LAPORAN PRAKTIKUM
MATA KULIAH PEMROGRAMAN MOBILE

U N I V E R S I T A S
AMIKOM PURWOKERTO

Oleh :

NAMA: FAJAR RAMADHAN

NIM: 23SA31A065

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER

UNIVERSITAS AMIKOM

PURWOKERTO

2025

PRAKTIKUM 4 OOP DART

Pada praktikum ke 4 ini mempelajari tentang konsep Object Oriented Programming (OOP) dalam bahasa Dart yang menjadi basis dari Flutter. Pada program yang dibuat pada praktikum ini mengimplementasikan dua konsep utama dari OOP adalah Inheritance pada class Tabung yang mewarisi properties dan method class Lingkaran, dan implementasi konsep Encapsulation dimana pada properties class Lingkaran dan Tabung dibuat private dan diberi getter dan setter walaupun getter dan setter pada praktikum ini memang termasuk unnecessary_getters_setters jika mengacu pada dokumentasi resmi dart karena hanya forwarding akses properties.

1. Kode Stateless widget dan main

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

void main(List<String> args) {
  runApp(const MyApp());
}

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

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

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

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}
```

2. Kode statefull widget

```dart
class Lingkaran {
  double? _jariJari;
  Lingkaran([this._jariJari]);

  double? get getJariJari => _jariJari;

  set setJariJari(jariJari) {
    _jariJari = jariJari;
  }

  double getLuasLingkaran() {
    return 22 / 7 * _jariJari! * _jariJari!;
  }

  double getKelilingLingkaran() => 2 * 22 / 7 * _jariJari!;
}

class Tabung extends Lingkaran {
  double? _tinggi;
  Tabung([super._jariJari, this._tinggi]);

  double? get getTinggi => _tinggi;

  set setTinggi(tinggi) {
    _tinggi = tinggi;
  }

  double getVolumeTabung() => getLuasLingkaran() * _tinggi!;

  double getLuasSelimutTabung() => getKelilingLingkaran() * _tinggi!;

  double getLuasPermukaanTabung() =>
      getLuasSelimutTabung() + 2 * getLuasLingkaran();
}
```

3. Kode Class dan logic perhitungan Lingkaran dan Tabung

```dart
class Lingkaran {
  double? _jariJari;
  Lingkaran([this._jariJari]);

  double? get getJariJari => _jariJari;

  set setJariJari(jariJari) {
    _jariJari = jariJari;
  }

  double getLuasLingkaran() {
    return 22 / 7 * _jariJari! * _jariJari!;
  }

  double getKelilingLingkaran() => 2 * 22 / 7 * _jariJari!;
}

class Tabung extends Lingkaran {
  double? _tinggi;
  Tabung([super._jariJari, this._tinggi]);

  double? get getTinggi => _tinggi;

  set setTinggi(tinggi) {
    _tinggi = tinggi;
  }

  double getVolumeTabung() => getLuasLingkaran() * _tinggi!;

  double getLuasSelimutTabung() => getKelilingLingkaran() * _tinggi!;

  double getLuasPermukaanTabung() =>
      getLuasSelimutTabung() + 2 * getLuasLingkaran();
}
```

4. Hasil