**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 wisata\_bandung. Hapus kode aplikasi counter yang diberikan ketika project dibuat.
2. Tuliskan kode dasar yang menampilkan widget MaterialAppseperti berikut:
   1. import 'package:flutter/material.dart';
   3. void main() => runApp(MyApp());
   5. class MyApp extends StatelessWidget {
   6. @override
   7. Widget build(BuildContext context) {
   8. return MaterialApp(
   9. title: 'Wisata Bandung',
   10. theme: ThemeData(),
   11. );
   12. }
   13. }
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.
   1. class DetailScreen extends StatelessWidget {
   2. @override
   3. Widget build(BuildContext context) {
   4. return Scaffold();
   5. }
   6. }

Jangan lupa untuk menambahkan widget DetailScreen sebagai home dari MaterialApp.

* 1. class MyApp extends StatelessWidget {
  2. @override
  3. Widget build(BuildContext context) {
  4. return MaterialApp(
  5. title: 'Wisata Bandung',
  6. theme: ThemeData(),
  7. home: DetailScreen(),
  8. );
  9. }
  10. }

1. Sesuai diagram di atas, kita akan menyusun beberapa widget secara vertikal sehingga kita perlu menggunakan widget Column.
   1. class DetailScreen extends StatelessWidget {
   2. @override
   3. Widget build(BuildContext context) {
   4. return Scaffold(
   5. body: Column(),
   6. );
   7. }
   8. }

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.

1. Komponen pertama yang akan kita buat adalah bagian judul dari halaman. Tentunya untuk menampilkan teks kita akan menggunakan widget Text.
   1. class DetailScreen extends StatelessWidget {
   2. @override
   3. Widget build(BuildContext context) {
   4. return Scaffold(
   5. body: Column(
   6. children: <Widget>[
   7. Text('Farm House Lembang'),
   8. ],
   9. ),
   10. );
   11. }
   12. }
2. 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.
3. Tambahkan margin atas supaya teks memiliki jarak terhadap bagian atas layar.
   1. class DetailScreen extends StatelessWidget {
   2. @override
   3. Widget build(BuildContext context) {
   4. return Scaffold(
   5. body: Column(
   6. children: <Widget>[
   7. Container(
   8. **margin: EdgeInsets.only(top: 16.0),**
   9. child: Text('Farm House Lembang'),
   10. ),
   11. ],
   12. ),
   13. );
   14. }
   15. }

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.

1. 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.
   1. class DetailScreen extends StatelessWidget {
   2. @override
   3. Widget build(BuildContext context) {
   4. return Scaffold(
   5. body: SafeArea(
   6. child: Column(
   7. children: <Widget>[
   8. Container(
   9. margin: EdgeInsets.only(top: 16.0),
   10. child: Text('Farm House Lembang'),
   11. )
   12. ],
   13. ),
   14. ),
   15. );
   16. }
   17. }

Widget ini akan memberikan padding yang secara otomatis menyesuaikan perangkat yang digunakan.

1. Selanjutnya, sesuai contoh kita akan membuat teks judul berada di tengah. Tambahkan parameter atau properti textAlign pada widget Text.
   1. Container(
   2. margin: EdgeInsets.only(top: 16.0),
   3. child: Text(
   4. 'Farm House Lembang',
   5. **textAlign: TextAlign.center,**
   6. style: TextStyle(
   7. fontSize: 30.0,
   8. fontWeight: FontWeight.bold,
   9. ),
   10. ),
   11. ),

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.  
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:

* 1. body: SafeArea(
  2. child: Column(
  3. **crossAxisAlignment: CrossAxisAlignment.stretch,**
  4. children: <Widget>[
  5. Container(
  6. margin: EdgeInsets.only(top: 16.0),
  7. child: Text(
  8. 'Farm House Lembang',
  9. textAlign: TextAlign.center,
  10. style: TextStyle(
  11. fontSize: 30.0,
  12. fontWeight: FontWeight.bold,
  13. ),
  14. ),
  15. ),
  16. ],
  17. ),
  18. ),

1. Setelah menyelesaikan judul, selanjutnya kita akan membuat bagian kedua yaitu informasi dari tempat wisata.  
   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.
   1. class DetailScreen extends StatelessWidget {
   2. @override
   3. Widget build(BuildContext context) {
   4. return Scaffold(
   5. body: SafeArea(
   6. child: Column(
   7. crossAxisAlignment: CrossAxisAlignment.stretch,
   8. children: <Widget>[
   9. Container(...),
   10. **Container(**
   11. **margin: EdgeInsets.symmetric(vertical: 16.0),**
   12. **child: Row(**
   13. **children: <Widget>[],**
   14. **),**
   15. **),**
   16. ],
   17. ),
   18. ),
   19. );
   20. }
   21. }
2. Buat widget Column untuk menyusun Icon dan Text.
   1. class DetailScreen extends StatelessWidget {
   2. @override
   3. Widget build(BuildContext context) {
   4. return Scaffold(
   5. body: SafeArea(
   6. child: Column(
   7. crossAxisAlignment: CrossAxisAlignment.stretch,
   8. children: <Widget>[
   9. Container(...),
   10. Container(
   11. margin: EdgeInsets.symmetric(vertical: 16.0),
   12. child: Row(
   13. children: <Widget>[
   14. **Column(**
   15. **children: <Widget>[**
   16. **Icon(Icons.calendar\_today),**
   17. **Text('Open Everyday'),**
   18. **],**
   19. **),**
   20. ],
   21. ),
   22. ),
   23. ],
   24. ),
   25. ),
   26. );
   27. }
   28. }

Jika Anda merasa jarak antara Icon dan Text terlalu rapat, Anda dapat menambahkan widget SizedBox untuk membuat “kotak” yang berguna untuk memberikan jarak.

* 1. Column(
  2. children: <Widget>[
  3. Icon(Icons.calendar\_today),
  4. **SizedBox(height: 8.0),**
  5. Text('Open Everyday'),
  6. ],
  7. ),

1. Selanjutnya sebagai tantangan, lengkapilah informasi tempat wisata dengan pasangan ikon dan teks sesuai contoh yang diberikan.
2. Untuk menyusun Row seperti di atas, pastikan menggunakan mainAxisAlignment seperti ini:
   1. class DetailScreen extends StatelessWidget {
   2. @override
   3. Widget build(BuildContext context) {
   4. return Scaffold(
   5. body: SafeArea(
   6. child: Column(
   7. crossAxisAlignment: CrossAxisAlignment.stretch,
   8. children: <Widget>[
   9. Container(...),
   10. Container(
   11. margin: EdgeInsets.symmetric(vertical: 16.0),
   12. child: Row(
   13. **mainAxisAlignment: MainAxisAlignment.spaceEvenly,**
   14. children: <Widget>[
   15. Column(...),
   16. Column(...),
   17. Column(...)
   18. ],
   19. ),
   20. ),
   21. ],
   22. ),
   23. ),
   24. );
   25. }
   26. }
3. 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.
   1. Container(
   2. padding: EdgeInsets.all(16.0),
   3. child: Text(
   4. '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.',
   5. textAlign: TextAlign.center,
   6. style: TextStyle(fontSize: 16.0),
   7. ),
   8. ),
4. Keseluruhan kode Anda akan seperti berikut:
   1. import 'package:flutter/material.dart';
   3. void main() => runApp(MyApp());
   5. class MyApp extends StatelessWidget {
   6. @override
   7. Widget build(BuildContext context) {
   8. return MaterialApp(
   9. title: 'Wisata Bandung',
   10. theme: ThemeData(),
   11. home: DetailScreen(),
   12. );
   13. }
   14. }
   16. class DetailScreen extends StatelessWidget {
   17. @override
   18. Widget build(BuildContext context) {
   19. return Scaffold(
   20. body: SafeArea(
   21. child: Column(
   22. crossAxisAlignment: CrossAxisAlignment.stretch,
   23. children: <Widget>[
   24. Container(
   25. margin: EdgeInsets.only(top: 16.0),
   26. child: Text(
   27. 'Farm House Lembang',
   28. textAlign: TextAlign.center,
   29. style: TextStyle(
   30. fontSize: 30.0,
   31. fontWeight: FontWeight.bold,
   32. ),
   33. ),
   34. ),
   35. Container(
   36. margin: EdgeInsets.symmetric(vertical: 16.0),
   37. child: Row(
   38. mainAxisAlignment: MainAxisAlignment.spaceEvenly,
   39. children: <Widget>[
   40. Column(
   41. children: <Widget>[
   42. Icon(Icons.calendar\_today),
   43. SizedBox(height: 8.0),
   44. Text('Open Everyday'),
   45. ],
   46. ),
   47. Column(
   48. children: <Widget>[
   49. Icon(Icons.access\_time),
   50. SizedBox(height: 8.0),
   51. Text('09:00 - 20:00'),
   52. ],
   53. ),
   54. Column(
   55. children: <Widget>[
   56. Icon(Icons.monetization\_on),
   57. SizedBox(height: 8.0),
   58. Text('RP 25.000'),
   59. ],
   60. ),
   61. ],
   62. ),
   63. ),
   64. Container(
   65. padding: EdgeInsets.all(16.0),
   66. child: Text(
   67. '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.',
   68. textAlign: TextAlign.center,
   69. style: TextStyle(fontSize: 16.0),
   70. ),
   71. )
   72. ],
   73. ),
   74. ),
   75. );
   76. }
   77. }

Anda juga dapat mengunduh keseluruhan kodenya pada tautan berikut: <https://github.com/dicodingacademy/a159-flutter-pemula-labs>

## Codelab 2: Perbaikan Tampilan Aplikasi

Setelah mempelajari beberapa materi tambahan, sekarang saatnya kita melanjutkan project aplikasi wisata kita. Pada codelab ini kita akan membuat aplikasi dengan tampilan seperti berikut:

1. Mari kita mulai dengan membuka dan melanjutkan codelab kita sebelumnya.
2. Untuk memudahkan dalam membaca sekaligus merapikan kode, mari kita pindahkan widget atau kelas **DetailScreen** ke sebuah file dart baru. Anda dapat membuat file baru dengan cara **klik kanan pada folder lib -> New -> Dart File**. Berikan nama **detail\_screen.dart**.
3. Anda akan mendapati beberapa eror akibat adanya library atau package yang belum terpasang. Pada file **detail\_screen.dart** tambahkan kode **import** berikut di baris paling atas untuk menggunakan package material design di dalam file.
   1. import 'package:flutter/material.dart';
4. Selanjutnya karena kita akan menggunakan file widget **DetailScreen** di file **main.dart**, maka kita juga perlu melakukan **import** berkas **detail\_screen.dart** ke dalam berkasi **main.dart**.
   1. import 'package:wisatabandung/detail\_screen.dart';
5. Kemudian kita akan menambahkan sebuah gambar ke tampilan paling atas halaman. Gambar ini akan kita ambil dari asset. Untuk itu, kita perlu menambahkan berkas yang ingin ditampilkan ke dalam project dan menambahkannya pada file **pubspec.yaml**. Aset gambar dapat Anda unduh pada tautan [berikut](https://github.com/dicodingacademy/assets/raw/main/flutter_pemula_academy/assets_wisata.zip).
   1. flutter:
   2. uses-material-design: true
   3. assets:
   4. - images/

Tambahkan widget Image di child paling atas dari Column.

* 1. class DetailScreen extends StatelessWidget {
  2. @override
  3. Widget build(BuildContext context) {
  4. return Scaffold(
  5. body: SafeArea(
  6. child: Column(
  7. crossAxisAlignment: CrossAxisAlignment.stretch,
  8. children: <Widget>[
  9. **Image.asset('images/farm-house.jpg'),**
  10. Container(...),
  11. Container(...),
  12. Container(...),
  13. ],
  14. ),
  15. ),
  16. );
  17. }
  18. }

Jalankan aplikasi Anda untuk melihat perubahan.

1. Selanjutnya kita akan menampilkan beberapa gambar lagi di bagian bawah. Kali ini kita akan mengambil gambar melalui url. Mari kita mulai dengan satu gambar terlebih dahulu.
   1. class DetailScreen extends StatelessWidget {
   2. @override
   3. Widget build(BuildContext context) {
   4. return Scaffold(
   5. body: SafeArea(
   6. child: Column(
   7. crossAxisAlignment: CrossAxisAlignment.stretch,
   8. children: <Widget>[
   9. Image.asset('images/farm-house.jpg'),
   10. Container(...),
   11. Container(...),
   12. Container(...),
   13. **Image.network(**
   14. **'https://media-cdn.tripadvisor.com/media/photo-s/0d/7c/59/70/farmhouse-lembang.jpg'),**
   15. **],**
   16. ),
   17. ),
   18. );
   19. }
   20. }

Apabila gambar yang kita tampilkan terlalu besar sementara layar pada perangkat terlalu kecil, maka akan terlihat tampilan garis hitam-kuning yang menunjukkan terjadi overflow. Kondisi overflow ini terjadi ketika konten yang kita tampilkan melebihi luas layar yang ada.

1. Sebagai solusi, tentunya kita bisa mengubah ukuran dari gambar, namun tentunya tidak praktis jika kita harus mengubah ukuran setiap gambar yang ditampilkan. Tentu ada banyak sekali ukuran layar yang tersedia, bukan? Solusi lainnya yaitu dengan menerapkan scrolling. Salah satu widget scrolling yang bisa kita manfaatkan adalah SingleChildScrollView. Widget ini membutuhkan satu child yang nantinya bisa di-scroll pada layar. Pindahkan widget Column ke dalam SingleChildScrollView supaya nantinya bisa di-scroll.
   1. class DetailScreen extends StatelessWidget {
   2. @override
   3. Widget build(BuildContext context) {
   4. return Scaffold(
   5. body: SafeArea(
   6. child: **SingleChildScrollView**(
   7. child: Column(
   8. crossAxisAlignment: CrossAxisAlignment.stretch,
   9. children: <Widget>[
   10. Image.asset('images/farm-house.jpg'),
   11. Container(...),
   12. Container(...),
   13. Container(...),
   14. Image.network(
   15. 'https://media-cdn.tripadvisor.com/media/photo-s/0d/7c/59/70/farmhouse-lembang.jpg'),
   16. ],
   17. ),
   18. ),
   19. ),
   20. );
   21. }
   22. }

Jalankan hot reload. Seharusnya masalah overflow sudah teratasi dengan adanya scrolling.

1. Selanjutnya kita akan menambahkan beberapa gambar lagi yang disusun secara horizontal. Anda mungkin mengira untuk menggunakan widget Row supaya gambar bisa tersusun secara horizontal. Namun, perlu diingat bahwa kita juga memerlukan fitur scrolling agar tidak terjadi overflow. Oleh karena itu, kita akan menggunakan ListView. Widget ini memungkinkan kita untuk menerapkan scrolling terhadap beberapa item (children).
   1. class DetailScreen extends StatelessWidget {
   2. @override
   3. Widget build(BuildContext context) {
   4. return Scaffold(
   5. body: SafeArea(
   6. child: SingleChildScrollView(
   7. child: Column(
   8. crossAxisAlignment: CrossAxisAlignment.stretch,
   9. children: <Widget>[
   10. Image.asset('images/farm-house.jpg'),
   11. Container(...),
   12. Container(...),
   13. Container(...),
   14. **ListView(**
   15. **children: [**
   16. **Image.network(**
   17. **'https://media-cdn.tripadvisor.com/media/photo-s/0d/7c/59/70/farmhouse-lembang.jpg'),**
   18. **Image.network(**
   19. **'https://media-cdn.tripadvisor.com/media/photo-w/13/f0/22/f6/photo3jpg.jpg'),**
   20. **Image.network(**
   21. **'https://media-cdn.tripadvisor.com/media/photo-m/1280/16/a9/33/43/liburan-di-farmhouse.jpg'),**
   22. **],**
   23. **),**
   24. ],
   25. ),
   26. ),
   27. ),
   28. );
   29. }
   30. }

Jika Anda menjalankan aplikasi atau melakukan hot reload, aplikasi Anda akan menjadi blank dan muncul pesan eror pada log. Kenapa ya? ListView diletakkan di dalam Column, di mana keduanya sama-sama memiliki atribut height yang memakan space di sepanjang layar. Sebagai solusi kita perlu memberikan ukuran tinggi yang statis terhadap ListView. Namun ListView tidak memiliki parameter height, lantas bagaimana nih? Caranya, gunakan widget lain yang memiliki parameter height. Anda dapat membungkus widget ListView ke dalam Container atau pun SizedBox. Ukuran tinggi ini nantinya juga digunakan sebagai tinggi Image yang tampil.

* 1. class DetailScreen extends StatelessWidget {
  2. @override
  3. Widget build(BuildContext context) {
  4. return Scaffold(
  5. body: SafeArea(
  6. child: SingleChildScrollView(
  7. child: Column(
  8. crossAxisAlignment: CrossAxisAlignment.stretch,
  9. children: <Widget>[
  10. Image.asset('images/farm-house.jpg'),
  11. Container(...),
  12. Container(...),
  13. Container(...),
  14. **Container(**
  15. **height: 150,**
  16. child: ListView(
  17. children: [
  18. Image.network(
  19. 'https://media-cdn.tripadvisor.com/media/photo-s/0d/7c/59/70/farmhouse-lembang.jpg'),
  20. Image.network(
  21. 'https://media-cdn.tripadvisor.com/media/photo-w/13/f0/22/f6/photo3jpg.jpg'),
  22. Image.network(
  23. 'https://media-cdn.tripadvisor.com/media/photo-m/1280/16/a9/33/43/liburan-di-farmhouse.jpg'),
  24. ],
  25. ),
  26. ),
  27. ],
  28. ),
  29. ),
  30. ),
  31. );
  32. }
  33. }

1. Secara default arah scroll dari ListView adalah vertikal. Untuk mengubahnya menjadi horizontal kita cukup menambahkan parameter scrollDirection bernilai Axis.horizontal.
   1. class DetailScreen extends StatelessWidget {
   2. @override
   3. Widget build(BuildContext context) {
   4. return Scaffold(
   5. body: SafeArea(
   6. child: SingleChildScrollView(
   7. child: Column(
   8. crossAxisAlignment: CrossAxisAlignment.stretch,
   9. children: <Widget>[
   10. Image.asset('images/farm-house.jpg'),
   11. Container(...),
   12. Container(...),
   13. Container(...),
   14. Container(
   15. height: 150,
   16. child: ListView(
   17. **scrollDirection: Axis.horizontal,**
   18. children: [
   19. Image.network(
   20. 'https://media-cdn.tripadvisor.com/media/photo-s/0d/7c/59/70/farmhouse-lembang.jpg'),
   21. Image.network(
   22. 'https://media-cdn.tripadvisor.com/media/photo-w/13/f0/22/f6/photo3jpg.jpg'),
   23. Image.network(
   24. 'https://media-cdn.tripadvisor.com/media/photo-m/1280/16/a9/33/43/liburan-di-farmhouse.jpg'),
   25. ],
   26. ),
   27. ),
   28. ],
   29. ),
   30. ),
   31. ),
   32. );
   33. }
   34. }
2. Selanjutnya, kita akan sedikit merapikan tampilan gambar supaya terlihat lebih rapi dan menarik. Tambahkan Padding pada masing-masing Image supaya antar gambar tidak terlalu rapat.
   1. Container(
   2. height: 150,
   3. child: ListView(
   4. scrollDirection: Axis.horizontal,
   5. children: <Widget>[
   6. **Padding(**
   7. **padding: const EdgeInsets.all(4.0),**
   8. **child: Image.network(         'https://media-cdn.tripadvisor.com/media/photo-s/0d/7c/59/70/farmhouse-lembang.jpg'),**
   9. **),**
   10. **Padding(**
   11. **padding: const EdgeInsets.all(4.0),**
   12. **child: Image.network(               'https://media-cdn.tripadvisor.com/media/photo-w/13/f0/22/f6/photo3jpg.jpg'),**
   13. **),**
   14. **Padding(**
   15. **padding: const EdgeInsets.all(4.0),**
   16. **child: Image.network(                      'https://media-cdn.tripadvisor.com/media/photo-m/1280/16/a9/33/43/liburan-di-farmhouse.jpg'),**
   17. **),**
   18. ],
   19. ),
   20. ),
3. Bagaimana membuat gambar memiliki sudut yang membulat seperti pada contoh? Sekali lagi, dokumentasi adalah sahabat terbaik Anda dalam mengembangkan aplikasi Flutter. Anda dapat memanfaatkan mesin pencari untuk menemukan widget sesuai keinginan. Misalnya, dengan memanfaatkan Google Anda dapat menemukan bahwa ada widget yang memungkinkan gambar memiliki radius, yaitu **ClipRRect**. Masukkan widget Image Anda sebagai **child** dari **ClipRRect** dan berikan **borderRadius**, maka Anda akan mendapatkan Image dengan sudut yang tak bersiku.
4. Terakhir, kita akan menggunakan custom Font. Anda bebas menggunakan font kesukaan Anda. Pada contoh ini akan menggunakan font [Staatliches](https://fonts.google.com/specimen/Staatliches) dan [Oxygen](https://fonts.google.com/specimen/Oxygen). Tambahkan font yang akan digunakan ke dalam project dan daftarkan pada **pubscpec.yaml**.
   1. fonts:
   2. - family: Staatliches
   3. fonts:
   4. - asset: fonts/Staatliches-Regular.ttf
   5. - family: Oxygen
   6. fonts:
   7. - asset: fonts/Oxygen-Regular.ttf
5. Tambahkan parameter **fontFamily** pada widget **TextStyle** untuk menerapkan style pada Text.
   1. Container(
   2. margin: EdgeInsets.only(top: 16.0),
   3. child: Text(
   4. 'Farm House Lembang',
   5. textAlign: TextAlign.center,
   6. style: TextStyle(
   7. fontSize: 30.0,
   8. **fontFamily: 'Staatliches',**
   9. ),
   10. ),
   11. ),
6. Jika Anda memiliki beberapa teks dengan style yang sama, Anda dapat menggunakan variabel untuk menyimpan **TextStyle**dan meringkas kode.
   1. var informationTextStyle = TextStyle(fontFamily: 'Oxygen');

Gunakan variabel tersebut pada masing-masing widgetyang membutuhkan.

* 1. children: <Widget>[
  2. Column(
  3. children: <Widget>[
  4. Icon(Icons.calendar\_today),
  5. SizedBox(height: 8.0),
  6. Text(
  7. 'Open Everyday',
  8. **style: informationTextStyle,**
  9. ),
  10. ],
  11. ),
  12. Column(
  13. children: <Widget>[
  14. Icon(Icons.access\_time),
  15. SizedBox(height: 8.0),
  16. Text(
  17. '09:00 - 20:00',
  18. **style: informationTextStyle,**
  19. ),
  20. ],
  21. ),
  22. Column(
  23. children: <Widget>[
  24. Icon(Icons.monetization\_on),
  25. SizedBox(height: 8.0),
  26. Text(
  27. 'Rp 25.000',
  28. **style: informationTextStyle,**
  29. ),
  30. ],
  31. ),
  32. ],

1. Jalankan aplikasi untuk melihat hasil akhir dari codelab ini.  
   Anda dapat menghapus widget SafeArea jika dirasa tampilan tanpa SafeArea jadi lebih baik.
2. Seluruh kodenya adalah seperti berikut:
   1. import 'package:flutter/material.dart';
   3. var informationTextStyle = TextStyle(fontFamily: 'Oxygen');
   5. class DetailScreen extends StatelessWidget {
   6. @override
   7. Widget build(BuildContext context) {
   8. return Scaffold(
   9. body: SingleChildScrollView(
   10. child: Column(
   11. crossAxisAlignment: CrossAxisAlignment.stretch,
   12. children: <Widget>[
   13. Image.asset('images/farm-house.jpg'),
   14. Container(
   15. margin: EdgeInsets.only(top: 16.0),
   16. child: Text(
   17. 'Farm House Lembang',
   18. textAlign: TextAlign.center,
   19. style: TextStyle(
   20. fontSize: 30.0,
   21. fontFamily: 'Staatliches',
   22. ),
   23. ),
   24. ),
   25. Container(
   26. margin: EdgeInsets.symmetric(vertical: 16.0),
   27. child: Row(
   28. mainAxisAlignment: MainAxisAlignment.spaceEvenly,
   29. children: <Widget>[
   30. Column(
   31. children: <Widget>[
   32. Icon(Icons.calendar\_today),
   33. SizedBox(height: 8.0),
   34. Text(
   35. 'Open Everyday',
   36. style: informationTextStyle,
   37. ),
   38. ],
   39. ),
   40. Column(
   41. children: <Widget>[
   42. Icon(Icons.access\_time),
   43. SizedBox(height: 8.0),
   44. Text(
   45. '09:00 - 20:00',
   46. style: informationTextStyle,
   47. )
   48. ],
   49. ),
   50. Column(
   51. children: <Widget>[
   52. Icon(Icons.monetization\_on),
   53. SizedBox(height: 8.0),
   54. Text(
   55. 'Rp 25.000',
   56. style: informationTextStyle,
   57. ),
   58. ],
   59. )
   60. ],
   61. ),
   62. ),
   63. Container(
   64. padding: EdgeInsets.all(16.0),
   65. child: Text(
   66. '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.',
   67. textAlign: TextAlign.center,
   68. style: TextStyle(
   69. fontSize: 16.0,
   70. fontFamily: 'Oxygen',
   71. ),
   72. ),
   73. ),
   74. Container(
   75. height: 150,
   76. child: ListView(
   77. scrollDirection: Axis.horizontal,
   78. children: <Widget>[
   79. Padding(
   80. padding: const EdgeInsets.all(4.0),
   81. child: ClipRRect(
   82. borderRadius: BorderRadius.circular(10),
   83. child: Image.network(
   84. 'https://media-cdn.tripadvisor.com/media/photo-s/0d/7c/59/70/farmhouse-lembang.jpg'),
   85. ),
   86. ),
   87. Padding(
   88. padding: const EdgeInsets.all(4.0),
   89. child: ClipRRect(
   90. borderRadius: BorderRadius.circular(10),
   91. child: Image.network(
   92. 'https://media-cdn.tripadvisor.com/media/photo-w/13/f0/22/f6/photo3jpg.jpg'),
   93. ),
   94. ),
   95. Padding(
   96. padding: const EdgeInsets.all(4.0),
   97. child: ClipRRect(
   98. borderRadius: BorderRadius.circular(10),
   99. child: Image.network(
   100. 'https://media-cdn.tripadvisor.com/media/photo-m/1280/16/a9/33/43/liburan-di-farmhouse.jpg'),
   101. ),
   102. ),
   103. ],
   104. ),
   105. ),
   106. ],
   107. ),
   108. ),
   109. );
   110. }
   111. }

Anda juga dapat mengunduh seluruh kodenya pada tautan berikut: <https://github.com/dicodingacademy/a159-flutter-pemula-labs/tree/codelab2-final>.

## Codelab 3: Menampilkan Daftar Tempat Wisata

Sekarang kita telah sampai pada codelab ketiga. Di akhir codelab ini kita akan menyelesaikan project aplikasi Wisata Bandung. Hasil akhir aplikasi akan seperti berikut:

Mari kita mulai. Buka kembali project codelab Anda sebelumnya.

1. Pertama kali yang kita lakukan adalah membuat halaman baru untuk menampilkan daftar tempat wisata. Buat berkas baru **main\_screen.dart** lalu buat widget untuk halaman **MainScreen**.
   1. import 'package:flutter/material.dart';
   3. class MainScreen extends StatelessWidget {
   4. @override
   5. Widget build(BuildContext context) {
   6. return Scaffold();
   7. }
   8. }
2. Jangan lupa untuk mengganti halaman utama yang ditampilkan pada berkas **main.dart.**
   1. void main() => runApp(MyApp());
   3. class MyApp extends StatelessWidget {
   4. @override
   5. Widget build(BuildContext context) {
   6. return MaterialApp(
   7. title: 'Wisata Bandung',
   8. theme: ThemeData(),
   9. home: MainScreen(),
   10. );
   11. }
   12. }
3. Pada **MainScreen** tambahkan **AppBar**untuk judul halaman.
   1. class MainScreen extends StatelessWidget {
   2. @override
   3. Widget build(BuildContext context) {
   4. return Scaffold(
   5. appBar: AppBar(
   6. title: Text('Wisata Bandung'),
   7. ),
   8. );
   9. }
   10. }
4. Sebagai body dari Scaffold kita akan menggunakan widget Card. Widget ini adalah widget material design yang menghasilkan tampilan seperti kartu dengan ujung yang membulat dan bayangan di belakang. Kemudian susun Row dan Column seperti contoh untuk menyusun child dari Card. Kodenya akan seperti berikut:
   1. class MainScreen extends StatelessWidget {
   2. @override
   3. Widget build(BuildContext context) {
   4. return Scaffold(
   5. appBar: AppBar(
   6. title: Text('Wisata Bandung'),
   7. ),
   8. body: Card(
   9. child: Row(
   10. crossAxisAlignment: CrossAxisAlignment.start,
   11. children: <Widget>[
   12. Image.asset('images/farm-house.jpg'),
   13. Padding(
   14. padding: const EdgeInsets.all(8.0),
   15. child: Column(
   16. crossAxisAlignment: CrossAxisAlignment.start,
   17. mainAxisSize: MainAxisSize.min,
   18. children: <Widget>[
   19. Text(
   20. 'Farm House Lembang',
   21. style: TextStyle(fontSize: 16.0),
   22. ),
   23. SizedBox(
   24. height: 10,
   25. ),
   26. Text('Lembang'),
   27. ],
   28. ),
   29. )
   30. ],
   31. ),
   32. ),
   33. );
   34. }
   35. }
5. Jalankan aplikasinya. Aplikasi akan mengalami overflowkarena aset gambar yang terlalu besar. Kita bisa saja mengatur tinggi gambar secara manual, namun kali ini kita akan memanfaatkan widget Expanded agar tampilan juga dapat menyesuaikan di perangkat yang lebih besar atau kecil. Bungkus masing-masing item dari widget row ke dalam Expanded. Berikan parameter flex yang menurut Anda cocok.
   1. class MainScreen extends StatelessWidget {
   2. @override
   3. Widget build(BuildContext context) {
   4. return Scaffold(
   5. appBar: AppBar(
   6. title: Text('Wisata Bandung'),
   7. ),
   8. body: Card(
   9. child: Row(
   10. crossAxisAlignment: CrossAxisAlignment.start,
   11. children: <Widget>[
   12. **Expanded(**
   13. **flex: 1,**
   14. **child: Image.asset('images/farm-house.jpg'),**
   15. **),**
   16. **Expanded(**
   17. **flex: 2,**
   18. **child: Padding(**
   19. **padding: const EdgeInsets.all(8.0),**
   20. **child: Column(**
   21. **crossAxisAlignment: CrossAxisAlignment.start,**
   22. **mainAxisSize: MainAxisSize.min,**
   23. **children: <Widget>[**
   24. **Text(**
   25. **'Farm House Lembang',**
   26. **style: TextStyle(fontSize: 16.0),**
   27. **),**
   28. **SizedBox(**
   29. **height: 10,**
   30. **),**
   31. **Text('Lembang'),**
   32. **],**
   33. **),**
   34. **),**
   35. **),**
   36. ],
   37. ),
   38. ),
   39. );
   40. }
   41. }
6. Item pertama Anda sudah siap. Selanjutnya kita akan membuat kartu ini bisa diklik untuk berpindah ke halaman detail. Kita bisa menggunakan widget InkWell yang menyediakan parameter onTap. Pindahkan widget Card Anda menjadi child dari InkWell.
   1. class MainScreen extends StatelessWidget {
   2. @override
   3. Widget build(BuildContext context) {
   4. return Scaffold(
   5. appBar: AppBar(
   6. title: Text('Wisata Bandung'),
   7. ),
   8. **body: InkWell(**
   9. **onTap: () {},**
   10. **child: Card(...),**
   11. **),**
   12. );
   13. }
   14. }
7. Parameter onTapmenerima argumen berupa sebuah fungsi lambda. Di sini kita akan menambahkan Navigator untuk berpindah ke halaman detail.
   1. onTap: () {
   2. Navigator.push(context, MaterialPageRoute(builder: (context) {
   3. return DetailScreen();
   4. }));
   5. },

Jalankan aplikasi. Seharusnya sampai langkah ini aplikasi Anda sudah dapat berpindah halaman ketika item diklik.

1. Selanjutnya kita akan menampilkan beberapa item ke MainScreen. Di kelas ini kita masih menggunakan data statis dan lokal yang disimpan pada objek List. Sebelumnya, buatlah kelas sebagai blueprint untuk menyimpan objek tempat wisata kita. Buat folder baru di dalam folder **lib** dengan cara klik kanan folder **lib -> New -> Package,**lalu berikan nama **model**. Di dalam folder model buat berkas dart bernama **tourism\_place.dart**.
2. Di dalam **tourism\_place.dart** buat data class yang akan menjadi blueprint objek tempat wisata.
   1. class TourismPlace {
   2. String name;
   3. String location;
   4. String description;
   5. String openDays;
   6. String openTime;
   7. String ticketPrice;
   8. String imageAsset;
   9. List<String> imageUrls;
   11. TourismPlace({
   12. required this.name,
   13. required this.location,
   14. required this.description,
   15. required this.openDays,
   16. required this.openTime,
   17. required this.ticketPrice,
   18. required this.imageAsset,
   19. required this.imageUrls,
   20. });
   21. }
3. Siapkan data statis yang ingin ditampilkan Anda dapat menyalin kode berikut dan taruh di berkas **tourism\_place.dart**paling bawah.
   1. var tourismPlaceList = [
   2. TourismPlace(
   3. name: 'Farm House Lembang',
   4. location: 'Lembang',
   5. description:
   6. '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.',
   7. openDays: 'Open Everyday',
   8. openTime: '09:00 - 20:00',
   9. ticketPrice: 'Rp 25000',
   10. imageAsset: 'images/farm-house.jpg',
   11. imageUrls: [
   12. 'https://media-cdn.tripadvisor.com/media/photo-s/0d/7c/59/70/farmhouse-lembang.jpg',
   13. 'https://media-cdn.tripadvisor.com/media/photo-w/13/f0/22/f6/photo3jpg.jpg',
   14. 'https://media-cdn.tripadvisor.com/media/photo-m/1280/16/a9/33/43/liburan-di-farmhouse.jpg'
   15. ],
   16. ),
   17. TourismPlace(
   18. name: 'Observatorium Bosscha',
   19. location: 'Lembang',
   20. description:
   21. 'Memiliki beberapa teleskop, antara lain, Refraktor Ganda Zeiss, Schmidt Bimasakti, Refraktor Bamberg, Cassegrain GOTO, dan Teleskop Surya. Refraktor Ganda Zeiss adalah jenis teleskop terbesar untuk meneropong bintang. Benda ini diletakkan pada atap kubah sehingga saat teropong digunakan, atap tersebut harus dibuka. Observatorium Bosscha boleh dikunjungi oleh siapa pun, tanpa tiket. Namun, bagi yang ingin menggunakan teleskop Zeiss, wajib mendaftarkan diri. Untuk instansi atau lembaga pendidikan, diberikan jadwal hari Selasa sampai Jumat. Sementara itu, kunjungan individu dibuka setiap hari Sabtu.',
   22. openDays: 'Open Tuesday - Saturday',
   23. openTime: '09:00 - 14:30',
   24. ticketPrice: 'Rp 20000',
   25. imageAsset: 'images/bosscha.jpg',
   26. imageUrls: [
   27. 'https://media-cdn.tripadvisor.com/media/photo-o/12/6b/63/0b/bosscha-observatory.jpg',
   28. 'https://media-cdn.tripadvisor.com/media/photo-p/0d/6a/88/9b/photo3jpg.jpg',
   29. 'https://media-cdn.tripadvisor.com/media/photo-o/11/3f/04/39/p-20171111-110220-largejpg.jpg',
   30. ],
   31. ),
   32. TourismPlace(
   33. name: 'Jalan Asia Afrika',
   34. location: 'Kota Bandung',
   35. description:
   36. 'Jalan Asia Afrika di Bandung memiliki kaitan yang sangat erat dengan pendirian kota Kembang ini. Karena pada saat itu, Gubernur Jenderal Herman Willem Deaendels dari Belanda menancapkan tongkatnya saat memerintahkan pendirian kota ini, yang kemudian diabadikan menjadi tugu Bandung Nol Kilometer.',
   37. openDays: 'Open Everyday',
   38. openTime: '24 hours',
   39. ticketPrice: 'Free',
   40. imageAsset: 'images/jalan-asia-afrika.jpg',
   41. imageUrls: [
   42. 'https://media-cdn.tripadvisor.com/media/photo-o/0d/c2/e7/e6/quotes-kota-bandung.jpg',
   43. 'https://media-cdn.tripadvisor.com/media/photo-w/17/f4/44/01/jalan-asia-afrika.jpg',
   44. 'https://media-cdn.tripadvisor.com/media/photo-s/0a/ef/36/e2/jalan-asia-afrika.jpg',
   45. ],
   46. ),
   47. TourismPlace(
   48. name: 'Stone Garden',
   49. location: 'Padalarang',
   50. description:
   51. 'Stone Garden atau Taman Batu di Padalarang – Bandung ini adalah nama secara harafiah untuk apa yang akan kita lihat jika berada di sana. Hamparan batu yang artistik membuat kita merasa tidak sedang berada di Bandung, apalagi di Padalarang. Hamparan batu yang dimaksud bukan terhampar begitu saja di atas tanah luas yang menjadi permukaannya. Batu-batu besar yang ukuran pastinya bervariasi tersusun seperti memiliki suatu formasi matematis.',
   52. openDays: 'Open Everyday',
   53. openTime: '06:00 - 17:00',
   54. ticketPrice: 'Rp 3000',
   55. imageAsset: 'images/stone-garden.jpg',
   56. imageUrls: [
   57. 'https://media-cdn.tripadvisor.com/media/photo-o/15/01/d7/4b/p-20180510-153310-01.jpg',
   58. 'https://media-cdn.tripadvisor.com/media/photo-w/15/68/00/32/stone-garden-citatah.jpg',
   59. 'https://media-cdn.tripadvisor.com/media/photo-o/0d/a2/cb/05/stone-garden-citatah.jpg',
   60. ],
   61. ),
   62. TourismPlace(
   63. name: 'Taman Film Pasopati',
   64. location: 'Kota Bandung',
   65. description:
   66. 'Menjadi salah satu tempat wisata di Bandung yang favorit, tentu Taman Film ini memiliki fasilitas cukup memadai. Pemberian fasilitas ini memiliki harapan para pengunjung akan merasa nyaman dan tak segan2 untuk kembali berkunjung terus menerus kesini. Beberapa fasilitas taman yang bisa kamu nikmati diantaranya seperti layar videotron besar berukuran 4×8 untuk memutar berbagai macam pilihan film seperti Film Indonesia, Bollywood, Korea, ataupun Indie Bandung.',
   67. openDays: 'Open Everyday',
   68. openTime: '24 hours',
   69. ticketPrice: 'Free',
   70. imageAsset: 'images/taman-film.jpg',
   71. imageUrls: [
   72. 'https://media-cdn.tripadvisor.com/media/photo-o/08/8b/87/50/bandung-movie-park.jpg',
   73. 'https://media-cdn.tripadvisor.com/media/photo-o/17/67/d5/53/img-20190505-114509-largejpg.jpg',
   74. 'https://media-cdn.tripadvisor.com/media/photo-w/09/73/33/05/taman-film-pasopati.jpg',
   75. ],
   76. ),
   77. TourismPlace(
   78. name: 'Museum Geologi',
   79. location: 'Kota Bandung',
   80. description:
   81. 'Museum Geologi didirikan pada tanggal 16 Mei 1929. Museum ini telah direnovasi dengan dana bantuan dari JICA (Japan International Cooperation Agency). Setelah mengalami renovasi, Museum Geologi dibuka kembali dan diresmikan oleh Wakil Presiden RI, Megawati Soekarnoputri pada tanggal 23 Agustus 2000. Sebagai salah satu monumen bersejarah, museum berada di bawah perlindungan pemerintah dan merupakan peninggalan nasional. Dalam Museum ini, tersimpan dan dikelola materi-materi geologi yang berlimpah, seperti fosil, batuan, mineral. Kesemuanya itu dikumpulkan selama kerja lapangan di Indonesia sejak 1850.',
   82. openDays: 'Open Saturday - Thursday',
   83. openTime: '09:00 - 15:30',
   84. ticketPrice: 'Rp 3000',
   85. imageAsset: 'images/museum-geologi.jpg',
   86. imageUrls: [
   87. 'https://media-cdn.tripadvisor.com/media/photo-w/19/1c/8e/f7/geology-museum.jpg',
   88. 'https://media-cdn.tripadvisor.com/media/photo-o/11/a7/35/b7/geology-museum.jpg',
   89. 'https://media-cdn.tripadvisor.com/media/photo-s/1a/55/e0/dc/geology-museum.jpg',
   90. ],
   91. ),
   92. TourismPlace(
   93. name: 'Floating Market',
   94. location: 'Lembang',
   95. description:
   96. 'Tempat wisata ini sepertinya memang ditujukan untuk wisata keluarga di Bandung. Di sini kita bisa menikmati suasana kawasan yang tertata rapi dan alami. Pada awalnya, floating market Lembang tidak begitu luas. Tapi sekarang sudah ekspansi dan memiliki banyak objek menarik baru. Nama floating market ini sepertinya merujuk pada stand tempat jualan makanan yang berada dalam perahu.',
   97. openDays: 'Open Everyday',
   98. openTime: '09:00 - 17:00',
   99. ticketPrice: 'Rp 20000',
   100. imageAsset: 'images/floating-market.png',
   101. imageUrls: [
   102. 'https://media-cdn.tripadvisor.com/media/photo-o/17/f9/ff/f8/floating-market-bandung.jpg',
   103. 'https://media-cdn.tripadvisor.com/media/photo-p/1a/86/d3/cd/20200103-125059-largejpg.jpg',
   104. 'https://media-cdn.tripadvisor.com/media/photo-p/19/ce/b4/9b/img20181224120857-largejpg.jpg',
   105. ],
   106. ),
   107. TourismPlace(
   108. name: 'Kawah Putih',
   109. location: 'Ciwidey',
   110. description:
   111. 'Kawah Putih adalah tempat wisata di Bandung yang paling terkenal. Berlokasi di Ciwidey, Jawa Barat, kurang lebih sekitar 50 KM arah selatan kota Bandung, Kawah Putih adalah sebuah danau yang terbentuk akibat dari letusan Gunung Patuha. Sesuai dengan namanya, tanah yang ada di kawasan ini berwarna putih akibat dari pencampuran unsur belerang.',
   112. openDays: 'Open Everyday',
   113. openTime: '07:00 - 17:00',
   114. ticketPrice: 'Rp 15000',
   115. imageAsset: 'images/kawah-putih.jpg',
   116. imageUrls: [
   117. 'https://media-cdn.tripadvisor.com/media/photo-o/0b/6e/7c/ce/rocks-sticking-out-of.jpg',
   118. 'https://media-cdn.tripadvisor.com/media/photo-p/0b/35/30/14/white-crater.jpg',
   119. 'https://media-cdn.tripadvisor.com/media/photo-o/0a/8b/9a/79/2945-t00572-www-initempatwisat.jpg',
   120. ],
   121. ),
   122. TourismPlace(
   123. name: 'Ranca Upas',
   124. location: 'Ciwidey',
   125. description:
   126. 'Ranca Upas Ciwidey adalah kawasan bumi perkemahan di bawah pengelolaan perhutani. Tempat ini berada di kawasan wisata Bandung Selatan, satu lokasi dengan kawah putih, kolam Cimanggu dan situ Patenggang. Banyak hal yang bisa dilakukan di kawasan wisata ini, seperti berkemah, berinteraksi dengan rusa, sampai bermain di water park dan mandi air panas.',
   127. openDays: 'Open Everyday',
   128. openTime: '24 hours',
   129. ticketPrice: 'Rp 20000',
   130. imageAsset: 'images/ranca-upas.jpg',
   131. imageUrls: [
   132. 'https://media-cdn.tripadvisor.com/media/photo-o/1a/e0/7f/9c/kampung-cai-ranca-upas.jpg',
   133. 'https://media-cdn.tripadvisor.com/media/photo-w/13/ee/2f/87/ranca-upas.jpg',
   134. 'https://media-cdn.tripadvisor.com/media/photo-w/13/ee/27/0a/ranca-upas.jpg',
   135. ],
   136. ),
   137. ];

Jangan lupa untuk menambahkan import berkas **tourism\_place.dart** pada file **main\_screen.dart**.

* 1. import 'package:wisatabandung/model/tourism\_place.dart';

Untuk berkas aset yang digunakan dapat Anda unduh pada tautan [berikut](https://github.com/dicodingacademy/assets/raw/main/flutter_pemula_academy/assets_wisata.zip).

1. Sesuai yang telah kita pelajari pada materi ListView, kita akan menampilkan variabel tourismPlaceList di atas menjadi item card yang dapat diklik. Tambahkan widget ListView sebagai body dari Scaffold**.** Pindahkan widget FlatButton dan seluruh konten di dalamnya sebagai widget dari setiap item di dalam tourismPlaceList.
   1. class MainScreen extends StatelessWidget {
   2. @override
   3. Widget build(BuildContext context) {
   4. return Scaffold(
   5. appBar: AppBar(
   6. title: Text('Wisata Bandung'),
   7. ),
   8. body: **ListView.builder(**
   9. **itemBuilder: (context, index) {**
   10. **final TourismPlace place = tourismPlaceList[index];**
   11. return InkWell(
   12. onTap: () {
   13. Navigator.push(context, MaterialPageRoute(builder: (context) {
   14. return DetailScreen();
   15. }));
   16. },
   17. child: Card(
   18. child: Row(
   19. crossAxisAlignment: CrossAxisAlignment.start,
   20. children: <Widget>[
   21. Expanded(
   22. flex: 1,
   23. **child: Image.asset(place.imageAsset),**
   24. ),
   25. Expanded(
   26. flex: 2,
   27. child: Padding(
   28. padding: const EdgeInsets.all(8.0),
   29. child: Column(
   30. crossAxisAlignment: CrossAxisAlignment.start,
   31. children: <Widget>[
   32. **Text(**
   33. **place.name,**
   34. **style: TextStyle(fontSize: 16.0),**
   35. **),**
   36. SizedBox(
   37. height: 10,
   38. ),
   39. **Text(place.location),**
   40. ],
   41. ),
   42. ),
   43. )
   44. ],
   45. ),
   46. ),
   47. );
   48. },
   49. **itemCount: tourismPlaceList.length,**
   50. ),
   51. );
   52. }
   53. }

Jangan lupa untuk mengganti tampilan item secara dinamis sesuai data dari objek TourismPlace. Jalankan aplikasi untuk melihat hasil perubahan.

1. Agar halaman detail bisa menampilkan informasi sesuai tempat wisata yang dipilih, kita perlu mengirimkan data TourismPlace melalui constructor. Buka berkas **detail\_screen.dart** lalu tambahkan variabel serta constructor-nya.
   1. class DetailScreen extends StatelessWidget {
   2. **final TourismPlace place;**
   3. **DetailScreen({required this.place});**
   5. @override
   6. Widget build(BuildContext context) {...}
   7. }

Tambahkan keyword required agar parameter place wajib disertakan ketika membuat objek **DetailScreen**. Sesuaikan juga informasi yang ditampilkan dengan property yang didapat dari constructor.

* 1. class DetailScreen extends StatelessWidget {
  2. final TourismPlace place;
  4. DetailScreen({required this.place});
  6. @override
  7. Widget build(BuildContext context) {
  8. return Scaffold(
  9. backgroundColor: Colors.black,
  10. body: SingleChildScrollView(
  11. child: Column(
  12. crossAxisAlignment: CrossAxisAlignment.stretch,
  13. children: <Widget>[
  14. **Image.asset(place.imageAsset),**
  15. Container(
  16. margin: EdgeInsets.only(top: 16.0),
  17. child: **Text(**
  18. **place.name,**
  19. **textAlign: TextAlign.center,**
  20. **style: TextStyle(**
  21. **fontSize: 30.0,**
  22. **fontFamily: 'Staatliches',**
  23. **),**
  24. **),**
  25. ),
  26. Container(
  27. margin: EdgeInsets.symmetric(vertical: 16.0),
  28. child: Row(
  29. mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  30. children: <Widget>[
  31. Column(
  32. children: <Widget>[
  33. Icon(Icons.calendar\_today),
  34. SizedBox(height: 8.0),
  35. **Text(**
  36. **place.openDays,**
  37. **style: informationTextStyle,**
  38. **),**
  39. ],
  40. ),
  41. Column(
  42. children: <Widget>[
  43. Icon(Icons.access\_time),
  44. SizedBox(height: 8.0),
  45. **Text(**
  46. **place.openTime,**
  47. **style: informationTextStyle,**
  48. **),**
  49. ],
  50. ),
  51. Column(
  52. children: <Widget>[
  53. Icon(Icons.monetization\_on),
  54. SizedBox(height: 8.0),
  55. **Text(**
  56. **place.ticketPrice,**
  57. **style: informationTextStyle,**
  58. **),**
  59. ],
  60. ),
  61. ],
  62. ),
  63. ),
  64. Container(
  65. padding: EdgeInsets.all(16.0),
  66. child: **Text(**
  67. **place.description,**
  68. **textAlign: TextAlign.center,**
  69. **style: TextStyle(**
  70. **fontSize: 16.0,**
  71. **fontFamily: 'Oxygen',**
  72. **),**
  73. ),
  74. ),
  75. Container(
  76. height: 150,
  77. child: ListView(
  78. scrollDirection: Axis.horizontal,
  79. children: **place.imageUrls.map((url) {**
  80. **return Padding(**
  81. **padding: const EdgeInsets.all(4.0),**
  82. **child: ClipRRect(**
  83. **borderRadius: BorderRadius.circular(10),**
  84. **child: Image.network(url),**
  85. **),**
  86. **);**
  87. **}).toList(),**
  88. ),
  89. ),
  90. ],
  91. ),
  92. ),
  93. );
  94. }
  95. }

Jangan lupa untuk menambahkan data **place** pada constructor.

* 1. Navigator.push(context, MaterialPageRoute(builder: (context) {
  2. **return DetailScreen(place: place);**
  3. }));

1. Selanjutnya, kita akan menambahkan tombol navigasi untuk kembali ke halaman daftar tempat wisata. Tombol ini akan kita taruh di atas gambar utama atau gambar dari aset. Kita akan menggunakan widget Stack. Widget ini digunakan untuk menyusun widget seperti Column atau Row, bedanya widget pada Stack disusun secara bertumpuk (stacked). Ubah kode Anda menjadi seperti berikut:
   1. Stack(
   2. children: <Widget>[
   3. Image.asset(place.imageAsset),
   4. IconButton(icon: Icon(Icons.arrow\_back), onPressed: () {})
   5. ],
   6. ),

Tambahkan fungsionalitas agar ketika icon back ini diklik akan kembali ke halaman sebelumnya.

* 1. IconButton(
  2. icon: Icon(Icons.arrow\_back),
  3. onPressed: () {
  4. Navigator.pop(context);
  5. },
  6. ),

1. Jika Anda jalankan aplikasi, ikon ini akan sedikit menabrak notification bar pada perangkat Android. Hal ini akan semakin terlihat apabila Anda menggunakan perangkat yang memiliki notch.  
     
   Lalu bagaimana mengatasinya? Masih ingat dengan SafeArea? Kita akan memanfaatkan widget SafeArea yang akan memberikan paddingsesuai dengan sistem operasi yang digunakan sehingga widget akan berada di area yang aman. Buat widget SafeArea lalu pindahkan IconButton ke dalamnya.
   1. SafeArea(
   2. child: IconButton(
   3. icon: Icon(Icons.arrow\_back),
   4. onPressed: () {
   5. Navigator.pop(context);
   6. },
   7. ),
   8. ),

Lakukan juga beberapa perubahan tampilan supaya ikon navigasi tidak bertabarakan dengan latar belakangnya.

* 1. SafeArea(
  2. child: Padding(
  3. padding: const EdgeInsets.all(8.0),
  4. child: CircleAvatar(
  5. backgroundColor: Colors.grey,
  6. child: IconButton(
  7. icon: Icon(
  8. Icons.arrow\_back,
  9. color: Colors.white,
  10. ),
  11. onPressed: () {
  12. Navigator.pop(context);
  13. },
  14. ),
  15. ),
  16. ),
  17. ),

1. Terakhir, kita akan membuat fitur untuk menambahkan favorit. Fitur favorit ini memang belum lengkap, namun setidaknya cukup memberikan Anda gambaran bagaimana mengubah state aplikasi dan bagaimana widget dapat tampil sesuai dengan state yang ada.  
     
   Buat StatefulWidget pada berkas **detail\_screen.dart**. Widget ini akan kita gunakan untuk menampilkan ikon favorit.
   1. class FavoriteButton extends StatefulWidget {
   2. @override
   3. \_FavoriteButtonState createState() => \_FavoriteButtonState();
   4. }
   6. class \_FavoriteButtonState extends State<FavoriteButton> {
   7. @override
   8. Widget build(BuildContext context) {
   9. return IconButton(
   10. icon: Icon(Icons.favorite\_border),
   11. onPressed: () {},
   12. );
   13. }
   14. }
2. Tambahkan variabel boolean pada class\_FavoriteButtonState. Variabel ini merupakan sebuah state yang dapat berubah dan widget kita akan tampil sesuai state-nya.
   1. bool isFavorite = false;

State isFavorite ini akan berubah ketika ikon favorit diklik. Sehingga tambahkan kode untuk mengubah variabel **isFavorite**. Pastikan Anda memanggil fungsi setState untuk mengubah state.

* 1. onPressed: () {
  2. setState(() {
  3. isFavorite = !isFavorite;
  4. });
  5. },

Ubah ikon yang ditampilkan sesuai dengan kondisi state. Pada kode di bawah ini kita menggunakan ekspresi ternary.

* 1. icon: Icon(
  2. **isFavorite ? Icons.favorite : Icons.favorite\_border,**
  3. color: Colors.red,
  4. ),

1. Tambahkan widget FavoriteButton ini sejajar dengan icon navigasi back.
   1. SafeArea(
   2. child: Padding(
   3. padding: const EdgeInsets.all(8.0),
   4. child: Row(
   5. **mainAxisAlignment: MainAxisAlignment.spaceBetween,**
   6. children: [
   7. CircleAvatar(
   8. backgroundColor: Colors.grey,
   9. child: IconButton(
   10. icon: Icon(
   11. Icons.arrow\_back,
   12. color: Colors.white,
   13. ),
   14. onPressed: () {
   15. Navigator.pop(context);
   16. },
   17. ),
   18. ),
   19. **FavoriteButton(),**
   20. ],
   21. ),
   22. ),
   23. ),

Jalankan aplikasi dan lihat perubahannya. Ketika ikon favorit diklik dan fungsi setState() dipanggil, maka method build akan kembali dijalankan dan widget akan dibuat dan ditampilkan menurut state-nya.

Selamat! Anda telah menyelesaikan seluruh codelab dan projek Wisata Bandung. Seluruh kodenya adalah seperti berikut:

* [**main\_screen.dart**](https://www.dicoding.com/academies/159/tutorials/8621?from=16760#tab1-code1)
* [detail\_screen.dart](https://www.dicoding.com/academies/159/tutorials/8621?from=16760#tab1-code2)
* [tourism\_place.dart](https://www.dicoding.com/academies/159/tutorials/8621?from=16760#tab1-code3)

1. import 'package:flutter/material.dart';
2. import 'package:wisatabandung/detail\_screen.dart';
3. import 'package:wisatabandung/model/tourism\_place.dart';

6. class MainScreen extends StatelessWidget {
7. @override
8. Widget build(BuildContext context) {
9. return Scaffold(
10. appBar: AppBar(
11. title: Text('Wisata Bandung'),
12. ),
13. body: ListView.builder(
14. itemBuilder: (context, index) {
15. final TourismPlace place = tourismPlaceList[index];
16. return InkWell(
17. onTap: () {
18. Navigator.push(context, MaterialPageRoute(builder: (context) {
19. return DetailScreen(place: place);
20. }));
21. },
22. child: Card(
23. child: Row(
24. crossAxisAlignment: CrossAxisAlignment.start,
25. children: <Widget>[
26. Expanded(
27. flex: 1,
28. child: Image.asset(place.imageAsset),
29. ),
30. Expanded(
31. flex: 2,
32. child: Padding(
33. padding: const EdgeInsets.all(8.0),
34. child: Column(
35. crossAxisAlignment: CrossAxisAlignment.start,
36. children: <Widget>[
37. Text(
38. place.name,
39. style: TextStyle(fontSize: 16.0),
40. ),
41. SizedBox(
42. height: 10,
43. ),
44. Text(place.location),
45. ],
46. ),
47. ),
48. )
49. ],
50. ),
51. ),
52. );
53. },
54. itemCount: tourismPlaceList.length,
55. ),
56. );
57. }
58. }

Anda dapat mengunduh seluruh kodenya pada tautan berikut: <https://github.com/dicodingacademy/a159-flutter-pemula-labs/tree/codelab3-final>

**Tambahan**: Flutter dikenal dengan framework-nya yang sangat mudah dalam menghadirkan tampilan yang menarik termasuk menambahkan animasi. Salah satu yang paling mudah adalah Hero Animation. Sebagai tantangan, bisakah Anda menambahkan Hero Animation pada aplikasi Anda? Dicoding sudah pernah mengulasnya dalam blog berikut: <https://www.dicoding.com/blog/menerapkan-animasi-pada-project-flutter/>

*const* DetailScreen({Key? key}) : super(key: key);