

PEMROGRAMAN MOBILE

“Laporan pertemuan 14”



Di susun oleh :

Andika Ainur Wibowo

NIM 2141720238

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

Source Code:

1. Inisialisasi dan Mendapatkan Kamera

```
1 void main() async {  
2   WidgetsFlutterBinding.ensureInitialized();  
3   final cameras = await availableCameras();  
4   final firstCamera = cameras.first;  
5   runApp(MyApp(camera: firstCamera));  
6 }
```

Penjelasan:

- Fungsi `availableCameras()` digunakan untuk mendapatkan daftar kamera yang tersedia.
- `WidgetsFlutterBinding.ensureInitialized()` memastikan bahwa Flutter framework telah diinisialisasi sebelum aplikasi dimulai.
- Aplikasi diinisialisasi dengan widget `MyApp` dan kamera belakang dipilih sebagai kamera default.

2. MyApp Widget

```
1 class MyApp extends StatelessWidget {  
2   final CameraDescription camera;  
3  
4   const MyApp({Key? key, required this.camera}) : super(key: key);  
5  
6   @override  
7   Widget build(BuildContext context) {  
8     return MaterialApp(  
9       home: CameraApp(camera: camera),  
10    );  
11  }  
12 }
```

Penjelasan: `MyApp` adalah widget utama yang mengatur tata letak aplikasi dan memberikan widget `CameraApp` dengan kamera yang telah dipilih.

3. CameraApp Widget

```
1 class CameraApp extends StatefulWidget {  
2   final CameraDescription camera;  
3  
4   const CameraApp({Key? key, required this.camera}) : super(key: key);  
5  
6   @override  
7   _CameraAppState createState() => _CameraAppState();  
8 }
```

Penjelasan:

`CameraApp` adalah widget stateful yang akan menangani logika untuk mengontrol kamera.

4. State dan Inisialisasi Kamera

```
1 class _CameraAppState extends State<CameraApp> {
2   late CameraController _controller;
3   late Future<void> _initializeControllerFuture;
4   bool isCameraOn = true; // Untuk melacak status kamera (nyala/mati)
5   ResolutionPreset currentResolution = ResolutionPreset.medium;
6
7   @override
8   void initState() {
9     super.initState();
10    _controller = CameraController(
11      widget.camera,
12      currentResolution,
13    );
14    _initializeControllerFuture = _controller.initialize();
15  }
```

Penjelasan:

- Objek `_controller` digunakan untuk mengendalikan kamera.
- `isCameraOn` melacak status kamera (nyala/mati).
- `currentResolution` menyimpan resolusi kamera saat ini.

5. Metode Toggle dan Change Resolution

```
1 void toggleCamera() {
2   setState(() {
3     isCameraOn = !isCameraOn;
4     if (isCameraOn) {
5       _controller = CameraController(
6         widget.camera,
7         currentResolution,
8       );
9       _initializeControllerFuture = _controller.initialize();
10    } else {
11      _controller.dispose();
12    }
13  });
14 }
15
16 // Metode untuk mengubah resolusi kamera
17 void changeResolution() {
18   setState(() {
19     currentResolution = currentResolution == ResolutionPreset.medium
20       ? ResolutionPreset.high
21       : ResolutionPreset.medium;
22     if (isCameraOn) {
23       _controller = CameraController(
24         widget.camera,
25         currentResolution,
26       );
27       _initializeControllerFuture = _controller.initialize();
28     }
29   });
30 }
```

Penjelasan:

- `toggleCamera()` mengubah status kamera dan menginisialisasi atau menutup kamera sesuai dengan statusnya.
- `changeResolution()` mengubah resolusi kamera saat ini.

6. Membangun Tampilan

```
1 @override
2 Widget build(BuildContext context) {
3   return Scaffold(
4     appBar: AppBar(
5       title: Text('Andika Ainur Wibowo'),
6     ),
7     body: FutureBuilder<void>{
8       future: _initializeControllerFuture,
9       builder: (context, snapshot) {
10        if (snapshot.connectionState == ConnectionState.done) {
11          return Center(
12            child: ClipRect(
13              child: OverflowBox(
14                alignment: Alignment.center,
15                child: FittedBox(
16                  fit: BoxFit.fitWidth,
17                  child: Container(
18                    width: _controller.value.previewSize!.height,
19                    height: _controller.value.previewSize!.width,
20                    child: CameraPreview(_controller),
21                  ),
22                ),
23              ),
24            ),
25          );
26        } else {
27          return Center(child: CircularProgressIndicator());
28        }
29      },
30    ),
31    floatingActionButton: Column(
32      mainAxisAlignment: MainAxisAlignment.end,
33      children: [
34        FloatingActionButton(
35          onPressed: toggleCamera,
36          child: Icon(isCameraOn ? Icons.videocam_off : Icons.videocam),
37        ),
38        SizedBox(height: 16),
39        FloatingActionButton(
40          onPressed: changeResolution,
41          child: Icon(Icons.settings),
42        ),
43      ],
44    ),
45  );
46 }
```

Penjelasan:

- Widget FutureBuilder digunakan untuk menampilkan indikator loading saat kamera diinisialisasi.
- Tampilan kamera ditampilkan di CameraPreview.
- Tombol-tombol untuk mengontrol kamera ditambahkan di floatingActionButton

Output :

