# **Codelab 1**

Pada kelas ini kita akan mengembangkan sebuah aplikasi yang menampilkan tempat-tempat wisata di Bandung. Hasil akhir dari keseluruhan codelab akan seperti berikut:

Dalam codelab pertama ini kita akan membuat sebuah tampilan yang menggabungkan semua widget-widget yang sebelumnya kita pelajari. Tampilannya adalah seperti berikut:

Sebelum kita membuat tampilan di atas, kita akan bedah terlebih dahulu *layout*-nya.

Pada layout di atas kita dapat memetakan widget-widget dalam bentuk diagram seperti di bawah ini:

1. Buat project Flutter baru dan berikan nama yang sesuai, misalnya **wisatabandung**. Hapus kode aplikasi counter yang diberikan ketika project dibuat.
2. Tuliskan kode dasar yang menampilkan widget MaterialAppseperti berikut:
   * import 'package:flutter/material.dart';
   * void main() => runApp(const MyApp());
   * class MyApp extends StatelessWidget {
   * const MyApp({Key? key}) : super(key: key);
   * @override
   * Widget build(BuildContext context) {
   * return MaterialApp(
   * title: 'Wisata Bandung',
   * theme: ThemeData(),
   * );
   * }
   * }
3. Lalu kita akan membuat kode untuk susunan widget sesuai diagram yang telah kita buat. Untuk membuat kode kita lebih rapi kita akan membuat kelas Stateless Widget baru untuk menampung kode tampilan kita. Mari namakan kelas ini DetailScreen.
   * class DetailScreen extends StatelessWidget {
   * const DetailScreen({Key? key}) : super(key: key);
   * @override
   * Widget build(BuildContext context) {
   * return Scaffold();
   * }
   * }
4. Jangan lupa untuk menambahkan widget DetailScreen sebagai home dari MaterialApp.
   * class MyApp extends StatelessWidget {
   * const MyApp({Key? key}) : super(key: key);
   * @override
   * Widget build(BuildContext context) {
   * return MaterialApp(
   * title: 'Wisata Bandung',
   * theme: ThemeData(),
   * **home: const DetailScreen(),**
   * );
   * }
   * }
5. Sesuai diagram di atas, kita akan menyusun beberapa widget secara vertikal sehingga kita perlu menggunakan widget Column.
   * class DetailScreen extends StatelessWidget {
   * const DetailScreen({Key? key}) : super(key: key);
   * @override
   * Widget build(BuildContext context) {
   * return Scaffold(
   * **body: Column(),**
   * );
   * }
   * }
6. Jalankan aplikasi Anda. Saat ini device atau emulator Anda memang masih belum menampilkan apa pun. Namun, kita akan memanfaatkan fitur hot reload untuk melihat perubahan-perubahan yang akan kita lakukan ke depan.
7. Komponen pertama yang akan kita buat adalah bagian judul dari halaman. Tentunya untuk menampilkan teks kita akan menggunakan widget Text.
   * class DetailScreen extends StatelessWidget {
   * const DetailScreen({Key? key}) : super(key: key);
   * @override
   * Widget build(BuildContext context) {
   * return Scaffold(
   * body: Column(
   * **children: <Widget>[**
   * **const Text('Farm House Lembang'),**
   * **],**
   * ),
   * );
   * }
   * }
8. Ketika Anda menyimpan project atau menjalankan hot reload, tampilan aplikasi Anda sekarang mungkin tidak sesuai dengan keinginan, seperti teks terlalu ke atas dan juga terlalu kecil.  
   Untuk itulah kita perlu membungkus widget Text ke dalam Container supaya kita dapat memberikan property seperti margin atau padding. Jika Anda menggunakan IDE Android Studio, Anda dapat memanfaatkan shortcut **Alt+Enter** untuk membungkus widget ke widget lain.
9. Tambahkan margin atas supaya teks memiliki jarak terhadap bagian atas layar.
   * class DetailScreen extends StatelessWidget {
   * const DetailScreen({Key? key}) : super(key: key);
   * @override
   * Widget build(BuildContext context) {
   * return Scaffold(
   * body: Column(
   * children: <Widget>[
   * Container(
   * **margin: const EdgeInsets.only(top: 16.0),**
   * child: const Text('Farm House Lembang'),
   * ),
   * ],
   * ),
   * );
   * }
   * }
10. Pada kode di atas kita hanya memberikan margin atas sebesar sebesar 16.0. Anda dapat memanfaatkan metode EdgeInsets lain seperti all() untuk memberikan margin ke semua sisi atau symmetric() apabila Anda ingin memberikan margin ke sisi vertikal atau horizontal.
11. Jika Anda kesulitan menentukan margin atas, khususnya pada perangkat yang memiliki notch yang umumnya memiliki status bar yang lebih besar, Anda dapat memanfaatkan widget SafeArea.
    * class DetailScreen extends StatelessWidget {
    * const DetailScreen({Key? key}) : super(key: key);
    * @override
    * Widget build(BuildContext context) {
    * return Scaffold(
    * body: **SafeArea**(
    * child: Column(
    * children: <Widget>[
    * Container(
    * margin: const EdgeInsets.only(top: 16.0),
    * child: const Text('Farm House Lembang'),
    * )
    * ],
    * ),
    * ),
    * );
    * }
    * }
12. Widget ini akan memberikan padding yang secara otomatis menyesuaikan perangkat yang digunakan.
13. Selanjutnya, sesuai contoh kita akan membuat teks judul berada di tengah. Tambahkan parameter atau properti textAlign pada widget Text. Selain itu, tambahkan juga style dengan memperbesar ukuran teks agar tulisan dapat dibaca.
    * Container(
    * margin: EdgeInsets.only(top: 16.0),
    * child: const Text(
    * 'Farm House Lembang',
    * **textAlign: TextAlign.center,**
    * **style: TextStyle(**
    * **fontSize: 30.0,**
    * **fontWeight: FontWeight.bold,**
    * **),**
    * ),
    * ),
14. Lakukan hot reload. Tidak ada perubahan, apa sebabnya? Jika menggunakan Android Studio Anda dapat memanfaatkan fitur Flutter Inspector untuk melihat layout widget di dalam aplikasi.
15. Dari gambar di atas bisa kita lihat ternyata layout aplikasi kita tidak penuh hingga seluruh halaman. Ini disebabkan sisi horizontal dari Column hanya menyesuaikan dengan konten yang ada di dalamnya. Untuk memaksimalkan ukuran lebar dari Column, tambahkan kode berikut:
    * body: SafeArea(
    * child: Column(
    * **crossAxisAlignment: CrossAxisAlignment.stretch,**
    * children: <Widget>[
    * Container(
    * margin: EdgeInsets.only(top: 16.0),
    * child: Text(
    * 'Farm House Lembang',
    * textAlign: TextAlign.center,
    * style: TextStyle(
    * fontSize: 30.0,
    * fontWeight: FontWeight.bold,
    * ),
    * ),
    * ),
    * ],
    * ),
    * ),
16. Setelah menyelesaikan judul, selanjutnya kita akan membuat bagian kedua yaitu informasi dari tempat wisata.
17. Seperti yang terlihat kita perlu menyusun widget secara horizontal dan vertikal. Mari tambahkan child kedua dari Column dengan sebuah Container berisi Row. Tambahkan juga margin pada sisi atas dan bawah untuk memberikan jarak antar widget.
    * class DetailScreen extends StatelessWidget {
    * const DetailScreen({Key? key}) : super(key: key);
    * @override
    * Widget build(BuildContext context) {
    * return Scaffold(
    * body: SafeArea(
    * child: Column(
    * crossAxisAlignment: CrossAxisAlignment.stretch,
    * children: <Widget>[
    * Container(...),
    * **Container(**
    * **margin: const EdgeInsets.symmetric(vertical: 16.0),**
    * **child: Row(**
    * **children: <Widget>[],**
    * **),**
    * **),**
    * ],
    * ),
    * ),
    * );
    * }
    * }
18. Buat widget Column untuk menyusun Icon dan Text.
    * class DetailScreen extends StatelessWidget {
    * const DetailScreen({Key? key}) : super(key: key);
    * @override
    * Widget build(BuildContext context) {
    * return Scaffold(
    * body: SafeArea(
    * child: Column(
    * crossAxisAlignment: CrossAxisAlignment.stretch,
    * children: <Widget>[
    * Container(...),
    * Container(
    * margin: EdgeInsets.symmetric(vertical: 16.0),
    * child: Row(
    * children: <Widget>[
    * **Column(**
    * **children: const <Widget>[**
    * **Icon(Icons.calendar\_today),**
    * **Text('Open Everyday'),**
    * **],**
    * **),**
    * ],
    * ),
    * ),
    * ],
    * ),
    * ),
    * );
    * }
    * }
19. Jika Anda merasa jarak antara Icon dan Text terlalu rapat, Anda dapat menambahkan widget SizedBoxuntuk membuat “kotak” yang berguna untuk memberikan jarak.
    * Column(
    * children: const <Widget>[
    * Icon(Icons.calendar\_today),
    * **SizedBox(height: 8.0),**
    * Text('Open Everyday'),
    * ],
    * ),
20. Selanjutnya sebagai tantangan, lengkapilah informasi tempat wisata dengan pasangan ikon dan teks sesuai contoh yang diberikan.
21. Untuk menyusun Row seperti di atas, pastikan menggunakan mainAxisAlignment seperti ini:
    * class DetailScreen extends StatelessWidget {
    * const DetailScreen({Key? key}) : super(key: key);
    * @override
    * Widget build(BuildContext context) {
    * return Scaffold(
    * body: SafeArea(
    * child: Column(
    * crossAxisAlignment: CrossAxisAlignment.stretch,
    * children: <Widget>[
    * Container(...),
    * Container(
    * margin: const EdgeInsets.symmetric(vertical: 16.0),
    * child: Row(
    * **mainAxisAlignment: MainAxisAlignment.spaceEvenly,**
    * children: <Widget>[
    * Column(...),
    * Column(...),
    * Column(...)
    * ],
    * ),
    * ),
    * ],
    * ),
    * ),
    * );
    * }
    * }
22. Pada langkah ini harusnya Anda sudah bisa menampilkan teks deskripsi sesuai langkah yang diberikan sebelumnya. Anda cukup menambahkan widget Container dan Textuntuk menampilkan konten deskripsi. Anda juga dapat menambahkan style sesuai selera Anda.
    * Container(
    * padding: const EdgeInsets.all(16.0),
    * child: const Text(
    * 'Berada di jalur utama Bandung-Lembang, Farm House menjadi objek wisata yang tidak pernah sepi pengunjung. Selain karena letaknya strategis, kawasan ini juga menghadirkan nuansa wisata khas Eropa. Semua itu diterapkan dalam bentuk spot swafoto Instagramable.',
    * textAlign: TextAlign.center,
    * style: TextStyle(fontSize: 16.0),
    * ),
    * ),
23. *Yey*, Anda berhasil menyusun widget dengan baik. Tampilan aplikasi akan seperti gambar berikut.  
    
24. Keseluruhan kode Anda akan seperti berikut:
    * import 'package:flutter/material.dart';
    * void main() => runApp(const MyApp());
    * class MyApp extends StatelessWidget {
    * const MyApp({Key? key}) : super(key: key);
    * @override
    * Widget build(BuildContext context) {
    * return MaterialApp(
    * title: 'Wisata Bandung',
    * theme: ThemeData(),
    * home: const DetailScreen(),
    * );
    * }
    * }
    * class DetailScreen extends StatelessWidget {
    * const DetailScreen({Key? key}) : super(key: key);
    * @override
    * Widget build(BuildContext context) {
    * return Scaffold(
    * body: SafeArea(
    * child: Column(
    * crossAxisAlignment: CrossAxisAlignment.stretch,
    * children: <Widget>[
    * Container(
    * margin: const EdgeInsets.only(top: 16.0),
    * child: const Text(
    * 'Farm House Lembang',
    * textAlign: TextAlign.center,
    * style: TextStyle(
    * fontSize: 30.0,
    * fontWeight: FontWeight.bold,
    * ),
    * ),
    * ),
    * Container(
    * margin: const EdgeInsets.symmetric(vertical: 16.0),
    * child: Row(
    * mainAxisAlignment: MainAxisAlignment.spaceEvenly,
    * children: <Widget>[
    * Column(
    * children: const <Widget>[
    * Icon(Icons.calendar\_today),
    * SizedBox(height: 8.0),
    * Text('Open Everyday'),
    * ],
    * ),
    * Column(
    * children: const <Widget>[
    * Icon(Icons.access\_time),
    * SizedBox(height: 8.0),
    * Text('09:00 - 20:00')
    * ],
    * ),
    * Column(
    * children: const <Widget>[
    * Icon(Icons.monetization\_on),
    * SizedBox(height: 8.0),
    * Text('Rp 25.000'),
    * ],
    * )
    * ],
    * ),
    * ),
    * Container(
    * padding: const EdgeInsets.all(16.0),
    * child: const Text(
    * 'Berada di jalur utama Bandung-Lembang, Farm House menjadi objek wisata yang tidak pernah sepi pengunjung. Selain karena letaknya strategis, kawasan ini juga menghadirkan nuansa wisata khas Eropa. Semua itu diterapkan dalam bentuk spot swafoto Instagramable.',
    * textAlign: TextAlign.center,
    * style: TextStyle(fontSize: 16.0),
    * ),
    * )
    * ],
    * ),
    * ),
    * );
    * }
    * }

Anda juga dapat mengunduh keseluruhan kodenya pada tautan berikut:

* [Codelab-1](https://github.com/dicodingacademy/a159-flutter-pemula-labs/tree/main/wisatabandung/codelab-1)