



Universitas Pamulang
Sistem Informasi

MOBILE PROGRAMMING

PERTEMUAN 13

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





TOPIK PEMBAHASAN

- Pengenalan Media Player
- Pengenalan Audio Player

01

02





Apa itu Media Player?

Media Player pada Android digunakan untuk memutar file audio atau video. Untuk video, Android menyediakan VideoView sebagai komponen UI yang praktis untuk menampilkan video dalam aplikasi, dengan bantuan MediaController untuk kontrol seperti play, pause, rewind dan forward.





Komponen Media Player

1. Video View : Komponen UI untuk menampilkan video
2. Media Controller : Kontrol antarmuka pengguna (play, pause, dll)
3. Media Player (Opsional) : Komponen yang bisa digunakan jika butuh kontrol lebih kompleks atas media



Contoh

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent">
5
6      <VideoView
7          android:id="@+id/videoView"
8          android:layout_width="match_parent"
9          android:layout_height="wrap_content"
10         android:layout_centerInParent="true"/>
11 </RelativeLayout>
```

```
1  import android.net.Uri;
2  import android.os.Bundle;
3  import android.widget.MediaController;
4  import android.widget.VideoView;
5  import androidx.appcompat.app.AppCompatActivity;
6
7  public class MainActivity extends AppCompatActivity {
8
9      VideoView videoView;
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_main);
15
16         videoView = findViewById(R.id.videoView);
17         String videoUrl = "https://www.example.com/video.mp4"; // Ganti dengan URL atau path lokal
18         Uri uri = Uri.parse(videoUrl);
19
20         videoView.setVideoURI(uri);
21         MediaController mediaController = new MediaController(this);
22         videoView.setMediaController(mediaController);
23         mediaController.setAnchorView(videoView);
24
25         videoView.start();
26     }
27 }
28
```




Flutter

Dalam Flutter, Media Player digunakan untuk memutar file audio atau video, baik dari lokal maupun internet (streaming). Flutter sendiri tidak menyediakan komponen bawaan untuk ini, tetapi mengandalkan plugin pihak ketiga.

Plugin umum:

1. **video_player** : Untuk memutar video
2. **chewie** : Antarmuka video player yang lebih interaktif (dibangun diatas video_player)
3. **just_audio** : Untuk audio player
4. **audioplayers** : Plugin audio sederhana



Contoh

```
1 import 'package:flutter/material.dart';
2 import 'package:video_player/video_player.dart';
3
4 void main() {
5   runApp(MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   @override
10  Widget build(BuildContext context) {
11    return MaterialApp(
12      title: 'Flutter Video Player',
13      home: VideoPlayerScreen(),
14    );
15  }
16 }
17
18 class VideoPlayerScreen extends StatefulWidget {
19   @override
20   _VideoPlayerScreenState createState() => _VideoPlayerScreenState();
21 }
22
23 class _VideoPlayerScreenState extends State<VideoPlayerScreen> {
24   late VideoPlayerController _controller;
25
26   @override
27   void initState() {
28     super.initState();
29     // Bisa menggunakan URL atau path lokal (assets)
30     _controller = VideoPlayerController.network(
31       'https://www.example.com/sample-video.mp4',
32     )..initialize().then((_) {
33       setState(() {}); // Update UI ketika video siap diputar
34     });
35     _controller.setLooping(true);
36     _controller.play();
37   }
38 }
```

```
1 @override
2 void dispose() {
3   _controller.dispose();
4   super.dispose();
5 }
6
7 @override
8 Widget build(BuildContext context) {
9   return Scaffold(
10     appBar: AppBar(title: Text("Video Player")),
11     body: Center(
12       child: _controller.value.isInitialized
13         ? AspectRatio(
14             aspectRatio: _controller.value.aspectRatio,
15             child: VideoPlayer(_controller),
16           )
17       : CircularProgressIndicator(),
18     ),
19     floatingActionButton: FloatingActionButton(
20       onPressed: () {
21         setState(() {
22           _controller.value.isPlaying
23             ? _controller.pause()
24             : _controller.play();
25         });
26       },
27       child:
28         Icon(_controller.value.isPlaying ? Icons.pause : Icons.play_arrow),
29     ),
30   );
31 }
32 }
```



Android VS Flutter

Aspek	Android Native	Flutter
Komponen Bawaan	Ya - VideoView, MediaPlayer tersedia langsung	Tidak - Harus pakai plugin dari pub.dev
Kemudahan Penggunaan	Cukup mudah, built-in SDK	Perlu install plugin, tapi API-nya sederhana
Kontrol Playback	Bisa kontrol play, pause, seek, looping	Bisa juga, tergantung plugin (video_player & chewie)
Dukungan Format	Format didukung oleh sistem Android (Mp4, 3gp, dll)	Menggunakan native Android/iOS backend melalui plugin
UI/UX Kustomisasi	Perlu kerja manual untuk kontrol kustom	Lebih fleksibel dengan plugin chewie atau custom widget
Streaming Support	Ya - Support RTSP, HLS dan format streaming lainnya	Ya - Jika plugin mendukung (misalnya video_player bisa streaming)
Pemutaran File Lokal	Ya - Melalui raw, assets, atau file system	Ya - Dengan video_player atau audioplayers
Cross-Platform	Tidak - Hanya untuk Android	Ya - Android, iOS, Web (terbatas), Desktop (tergantung plugin)
Performa	Optimal karena native	Sedikit overhead karena jembatan platform (Platform Channel)
Fitur Tambahan (EQ, DSP)	Ada di MediaPlayer atau ExoPlayer (Equalizer, dll)	Butuh plugin tambahan atau native integration (lebih rumit)
Dukungan Offline/Cache	Ada - Bisa implementasi cache video	Terbatas - Tergantung plugin (misalnya better_player)



Android vs Flutter

```
1 import android.net.Uri;
2 import android.os.Bundle;
3 import android.widget.MediaController;
4 import android.widget.VideoView;
5 import androidx.appcompat.app.AppCompatActivity;
6
7 public class MainActivity extends AppCompatActivity {
8     VideoView videoView;
9
10    @Override
11    protected void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        setContentView(R.layout.activity_main);
14
15        videoView = findViewById(R.id.videoView);
16        Uri uri = Uri.parse("https://www.example.com/video.mp4");
17        videoView.setVideoURI(uri);
18
19        MediaController mediaController = new MediaController(this);
20        videoView.setMediaController(mediaController);
21        mediaController.setAnchorView(videoView);
22
23        videoView.start();
24    }
25 }
26
```

```
1 import 'package:flutter/material.dart';
2 import 'package:video_player/video_player.dart';
3 import 'package:chewie/chewie.dart';
4
5 void main() => runApp(MyApp());
6
7 class MyApp extends StatelessWidget {
8   @override
9   Widget build(BuildContext context) {
10     return MaterialApp(title: 'Flutter Video Player', home: VideoPlayerScreen());
11   }
12 }
13
14 class VideoPlayerScreen extends StatefulWidget {
15   @override
16   _VideoPlayerScreenState createState() => _VideoPlayerScreenState();
17 }
18
19 class _VideoPlayerScreenState extends State<VideoPlayerScreen> {
20   late VideoPlayerController _videoPlayerController;
21   ChewieController? _chewieController;
22
23   @override
24   void initState() {
25     super.initState();
26     _videoPlayerController =
27       VideoPlayerController.network('https://www.example.com/video.mp4');
28     _videoPlayerController.initialize().then((_) {
29       setState(() {
30         _chewieController = ChewieController(
31           videoPlayerController: _videoPlayerController,
32           autoPlay: true,
33           looping: true,
34         );
35       });
36     });
37   }
38
39   @override
40   void dispose() {
41     _videoPlayerController.dispose();
42     _chewieController?.dispose();
43     super.dispose();
44   }
45
46   @override
47   Widget build(BuildContext context) {
48     return Scaffold(
49       appBar: AppBar(title: Text('Flutter Video Player')),
50       body: _chewieController != null && _chewieController!.videoPlayerController.value.isInitialized
51         ? Chewie(controller: _chewieController!)
52         : Center(child: CircularProgressIndicator()),
53     );
54   }
55 }
56
```




Installation

Ketikan pada Terminal :

```
$ flutter pub add salomon_bottom_bar
```

```
$ flutter pub add icons_plus
```

```
$ flutter pub add audioplayers
```

```
$ flutter pub add video_player
```

```
$ flutter pub add file_picker
```

```
flutter pub outdated
```

Pubspec.yaml

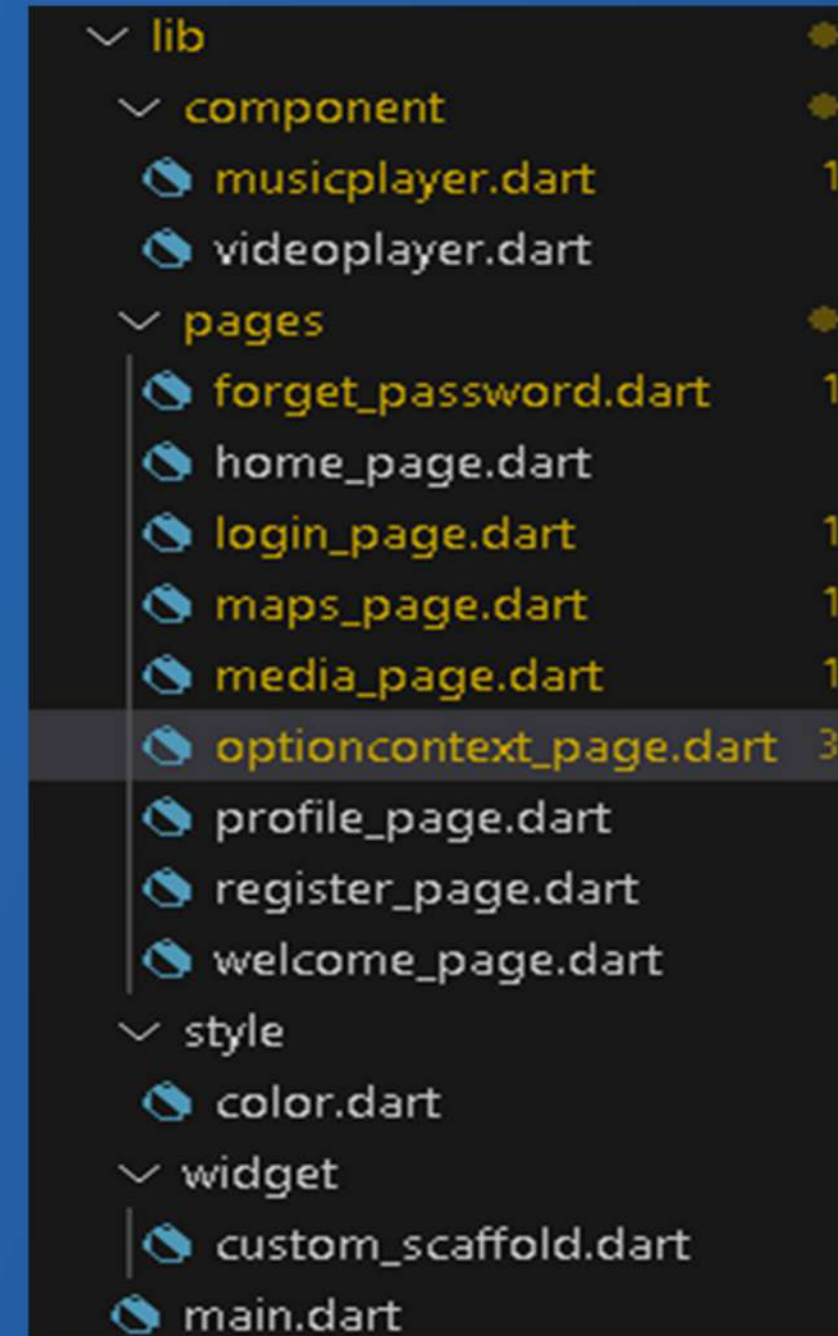
```
dependencies:  
  salomon_bottom_bar: ^3.3.2  
  icons_plus: ^5.0.0  
  audioplayers: ^6.5.0  
  video_player: ^2.10.0  
  file_picker: ^10.2.0
```




Installation

Buat susunan folder dan dua file seperti gambar disamping. Dengan cara :

- Klik kanan folder lib > New Folder > beri nama foldernya dengan 'pages', 'style', 'component' dan 'widget'
- Klik kanan folder pages > New File > beri nama filenya dengan 'profile_page.dart', 'welcome_page.dart', 'login_page.dart', 'register_page.dart', 'forgot_password.dart', 'media_page.dart', 'maps_page.dart', 'home_page.dart', 'optioncontext.dart', 'musicplayer.dart', dan 'videoplayer.dart'





SOURCE CODE

main.dart

```
import 'package:flutter/material.dart';  
import 'package:flutter_pert13/pages/welcome_page.dart';  
  
void main(){  
  runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget{  
  @override  
  Widget build(BuildContext context){  
    return MaterialApp(  
      debugShowCheckedModeBanner: false,  
      home: WelcomePage(),  
    );  
  }  
}
```

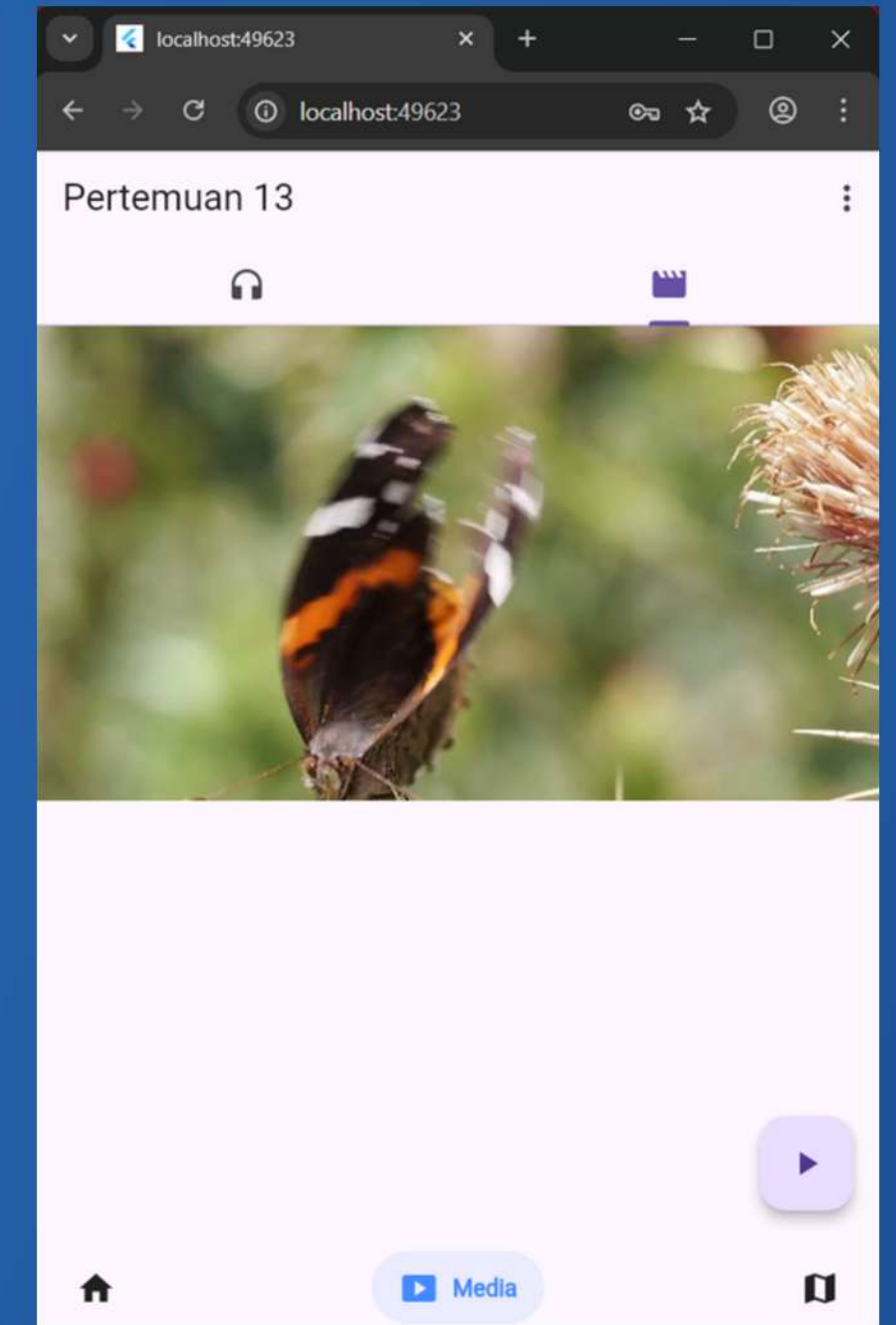
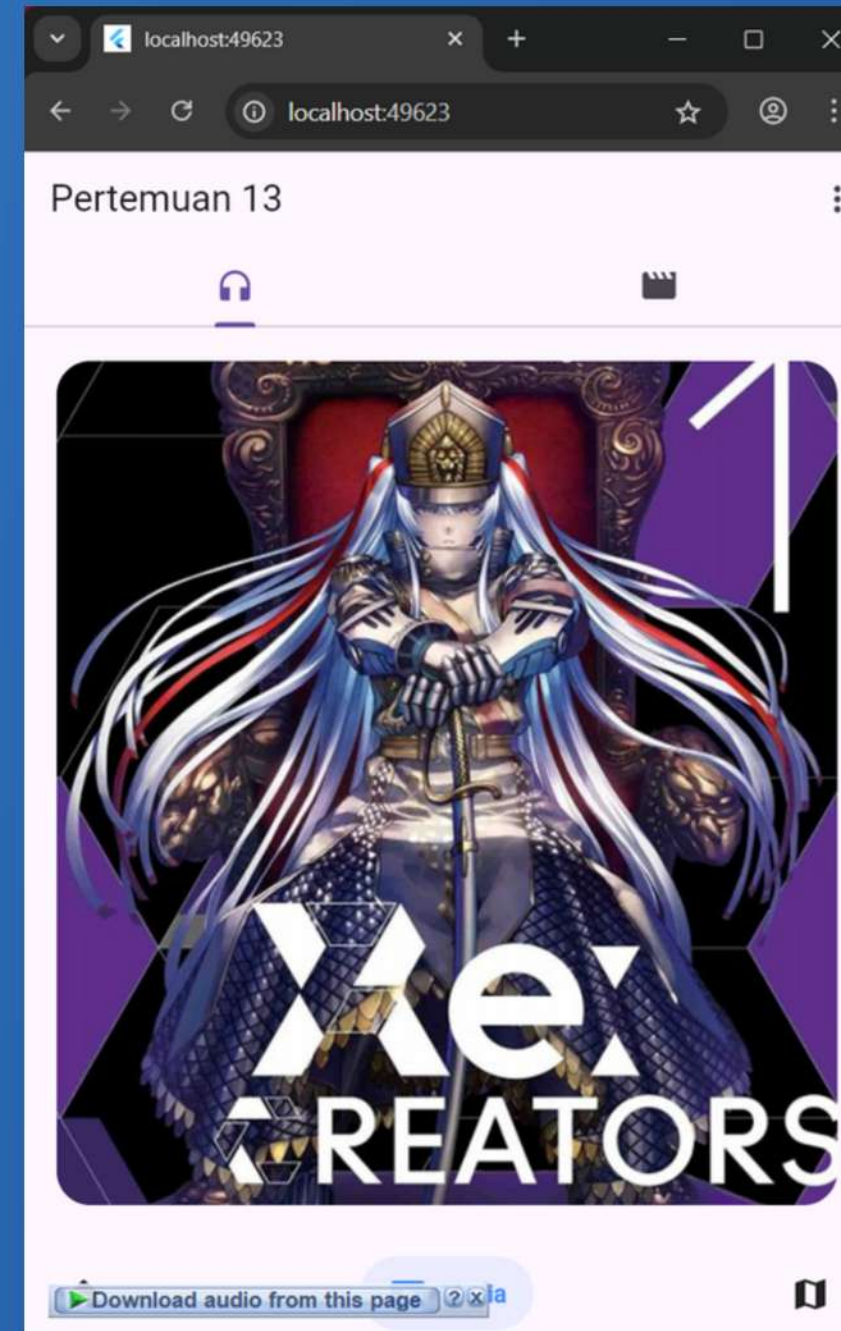
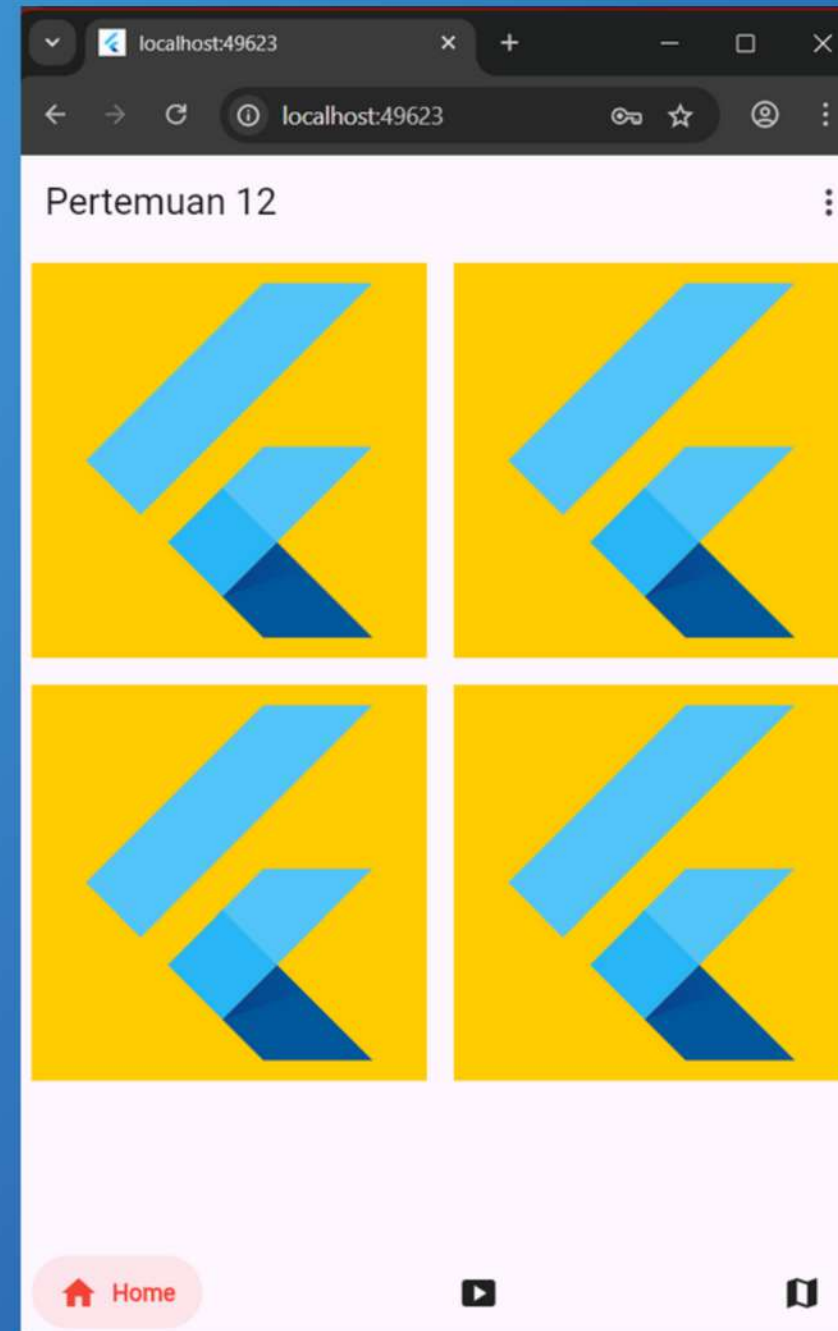
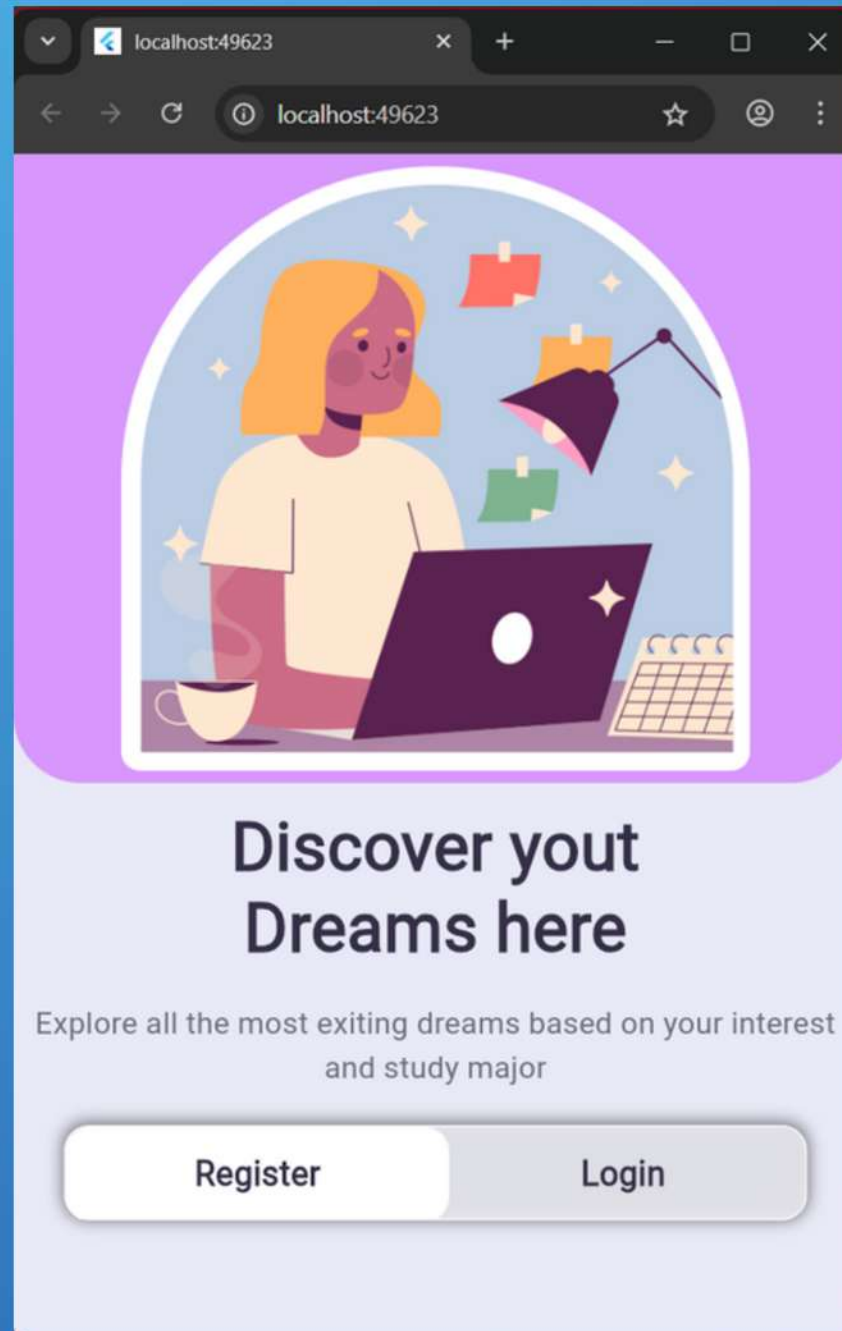



SOURCE CODE





HASIL





TUGAS

Silahkan demonstrasikan dan dokumentasikan

Batas pengumpulan 21 Juni 2025





Universitas Pamulang
Sistem Informasi

Thank You!

