📘 **MODUL PRAKTIK**

Pemanfaatan Fitur API dan Sensor (Kamera & Accelerometer) di Aplikasi Mobile

Mata Kuliah: Pemrograman Aplikasi Mobile
Tools: Visual Studio Code + Android Studio Emulator
Durasi: ±2 jam

Tujuan Pembelajaran

Setelah mengikuti modul ini, mahasiswa mampu:

1. Membuat aplikasi Flutter dengan beberapa tombol interaktif.
2. Mengintegrasikan **fitur kamera** untuk mengambil foto.
3. Menggunakan **sensor accelerometer** untuk mendeteksi pergerakan perangkat.
4. Memahami hubungan antara **UI (tombol & widget)** dengan **hardware API (kamera, sensor)**.

A. Persiapan Awal

1. Buka **VS Code**.
2. Jalankan **Android Emulator** dari Android Studio.
3. Buat proyek Flutter baru:
4. flutter create sensor_camera_app
5. cd sensor_camera_app
6. Buka file pubspec.yaml, tambahkan dependensi berikut:
7. dependencies:
8. flutter:
9. sdk: flutter
10. camera: ^0.10.5+5
11. sensors_plus: ^4.0.2
12. path_provider: ^2.1.2

13. Simpan, lalu jalankan:

14. flutter pub get

B. Membuat Struktur Folder

Struktur sederhana agar mudah diikuti:

```
lib/
 ├── main.dart
 ├── pages/
 |    ├── camera_page.dart
 |    └── accelerometer_page.dart
```

C. File main.dart

```
import 'package:flutter/material.dart';
import 'pages/camera_page.dart';
import 'pages/accelerometer_page.dart';

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
```

```
      title: 'Sensor & API Demo',
      theme: ThemeData(primarySwatch: Colors.teal),
      home: const HomePage(),
    );
  }
}

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text("Demo Sensor dan Kamera")),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            ElevatedButton.icon(
              icon: const Icon(Icons.camera_alt),
              label: const Text("Gunakan Kamera"),
              onPressed: () {
                Navigator.push(context,
                    MaterialPageRoute(builder: (_) => const CameraPage()));
              },
            ),
```

```
          const SizedBox(height: 20),
          ElevatedButton.icon(
            icon: const Icon(Icons.sensors),
            label: const Text("Lihat Accelerometer"),
            onPressed: () {
              Navigator.push(context,
                  MaterialPageRoute(builder: (_) => const AccelerometerPage()));
            },
          ),
        ],
      ),
    ),
  );
 }
}
```

Penjelasan:

- Terdapat dua tombol utama:
    - **Gunakan Kamera:** membuka halaman kamera.
    - **Lihat Accelerometer:** membuka halaman pembacaan sensor gerak.
- MaterialPageRoute digunakan untuk berpindah antar halaman.

D. File camera_page.dart

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

class CameraPage extends StatefulWidget {
```

```
  const CameraPage({super.key});

  @override
  State<CameraPage> createState() => _CameraPageState();
}

class _CameraPageState extends State<CameraPage> {
  CameraController? controller;
  List<CameraDescription>? cameras;
  XFile? capturedImage;

  @override
  void initState() {
    super.initState();
    _initCamera();
  }

  Future<void> _initCamera() async {
    cameras = await availableCameras();
    controller = CameraController(cameras![0], ResolutionPreset.medium);
    await controller!.initialize();
    setState(() {});
  }

  Future<void> _capturePhoto() async {
    final image = await controller!.takePicture();
```

```
  setState(() {
    capturedImage = image;
  });
}

@override
void dispose() {
  controller?.dispose();
  super.dispose();
}

@override
Widget build(BuildContext context) {
  if (controller == null || !controller!.value.isInitialized) {
    return const Scaffold(
      body: Center(child: CircularProgressIndicator()),
    );
  }

  return Scaffold(
    appBar: AppBar(title: const Text("Kamera Aktif")),
    body: Column(
      children: [
        Expanded(flex: 2, child: CameraPreview(controller!)),
        ElevatedButton.icon(
          onPressed: _capturePhoto,
```

```
          icon: const Icon(Icons.camera),
          label: const Text("Ambil Foto"),
        ),
        if (capturedImage != null)
          Expanded(
            flex: 1,
            child: Image.file(
              File(capturedImage!.path),
              fit: BoxFit.cover,
            ),
          ),
      ],
    ),
  );
 }
}
```

Penjelasan Fungsi Kode:

- availableCameras() → Mendapatkan daftar kamera yang tersedia (depan/belakang).
- CameraController → Mengatur preview dan pengambilan foto.
- takePicture() → Menangkap gambar dari kamera.
- CameraPreview() → Menampilkan tampilan langsung dari kamera di layar.

Hasil yang diharapkan:
Saat tombol **"Gunakan Kamera"** diklik → kamera aktif → tampil live preview → ketika klik **Ambil Foto**, hasil foto muncul di bawah.

E. File accelerometer_page.dart

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

```
import 'package:sensors_plus/sensors_plus.dart';

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

  @override
  State<AccelerometerPage> createState() => _AccelerometerPageState();
}

class _AccelerometerPageState extends State<AccelerometerPage> {
  double x = 0, y = 0, z = 0;

  @override
  void initState() {
    super.initState();
    accelerometerEvents.listen((AccelerometerEvent event) {
      setState(() {
        x = event.x;
        y = event.y;
        z = event.z;
      });
    });
  }

  @override
  Widget build(BuildContext context) {
```

```
  return Scaffold(
    appBar: AppBar(title: const Text("Sensor Accelerometer")),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Text("X: ${x.toStringAsFixed(2)}"),
          Text("Y: ${y.toStringAsFixed(2)}"),
          Text("Z: ${z.toStringAsFixed(2)}"),
          const SizedBox(height: 30),
          const Text(
            "Coba gerakkan emulator (extend → Virtual Sensors)\nuntuk melihat nilai berubah!",
            textAlign: TextAlign.center,
          ),
        ],
      ),
    ),
  );
 }
}
```

🧠 **Penjelasan Fungsi Kode:**

- accelerometerEvents.listen() → Mendengarkan perubahan nilai sensor.
- Nilai **X, Y, Z** menunjukkan arah gravitasi atau gerakan perangkat.
- Di emulator, buka:
  Extended Controls → Sensors → Accelerometer
  lalu gerakkan slider — maka nilai X/Y/Z akan berubah secara real-time.

🧩 F. Uji Coba Aplikasi

1. Jalankan perintah:
2. flutter run
3. Tekan tombol:
    - **Gunakan Kamera** → ambil foto.
    - **Lihat Accelerometer** → buka virtual sensor dan lihat nilai berubah.

🎓 Tugas Akhir

Tambahkan satu tombol baru:

- **“Gunakan API Cuaca”**, yang memanggil API cuaca (mis. OpenWeatherMap) dan menampilkan suhu saat ini berdasarkan lokasi GPS mahasiswa.