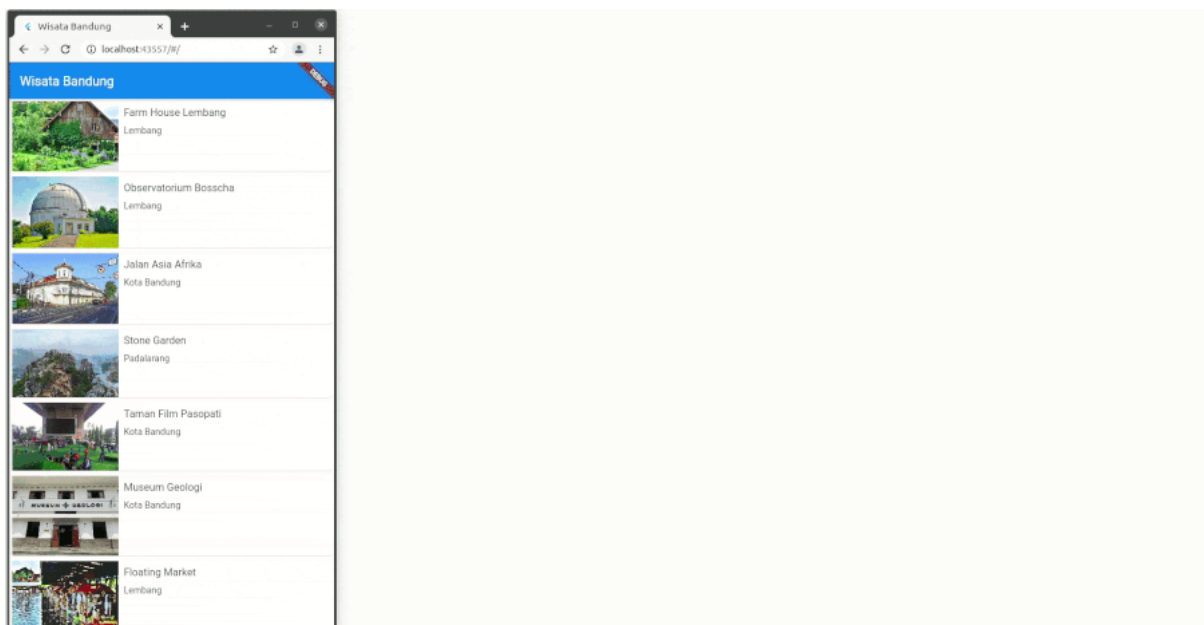


Codelab 4: Pengembangan untuk Web

Sejauh ini kita telah mengembangkan aplikasi Wisata Bandung untuk platform mobile. Dengan Flutter mengumumkan Flutter Web yang sudah masuk versi stable, artinya terdapat target platform tambahan yang perlu kita kerjakan jika ingin menyasar pengguna yang lebih luas. Bukan hanya web, pada codelab kali ini kita juga akan menerapkan layout yang responsif untuk platform mobile untuk ukuran layar yang lebih besar seperti tablet.

Hasil akhir dari codelab ini akan seperti berikut:

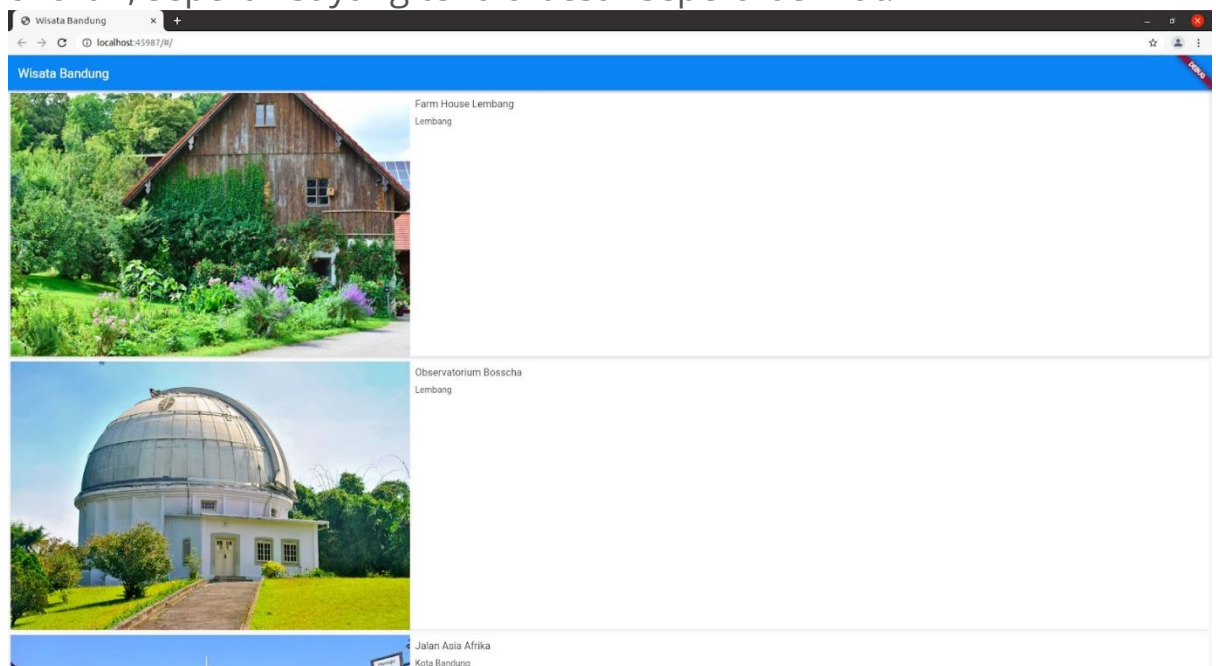


Mari kita mulai!

1. Silakan buka project aplikasi Wisata Bandung Anda.
2. Karena kita akan menambahkan target dukungan untuk web, jalankan perintah berikut pada terminal:
 1. `flutter create .`

Notes: Saat ini Flutter Web belum mendukung hot reload, tetapi hanya hot restart. Jika Anda ingin menggunakan hot reload untuk proses pengembangan yang lebih cepat, kami sarankan menggunakan platform Flutter Desktop yang bisa Anda aktifkan dengan mengikuti langkah Set up pada tautan [berikut](#).

3. Sekarang jalankan aplikasi Anda. Aplikasi Akan berjalan dengan baik di browser, tetapi kita akan menemui masalah dalam aspek ukuran, seperti list yang terlalu besar seperti berikut:



4. Karena kita telah membuat aplikasi mobile lalu ingin membuat versi webnya, maka secara tidak langsung kita telah menerapkan *mobile-first design*. Sekarang kita perlu menentukan pada ukuran berapa *layout mobile* ini sudah tidak sesuai dan perlu diubah tampilannya. Untuk memudahkan, mari kita tampilkan lebar browser atau layar ke dalam teks AppBar.

```
1. AppBar(  
2.   title:  
3.     Text('Wisata Bandung. Size: ${MediaQuery.of(context).size.width}'),  
4. ),
```


5. Cobalah mengubah ukuran window browser dan cari breakpoint atau titik di mana layout sudah tidak sesuai.

Wisata Bandung

localhost:45987/#/


☆

Wisata Bandung. Size: 606




Farm House Lembang

Lembang




Observatorium Bosscha

Lembang




Jalan Asia Afrika

Kota Bandung




Stone Garden

Padalarang




Taman Film Pasopati

Kota Bandung



Museum Geologi

Kota Bandung



Floating Market

Lembang

Sebagai contoh, pada ukuran lebar di atas 600 sudah banyak ruang kosong yang lebih baik ditempati oleh widget lain. Maka, kita akan gunakan ukuran lebar 600 sebagai batas breakpoint ukuran mobile.

6. Pada ukuran lebar di atas 600 kita akan buat tampilan yang berbeda menggunakan grid. Sebelumnya, mari pindahkan tampilan ListView menjadi widget tersendiri.

```
1. class MainScreen extends StatelessWidget {  
2.   @override  
3.   Widget build(BuildContext context) {  
4.     return Scaffold(  
5.       appBar: AppBar(  
6.         title:  
7.           Text('Wisata Bandung. Size: ${MediaQuery.of(context).size.width}'),  
8.       ),  
9.       body: TourismPlaceList(),  
10.    );  
11.  }  
12. }  
13.  
14. class TourismPlaceList extends StatelessWidget {  
15.   @override  
16.   Widget build(BuildContext context) {  
17.     return ListView.builder(...);  
18.   }  
19. }
```

7. Kemudian buat widget baru untuk menampilkan GridView.

```
1. class TourismPlaceGrid extends StatelessWidget {  
2.   @override  
3.   Widget build(BuildContext context) {  
4.     return Padding(  
5.       padding: const EdgeInsets.all(24.0),  
6.       child: GridView.count(  
7.         crossAxisCount: 4,  
8.         children: [],  
9.       ),  
10.    );  
11.  }  
12. }
```

8. Selanjutnya ubah body dari MainScreen untuk menampilkan TourismPlaceList atau TourismPlaceGrid. Kita akan memanfaatkan widget LayoutBuilder untuk mendapatkan ukuran layar.

```
1. class MainScreen extends StatelessWidget {  
2.   @override  
3.   Widget build(BuildContext context) {  
4.     return Scaffold(  
5.       appBar: AppBar(  

```

```

6.         title:
7.             Text('Wisata Bandung. Size: ${MediaQuery.of(context).size.width}'),
8.         ),
9.         body: LayoutBuilder(
10.            builder: (BuildContext context, BoxConstraints constraints) {
11.                if (constraints.maxWidth <= 600) {
12.                    return TourismPlaceList();
13.                } else {
14.                    return TourismPlaceGrid();
15.                }
16.            },
17.        ),
18.    );
19. }
20. }

```

Tampilan aplikasi di atas lebar 600 akan kosong karena kita belum mendefinisikan item untuk ditampilkan.

9. Tampilan item grid yang akan kita buat akan seperti di bawah ini. Cobalah untuk membuat tampilan Card seperti ini dulu sebelum melihat kode pada langkah berikutnya.



10. Berikut ini adalah tampilan untuk Card di atas:

```

1. GridView.count(
2.   crossAxisCount: 4,
3.   children: tourismPlaceList.map((place) {
4.     return InkWell(
5.       onTap: () {
6.         Navigator.push(context, MaterialPageRoute(builder: (context) {
7.           return DetailScreen(place: place);
8.         }));

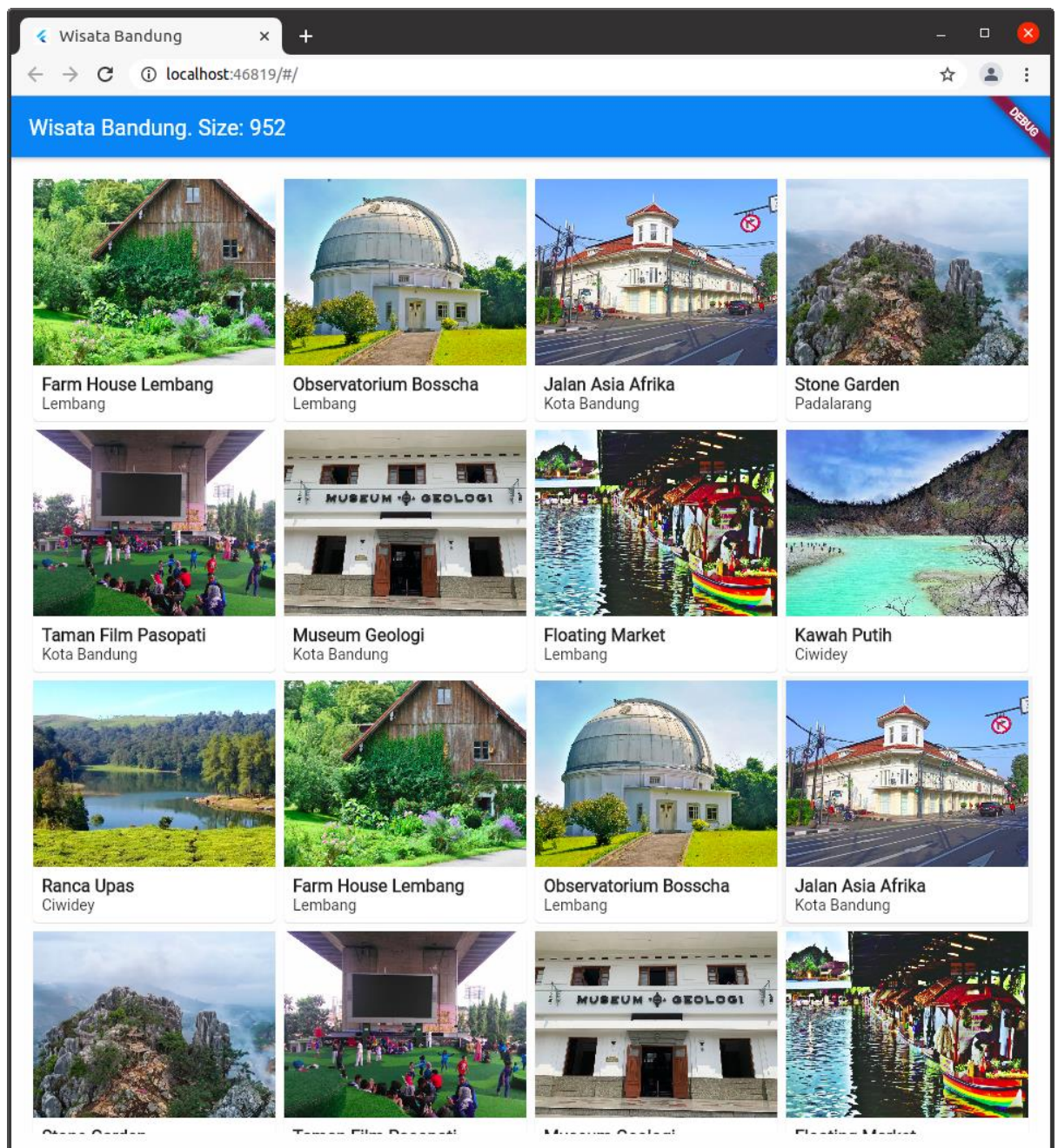
```

```

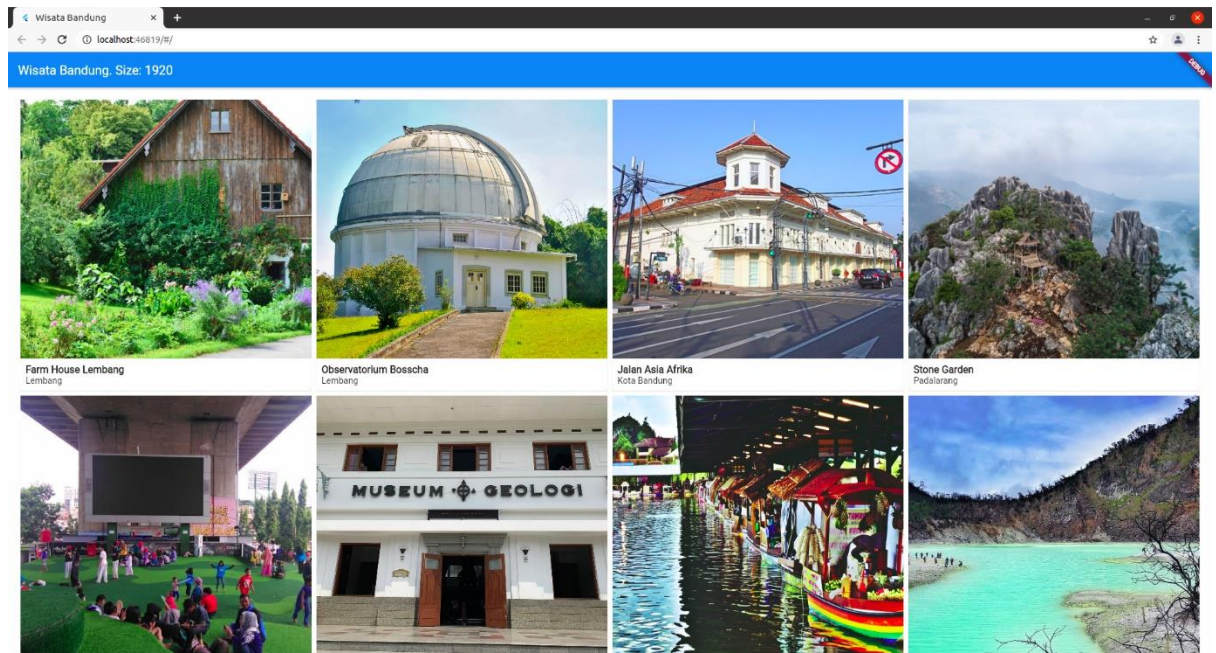
9.      },
10.     child: Card(
11.       child: Column(
12.         crossAxisAlignment: CrossAxisAlignment.stretch,
13.         children: [
14.           Expanded(
15.             child: Image.asset(
16.               place.imageAsset,
17.               fit: BoxFit.cover,
18.             ),
19.           ),
20.           SizedBox(height: 8),
21.           Padding(
22.             padding: const EdgeInsets.only(left: 8.0),
23.             child: Text(
24.               place.name,
25.               style: TextStyle(
26.                 fontSize: 16.0,
27.                 fontWeight: FontWeight.bold,
28.               ),
29.             ),
30.           ),
31.           Padding(
32.             padding: const EdgeInsets.only(left: 8.0, bot
tom: 8.0),
33.             child: Text(
34.               place.location,
35.             ),
36.           ),
37.         ],
38.       ),
39.     ),
40.   );
41.   }).toList(),
42. ),

```

11. Jalankan aplikasi untuk melihat perubahan. Anda dapat melakukan duplikasi data dummy agar tampilan data menjadi semakin banyak.



Namun, untuk ukuran resolusi layar yang lebih besar lagi tampilan grid dengan empat kolom akan terlihat terlalu besar.



Geser kembali ukuran jendela browser untuk menemukan titik di mana tampilan sudah harus berubah. Misalnya, kita ambil ukuran lebar di atas 1200 akan menampilkan 6 kolom.

12. Tambahkan lagi kondisi pada LayoutBuilder. Kita akan menentukan jumlah kolom yang digunakan melalui parameter constructor.

```
1. LayoutBuilder(
2.   builder: (BuildContext context, BoxConstraints constraints)
3.   {
4.     if (constraints.maxWidth <= 600) {
5.       return TourismPlaceList();
6.     } else if (constraints.maxWidth <= 1200) {
7.       return TourismPlaceGrid(gridCount: 4);
8.     } else {
9.       return TourismPlaceGrid(gridCount: 6);
10.    }
11.  },
12. ),
```

13. Tambahkan constructor pada kelas TourismPlaceGrid.

```
1. class TourismPlaceGrid extends StatelessWidget {
2.   final int gridCount;
3.
4.   TourismPlaceGrid({required this.gridCount});
5.
6.   @override
7.   Widget build(BuildContext context) {
8.     return Padding(
9.       padding: const EdgeInsets.all(16.0),
10.      child: GridView.count(
11.        crossAxisCount: gridCount,
12.        children: tourismPlaceList.map((place) {
```

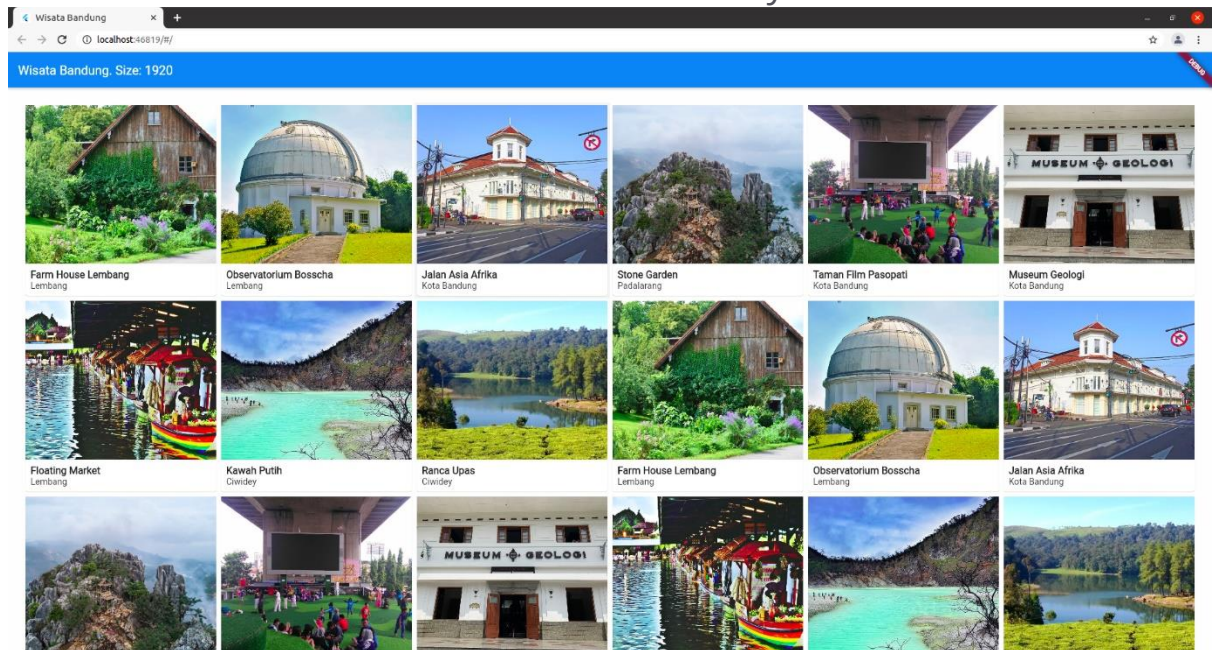


```

13.         return InkWell(...);
14.     }).toList(),
15.     ),
16.     );
17. }
18. }

```

14. Jalankan aplikasi dan lihat perubahan. Tampilan untuk lebar resolusi 1920 sudah lebih baik dari sebelumnya.



15. Selanjutnya mari tambahkan jarak antara item supaya tidak terlalu rapat.

```

1. GridView.count(
2.   crossAxisCount: gridCount,
3.   crossAxisSpacing: 16,
4.   mainAxisSpacing: 16,
5.   children: tourismPlaceList.map((place) {
6.     return InkWell(...);
7.   }).toList(),
8. ),

```

16. Terakhir untuk halaman daftar tempat wisata, karena Flutter web belum secara otomatis memiliki scrollbar untuk browser, maka kita perlu menambahkannya sendiri. Tambahkan widget Scrollbar sebagai widget paling atas dari TourismPlaceGrid.

```

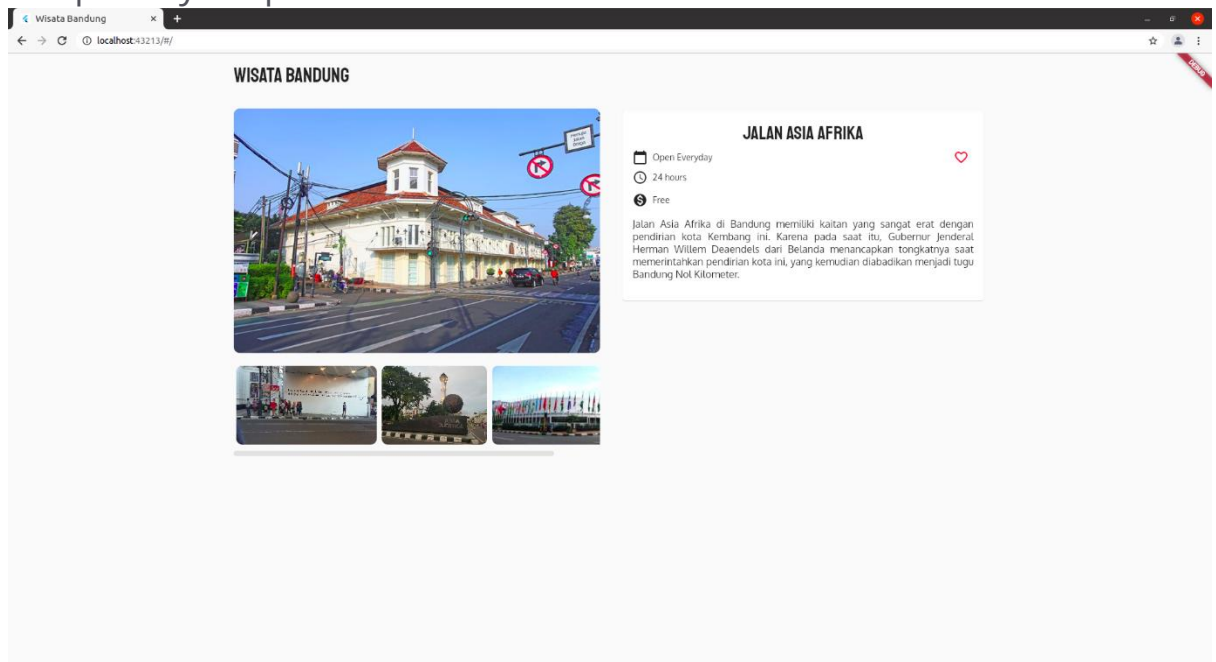
1. Scrollbar(
2.   isAlwaysShown: true,
3.   child: Padding(
4.     padding: const EdgeInsets.all(24.0),
5.     child: GridView.count(...),
6.   ),
7. );

```

17. Jangan lupa untuk menghapus kembali teks ukuran pada AppBar.

```
1. AppBar(  
2.   title: Text('Wisata Bandung'),  
3. ),
```

18. Selanjutnya kita akan mulai memodifikasi tampilan halaman detail. Seperti yang Anda lihat ketika menjalankan aplikasi, halaman detail tampak kurang estetik untuk ukuran resolusi yang lebar. Untuk ukuran lebar di atas 800, kita akan membuat tampilannya seperti ini:



19. Kita gunakan LayoutBuilder untuk memisahkan tampilan berdasarkan breakpoint.

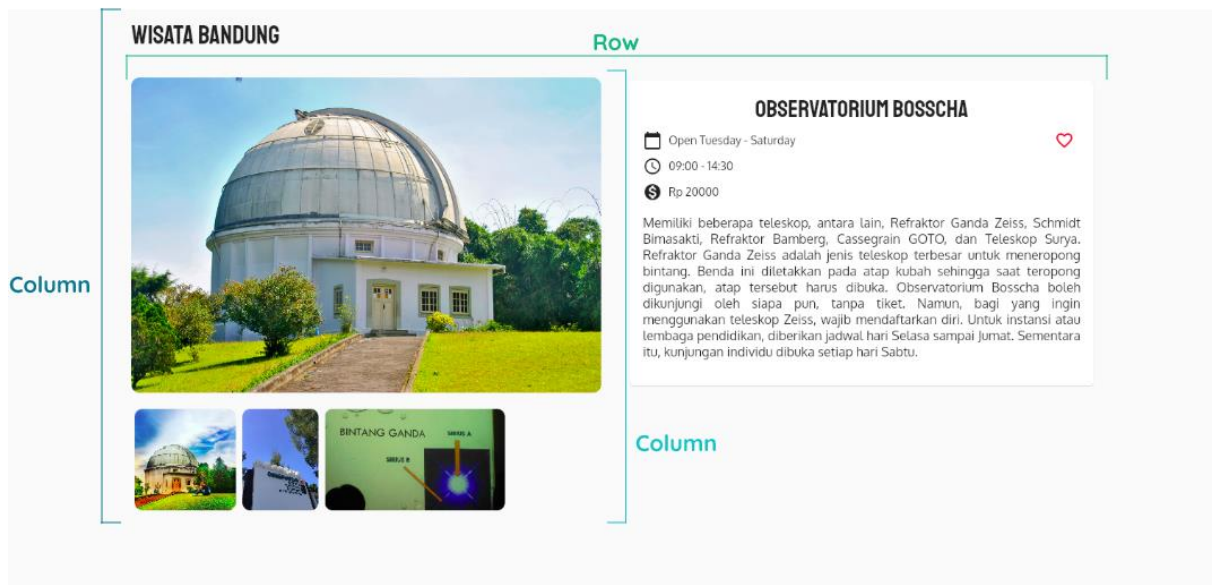
```
1. class DetailScreen extends StatelessWidget {  
2.   final TourismPlace place;  
3.  
4.   DetailScreen({required this.place});  
5.  
6.   @override  
7.   Widget build(BuildContext context) {  
8.     return LayoutBuilder(  
9.       builder: (BuildContext context, BoxConstraints constrain  
10.        ts) {  
11.         if (constraints.maxWidth > 800) {  
12.           return DetailWebPage(place: place);  
13.         } else {  
14.           return DetailMobilePage(place: place);  
15.         }  
16.       },  
17.     );  
18.   }
```

Pindahkan widget yang sudah ada ke dalam kelas DetailMobilePage.

```
19. class DetailMobilePage extends StatelessWidget {  
20.   final TourismPlace place;  
21.  
22.   const DetailMobilePage({required this.place});  
23.  
24.   @override  
25.   Widget build(BuildContext context) {  
26.     return Scaffold(...);  
27.   }  
28. }
```

Buat juga kelas serupa untuk DetailWebPage.

20. Tidak ada perbedaan konten yang ditampilkan pada halaman detail, tetapi hanya susunannya saja yang berbeda. Dari gambar sebelumnya apakah Anda dapat menganalisis bagaimana layoutnya disusun? Jika belum, maka kurang lebihnya akan seperti berikut:



21. Mari kita mulai dengan Column pertama. Tambahkan Text untuk judul, lalu Row untuk struktur layout di bawahnya. Tambahkan jarak menggunakan widget SizedBox.

```
1. Scaffold(  
2.   body: Column(  
3.     crossAxisAlignment: CrossAxisAlignment.start,  
4.     children: [  
5.       Text(  
6.         'Wisata Bandung',  
7.         style: TextStyle(  

```

```

8.         fontFamily: 'Staatliches',
9.         fontSize: 32,
10.      ),
11.    ),
12.    SizedBox(height: 32),
13.    Row(
14.      children: [],
15.    ),
16.  ],
17. ),
18. );

```

22. Widget Row akan berisi children seperti berikut:

```

1. Row(
2.   crossAxisAlignment: CrossAxisAlignment.start,
3.   children: [
4.     Expanded(
5.       child: Column(
6.         children: [],
7.       ),
8.     ),
9.     SizedBox(width: 32),
10.    Expanded(
11.      child: Card(...),
12.    ),
13.  ],
14. ),

```

Sementara Card untuk detail informasi tempat wisata memiliki widget seperti berikut:

```

15. Card(
16.   child: Container(
17.     padding: const EdgeInsets.all(16),
18.     child: Column(
19.       mainAxisAlignment: MainAxisAlignment.max,
20.       children: <Widget>[
21.         Container(
22.           child: Text(
23.             place.name,
24.             textAlign: TextAlign.center,
25.             style: TextStyle(
26.               fontSize: 30.0,
27.               fontFamily: 'Staatliches',
28.             ),
29.         ),
30.       ],
31.     Row(
32.       mainAxisAlignment: MainAxisAlignment.spaceBetween,
33.       children: [
34.         Row(
35.           children: <Widget>[

```



```

36.         Icon(Icons.calendar_today),
37.         SizedBox(width: 8.0),
38.         Text(
39.             place.openDays,
40.             style: informationTextStyle,
41.         ),
42.     ],
43. ),
44.     FavoriteButton(),
45. ],
46. ),
47. Row(
48.     children: <Widget>[
49.         Icon(Icons.access_time),
50.         SizedBox(width: 8.0),
51.         Text(
52.             place.openTime,
53.             style: informationTextStyle,
54.         ),
55.     ],
56. ),
57.     SizedBox(height: 8.0),
58.     Row(
59.         children: <Widget>[
60.             Icon(Icons.monetization_on),
61.             SizedBox(width: 8.0),
62.             Text(
63.                 place.ticketPrice,
64.                 style: informationTextStyle,
65.             ),
66.         ],
67.     ),
68.     Container(
69.         padding: EdgeInsets.symmetric(vertical: 16.0),
70.         child: Text(
71.             place.description,
72.             textAlign: TextAlign.justify,
73.             style: TextStyle(
74.                 fontSize: 16.0,
75.                 fontFamily: 'Oxygen',
76.             ),
77.         ),
78.     ),
79. ],
80. ),
81. ),
82. ),

```

23. Widget Column terakhir kita gunakan untuk menampilkan gambar dari aset dan daftar gambar dari internet.

```

1. Column(
2.     children: [
3.         ClipRRect(
4.             child: Image.asset(place.imageAsset),

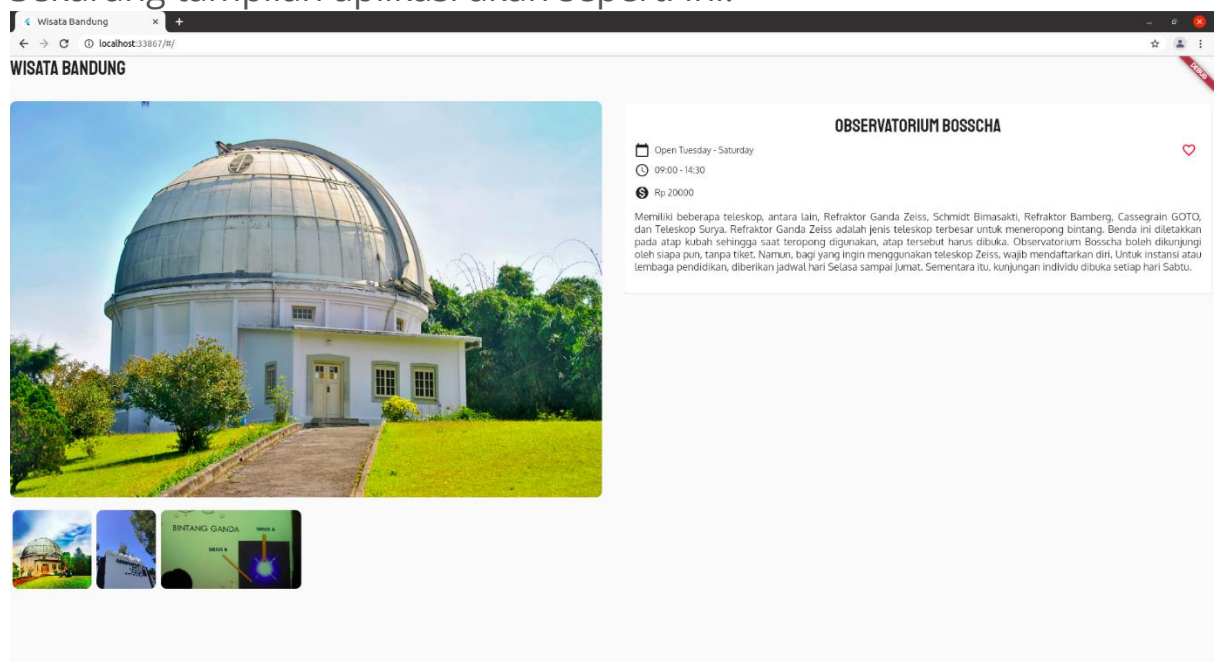
```

```

5.     borderRadius: BorderRadius.circular(10),
6.   ),
7.   SizedBox(height: 16),
8.   Container(
9.     height: 150,
10.    padding: const EdgeInsets.only(bottom: 16),
11.    child: ListView(
12.      scrollDirection: Axis.horizontal,
13.      children: place.imageUrl.map((url) {
14.        return Padding(
15.          padding: const EdgeInsets.all(4.0),
16.          child: ClipRect(
17.            borderRadius: BorderRadius.circular(10),
18.            child: Image.network(url),
19.          ),
20.        );
21.      }).toList(),
22.    ),
23.  ),
24. ],
25. ),

```

Sekarang tampilan aplikasi akan seperti ini:



24. Mari tambahkan Padding untuk memberikan jarak antara konten dengan pinggir halaman.

```

1. Scaffold(
2.   body: Padding(
3.     padding: const EdgeInsets.symmetric(
4.       vertical: 16,
5.       horizontal: 64,
6.     ),
7.     child: Column(...),

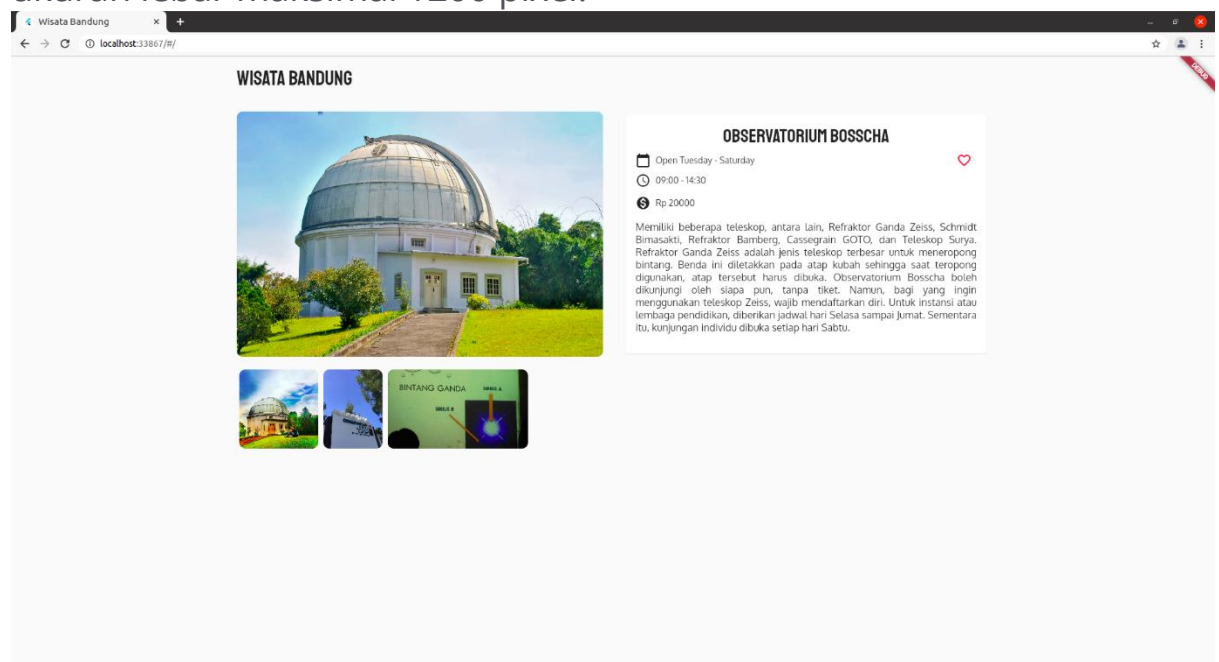
```

```
8.    ),
9.    );
```

25. Kita juga dapat menentukan ukuran lebar maksimum dari konten yang tampil menggunakan Container.

```
1.  Padding(
2.    padding: const EdgeInsets.symmetric(
3.      vertical: 16,
4.      horizontal: 64,
5.    ),
6.    child: Center(
7.      child: Container(
8.        width: 1200,
9.        child: Column(...),
10.      ),
11.    ),
12.  ),
```

Lihat perubahan aplikasi, sekarang halaman detail akan memiliki ukuran lebar maksimal 1200 pixel.



26. Satu hal yang penting untuk kita perhatikan adalah menghadirkan pengalaman pengguna di mana pun platform yang digunakan. Untuk itu, tambahkan Scrollbar pada daftar gambar.

```
1.  Scrollbar(
2.    isAlwaysShown: true,
3.    child: ListView(
4.      padding: const EdgeInsets.only(bottom: 16),
5.      scrollDirection: Axis.horizontal,
6.      children: place.imageUrl.map((url) {
7.        return Padding(...);
8.      }).toList(),
```

```
9.    ),  
10.   ),
```

27. Jika Anda menggunakan touchpad pada laptop, gestur swipe pada scrollbar akan berjalan dengan normal. Namun, ketika melakukan drag menggunakan mouse, fitur scrollbar tidak berjalan. Solusi yang bisa Anda lakukan adalah menggunakan ScrollController. Inisialisasikan objek ScrollController di atas method build().

```
1. final _scrollController = ScrollController();
```

Tambahkan _scrollController pada Scrollbar dan ListView.

```
2. Scrollbar(  
3.   isAlwaysShown: true,  
4.   controller: _scrollController,  
5.   child: ListView(  
6.     controller: _scrollController,  
7.     padding: const EdgeInsets.only(bottom: 16),  
8.     scrollDirection: Axis.horizontal,  
9.     children: place.imageUrls.map((url) {  
10.       return Padding(...);  
11.     }).toList(),  
12.   ),  
13. ),
```

28. Seperti halnya TextEditingController, ScrollController juga harus di-dispose ketika widget sudah tidak lagi digunakan. Ubahlah DetailWebPage menjadi StatefulWidget supaya kita bisa memanggil method dispose.

```
class DetailMobilePage extends StatelessWidget {...}  
  
class DetailWebPage extends StatefulWidget {  
  final TourismPlace place;  
  
  DetailWebPage({required this.place});  
  
  final _scrollController = ScrollController();  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      body: Padding(  
        padding: const EdgeInsets.symmetric(  
          vertical: 16,  
          horizontal: 64
```

29. Tambahkan method dispose setelah method build() lalu panggil _scrollController.dispose();

```
1. @override  
2. void dispose() {
```



```
3. _scrollController.dispose();  
4. super.dispose();  
5. }
```

30. Selamat! Sekarang tampilan aplikasi Wisata Bandung sudah lebih sesuai untuk ukuran resolusi layar yang lebih besar. Anda telah belajar banyak menyusun halaman menggunakan widget, hingga membuat halaman yang dapat responsif terhadap perbedaan ukuran layar. Kode selengkapnya untuk codelab ini dapat Anda temukan pada tautan: <https://github.com/dicodingacademy/a159-flutter-pemula-labs/tree/codelab4-final>.