengan **TabBarView**



Contoh

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   @override
9   Widget build(BuildContext context) {
10    return MaterialApp(
11      title: 'Contoh TabBar Flutter',
12      theme: ThemeData(primarySwatch: Colors.purple),
13      home: TabBarExample(),
14    );
15  }
16 }
```

```
1 class TabBarExample extends StatelessWidget {
2   @override
3   Widget build(BuildContext context) {
4     return DefaultTabController(
5       length: 3, // jumlah tab
6       child: Scaffold(
7         appBar: AppBar(
8           title: Text('TabBar Flutter'),
9           bottom: TabBar(
10             tabs: [
11               Tab(text: "Beranda", icon: Icon(Icons.home)),
12               Tab(text: "Pesan", icon: Icon(Icons.message)),
13               Tab(text: "Profil", icon: Icon(Icons.person)),
14             ],
15           ),
16         ),
17         body: TabBarView(
18           children: [
19             Center(child: Text('Halaman Beranda')),
20             Center(child: Text('Halaman Pesan')),
21             Center(child: Text('Halaman Profil')),
22           ],
23         ),
24       ),
25     );
26   }
27 }
28
```




Penjelasan Contoh

- DefaultTabController - Mengelola controller tab otomatis
- AppBar.bottom - Tempat TabBar ditampilkan
- TabBarView - Isi dari masing-masing tab
- Tab - Elemen dalam TabBar (bisa teks, ikon atau keduanya)

Kelebihan

- Mudah digunakan dengan DefaultTabController
- Sangat fleksibel dan customizable
- Terintegrasi baik dengan Material Design

Kekurangan

- Navigasi terbatas ke horizontal
- Tidak cocok untuk jumlah tab terlalu banyak
- Perlu pengelolaan controller jika dinamis



Implementasi PageView

```
1 import 'package:flutter/material.dart';  
2  
3 void main() => runApp(MyApp());  
4  
5 class MyApp extends StatelessWidget {  
6   @override  
7   Widget build(BuildContext context) {  
8     return MaterialApp(  
9       home: PageViewExample(),  
10    );  
11  }  
12 }
```

```
1 class PageViewExample extends StatelessWidget {  
2   @override  
3   Widget build(BuildContext context) {  
4     return Scaffold(  
5       body: PageView(  
6         children: [  
7           Container(color: Colors.red, child: Center(child: Text("Page 1"))),  
8           Container(color: Colors.green, child: Center(child: Text("Page 2"))),  
9           Container(color: Colors.blue, child: Center(child: Text("Page 3"))),  
10        ],  
11      ),  
12    );  
13  }  
14 }  
15
```




Android vs Flutter

Aspek	Android (TabLayout + ViewPager)	Flutter (TabBar + PageView)
Komponen tab	TabLayout dari Material Components	TabBar dari Material
Komponen halaman swipe	ViewPager2	TabBarView/PageView
Kontrol manual halaman	PageController, ViewPager2 method	PageController
Penghubung tab dan page	TabLayoutMediator	DefaultTabController
Customisasi tab	Sangat fleksibel	Sangat fleksibel
Arah geser halaman	Horizontal/Vertikal (dengan ViewPager2)	Horizontal / Vertikal (scrollDirection)
Adapter / builder	Gunakan FragmentStateAdapter	Bisa pakai builder : PageView.builder()



Android vs Flutter



```
1 val viewPager = findViewById<ViewPager2>(R.id.view_pager)
2 val tabLayout = findViewById<TabLayout>(R.id.tab_layout)
3
4 viewPager.adapter = TabAdapter(this)
5
6 TabLayoutMediator(tabLayout, viewPager) { tab, position ->
7     tab.text = "Tab ${position + 1}"
8 }.attach()
9
```



```
1 DefaultTabController(
2     length: 3,
3     child: Scaffold(
4         appBar: AppBar(
5             bottom: TabBar(tabs: [
6                 Tab(text: "Tab 1"),
7                 Tab(text: "Tab 2"),
8                 Tab(text: "Tab 3"),
9             ]),
10    ),
11    body: TabBarView(
12        children: [
13            Center(child: Text("Halaman 1")),
14            Center(child: Text("Halaman 2")),
15            Center(child: Text("Halaman 3")),
16        ],
17    ),
18 ),
19 );
```




SOURCE CODE



main.dart

```
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: TabLayoutPage(),
    );
  }
}
```

```
class TabLayoutPage extends StatefulWidget {
  const TabLayoutPage({super.key});

  @override
  State<TabLayoutPage> createState() => _TabLayoutState();
}

class _TabLayoutState extends State<TabLayoutPage> {
  @override
  Widget build(BuildContext context) {
    return const SimpleAppBar();
  }
}
```




SOURCE CODE

```
class SimpleAppBar extends StatelessWidget {
  const SimpleAppBar({super.key});

  @override
  Widget build(BuildContext context) {
    PageController controller = PageController(
      initialPage: 0,
      viewportFraction: 0.8,
    );

    List<Color> colors = [Colors.red, Colors.blue, Colors.green];

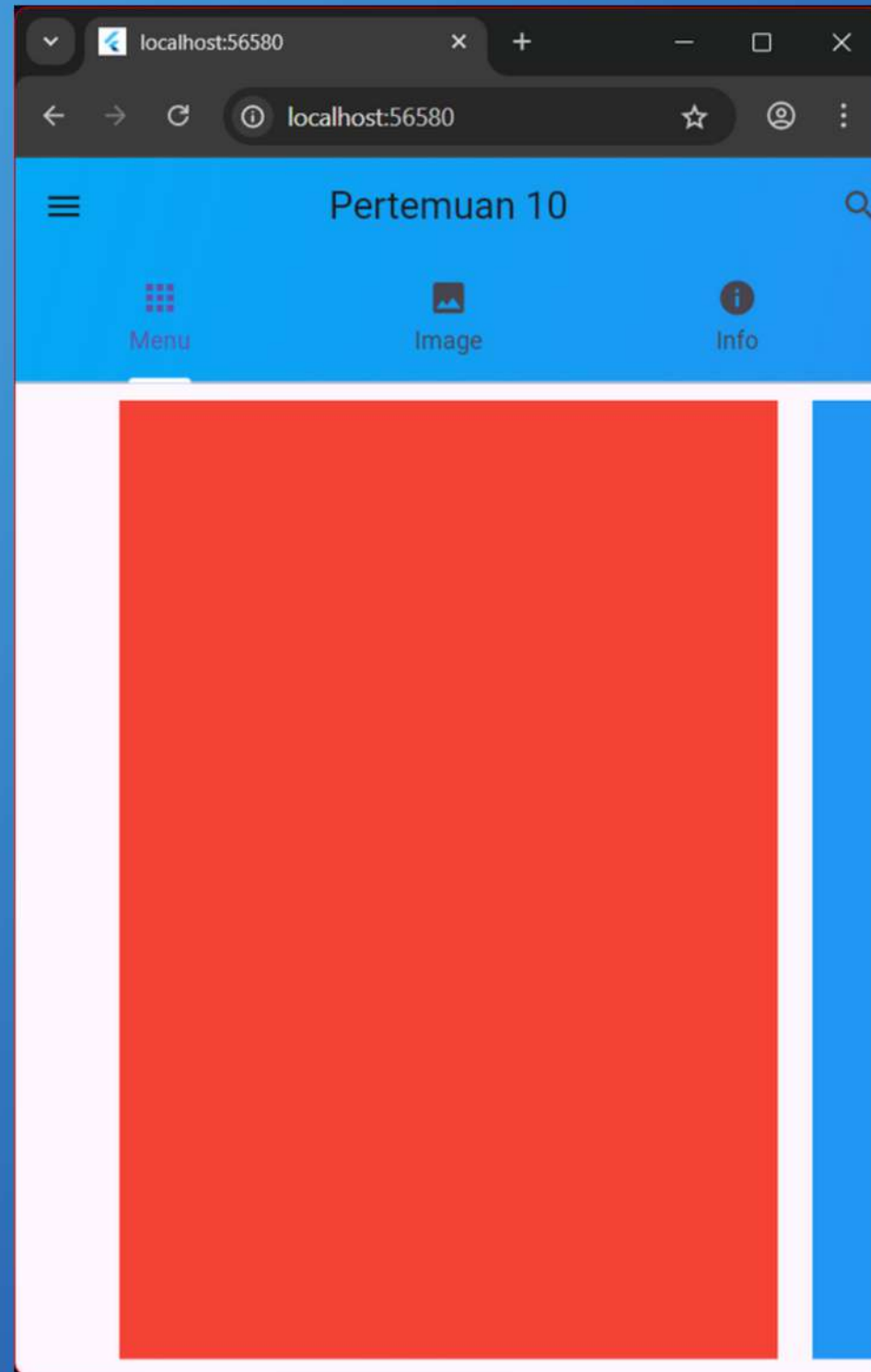
    return DefaultTabController(
      length: 4,
      child: Scaffold(
        appBar: AppBar(
          leading: const Icon(Icons.menu),
          title: const Text("Pertemuan 10"),
          centerTitle: true,
          actions: const [
            Icon(Icons.search),
          ],
        ),
      ),
    );
  }
}
```

```
background-color: Colors.red,
flexibleSpace: Container(
  decoration: const BoxDecoration(
    gradient: LinearGradient(
      colors: [Colors.blue, Colors.lightBlue]
      begin: Alignment.bottomRight,
      end: Alignment.topLeft,
    ),
  ),
),
bottom: const TabBar(
  indicatorColor: Colors.white,
  tabs: [
    Tab(
      icon: Icon(Icons.apps),
      text: "Menu",
    ),
    Tab(
      icon: Icon(Icons.image),
      text: "Image",
    ),
    Tab(
      icon: Icon(Icons.info),
      text: "Info",
    ),
  ],
),
),
),
```

```
body: TabBarView(
  children: [
    Center(
      child: PageView.builder(
        controller: controller,
        itemCount: colors.length,
        itemBuilder: (context, index) {
          return Container(
            color: colors[index],
            width: 200,
            height: 200,
            margin: const
EdgeInsets.all(10));
        },
      ),
    ),
    const Center(child: Text("Image")),
    const Center(child: Text("Info")),
  ],
),
);
}
```




HASIL





CARA MENJALANKAN APLIKASI FLUTTER PADA REAL DEVICE

Step 1. Melakukan Pengaturan Pada Smartphone Kalian

Hal pertama yang harus dilakukan adalah, melakukan pengaturan Settings pada smartphone yang ingin kalian gunakan. Caranya :

1. Pilih menu Settings atau Setelan pada smartphone kalian
2. Pilih Additional Setting atau Setelan Tambahan
3. Pilih Developer Options
4. Aktifkan USB Debugging atau menu Debugging

Step 2. Membuka Project Flutter

1. Setelah mengatur pengaturan pada smartphone kalian, langkah berikutnya adalah
2. membuka project Flutter yang telah kalian buat.

Step 3. Menjalankan Pada Real Device

Langkah berikutnya adalah menjalankan aplikasi pada smartphone kalian.

1. Pertama-tama klik Run, kemudian pilih Start Debugging
2. Kemudian akan muncul nama device smartphone kalian.
3. Tunggu sampai prosesnya selesai

Step 4. Selesai



TUGAS

Buatlah TabLayout dan Viewpage dengan kreasi masing-masing dengan menggunakan Salomon dengan untuk 2 halaman meliputi (Beranda, TabLayout + Viewpage dan Pofile)

Tidak boleh ada yang sama

Batas pengumpulan 31 Mei 2025



Universitas Pamulang
Sistem Informasi

Thank You!

