

Nama : Muhammad Nur Ramadhan

NIM : 2309106053

PEMROGRAMAN PIRANTI BERGERAK

POSTTEST 1

LinkDemo:

<https://drive.google.com/file/d/1I6WmXWl9hI9D1MA1bepaYF9nzgAYGepW/view?usp=sharing>

```
import 'dart:io';
```

```
//Mengimpor library bawaan Dart untuk input/output melalui terminal.
```

```
//Digunakan untuk stdin.readLineSync() (baca input) dan stdout.write()/print() (tampilkan output).
```

```
// Class
```

```
class Resep {
```

```
  final String id;
```

```
  String judul;
```

```
  List<String> bahan;
```

```
  List<String> langkah;
```

```
  String? kategori; // nullable
```

```
  double? rating; // -----
```

```
//Membuat class Resep sebagai blueprint resep.
```

```
//id final, tidak bisa diubah setelah dibuat.
```

```
//judul, bahan, Langkah, wajib diisi.
```

```
//kategori dan rating, nullable (String?, double?) → penerapan Null Safety.
```

```
  Resep({
```

```
    required this.id,
```

```
required this.judul,  
required this.bahan,  
required this.langkah,  
this.kategori,  
this.rating,  
});
```

//Constructor dengan parameter required, harus diisi saat membuat objek.

//kategori dan rating opsional.

```
void tampilkanDetail() {  
    print("\n=== Detail Resep ===");  
    print("Judul    : $judul");  
    print("Kategori : ${kategori ?? "Belum ditentukan"}"); // Null Conditional  
    print("Bahan     : ${bahan.join(', ')}");  
    print("Langkah    : ${langkah.join(' -> ')}");  
    print("Rating     : ${rating?.toStringAsFixed(1) ?? "Belum dinilai"}"); // Null Conditional
```

//Menampilkan detail resep ke layar.

//kategori ?? "Belum ditentukan" → Null Conditional + default.

//rating?.toStringAsFixed(1) ?? "Belum dinilai" → kalau rating null, tampilkan teks default.

```
// Ternary operator  
String status = (rating != null && rating! >= 4.0)  
    ? "Rekomendasi 🍌 "  
    : "Biasa saja 🍷 ";  
print("Status    : $status");
```

```
}
```

```
}
```

//Jika rating $\geq 4.0 \rightarrow$ status “Rekomendasi 👍”.

//Jika tidak \rightarrow status “Biasa saja 🤔”.

```
class ResepManager {
```

```
    final List<Resep> _resepList = [];
```

//ResepManager bertugas mengelola koleksi resep.

//_resepList menyimpan semua resep (list of Resep).

```
    void tambahResep(Resep r) => _resepList.add(r);
```

```
    void tampilkanSemua() {
```

```
        if (_resepList.isEmpty) {
```

```
            print("Belum ada resep tersimpan.");
```

```
            return;
```

```
        }
```

```
        print("\n=== Daftar Resep ===");
```

```
        for (var r in _resepList) {
```

```
            print("- ${r.judul} (${r.kategori ?? "Umum"})");
```

```
        }
```

```
    }
```

//Jika list kosong maka tampilkan pesan.

//Kalau ada isi maka print semua judul resep, default kategori = "Umum" bila null.

```
    Resep? cariResep(String judul) {
```

```

    try {
        return _resepList.firstWhere(
            (r) => r.judul.toLowerCase() == judul.toLowerCase(),
        );
    } catch (e) {
        return null;
    }
}
}
}

```

Mencari resep berdasarkan judul (case-insensitive).

```

// CLI

void main() {
    final manager = ResepManager();

    while (true) {
        print("\n=== MENU KATALOG RESEP ===");
        print("1. Tambah Resep");
        print("2. Lihat Semua Resep");
        print("3. Cari Resep");
        print("4. Keluar");

        stdout.write("Pilih menu: ");
        String? pilihan = stdin.readLineSync();
    }
}

```

//stdout.write untuk menampilkan teks tanpa enter otomatis.

//stdin.readLineSync() untuk membaca input user.

```

switch (pilihan) {
    case '1':
        stdout.write("Judul Resep: ");
        String? judul = stdin.readLineSync();

        stdout.write("Bahan (pisahkan dengan koma): ");
        String? inputBahan = stdin.readLineSync();
        List<String> bahan =
            inputBahan?.split(',').map((e) => e.trim()).toList() ?? [];
//User memasukkan judul, bahan, langkah.
//Input dipisah koma , lalu diubah jadi list.
//Kalau null, hasil default = list kosong.

        stdout.write("Langkah (pisahkan dengan koma): ");
        String? inputLangkah = stdin.readLineSync();
        List<String> langkah =
            inputLangkah?.split(',').map((e) => e.trim()).toList() ?? [];

        stdout.write("Kategori (opsional): ");
        String? kategori = stdin.readLineSync();
        kategori =
            (kategori != null && kategori.isNotEmpty) ? kategori : null;
//kategori hanya disimpan jika tidak kosong.
//Kalau kosong, null.

```

```
stdout.write("Rating (opsional, 0-5): ");
```

```
String? ratingStr = stdin.readLineSync();
```

```
double? rating = double.tryParse(ratingStr ?? "");
```

//rating bisa gagal parsing, null jika tidak valid.

```
var resep = Resep(
```

```
  id: DateTime.now().millisecondsSinceEpoch.toString(),
```

```
  judul: judul ?? "Tanpa Judul",
```

```
  bahan: bahan,
```

```
  langkah: langkah,
```

```
  kategori: kategori,
```

```
  rating: rating,
```

```
);
```

```
manager.tambahResep(resep);
```

```
print("Resep berhasil ditambahkan!");
```

```
break;
```

```
case '2':
```

```
  manager.tampilkanSemua();
```

```
  break;
```

```
case '3':
```

```
  stdout.write("Masukkan judul resep: ");
```

```
  String? cari = stdin.readLineSync();
```

```
  var hasil = manager.cariResep(cari ?? "");
```

```
  if (hasil != null) {
```

```
        hasil.tampilkanDetail();
    } else {
        print("Resep tidak ditemukan.");
    }
    break;

//Minta input judul.
//Cari dengan manager.cariResep().
//Kalau ada maka tampilkan detail.
//Kalau tidak ada maka print resep tidak ditemukan.
```

```
    case '4':
        print("Terima kasih, keluar aplikasi.");
        return;
```

```
    default:
        print("Pilihan tidak valid!");
    }
}
}
```