Responsive Layout

Seperti yang kita tahu, Flutter merupakan framework untuk mengembangkan aplikasi pada berbagai platform. Pada platform mobile sendiri tersedia banyak ukuran layar dari ukuran jam hingga tablet. Ditambah Flutter baru saja mengumumkan dukungan untuk platform web dan desktop. Itu artinya, satu hal yang penting untuk kita pahami adalah bagaimana menerapkan layout yang mampu beradaptasi dengan berbagai ukuran layar yang berbeda.

Pada materi ini kita akan mulai membahas bagaimana mengimplementasikan layout yang responsif.

MediaQuery

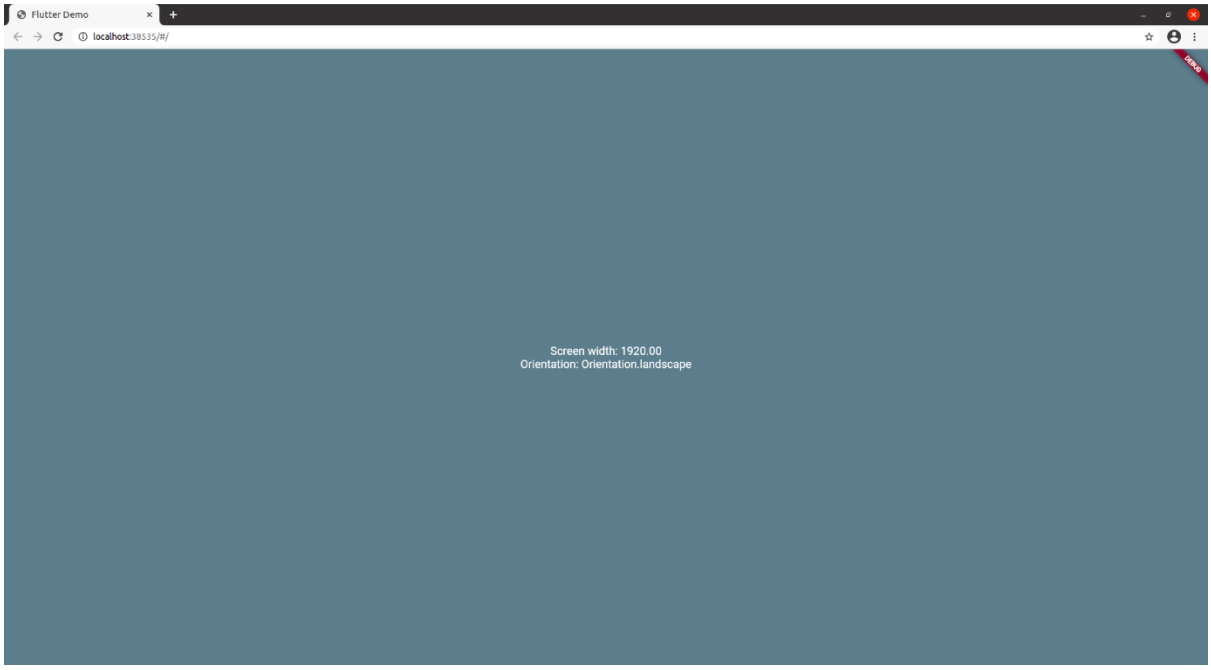
Pendekatan pertama yang akan kita lakukan adalah menggunakan Media Query. Jika Anda sudah familier dengan pengembangan web, mungkin Anda sudah tidak asing dengan konsep ini. MediaQuery adalah kelas yang dapat kita gunakan untuk mendapatkan ukuran dan juga orientasi layar.

Mari kita lihat contoh penerapan MediaQuery.

```
1. class HomePage extends StatelessWidget {
2.   @override
3.   Widget build(BuildContext context) {
4.     Size screenSize = MediaQuery.of(context).size;
5.     Orientation orientation = MediaQuery.of(context).orientation;
6.
7.     return Scaffold(
8.       backgroundColor: Colors.blueGrey,
9.       body: Column(
10.          mainAxisAlignment: MainAxisAlignment.center,
11.          crossAxisAlignment: CrossAxisAlignment.stretch,
12.          children: [
13.            Text(
14.              'Screen width: ${screenSize.width.toStringAsFixed(2)}',
15.              style: TextStyle(color: Colors.white, fontSize: 18),
16.              textAlign: TextAlign.center,
17.            ),
18.            Text(
19.              'Orientation: $orientation',
20.              style: TextStyle(color: Colors.white, fontSize: 18),
```

```
21.                 textAlign: TextAlign.center,
22.               ),
23.             ],
24.           ),
25.       );
26.     }
27.   }
```

Sekarang jalankan aplikasi untuk melihat ukuran layarnya.




12:31
DEBUG
Screen width: 411.43
Orientation: Orientation.portrait

LayoutBuilder

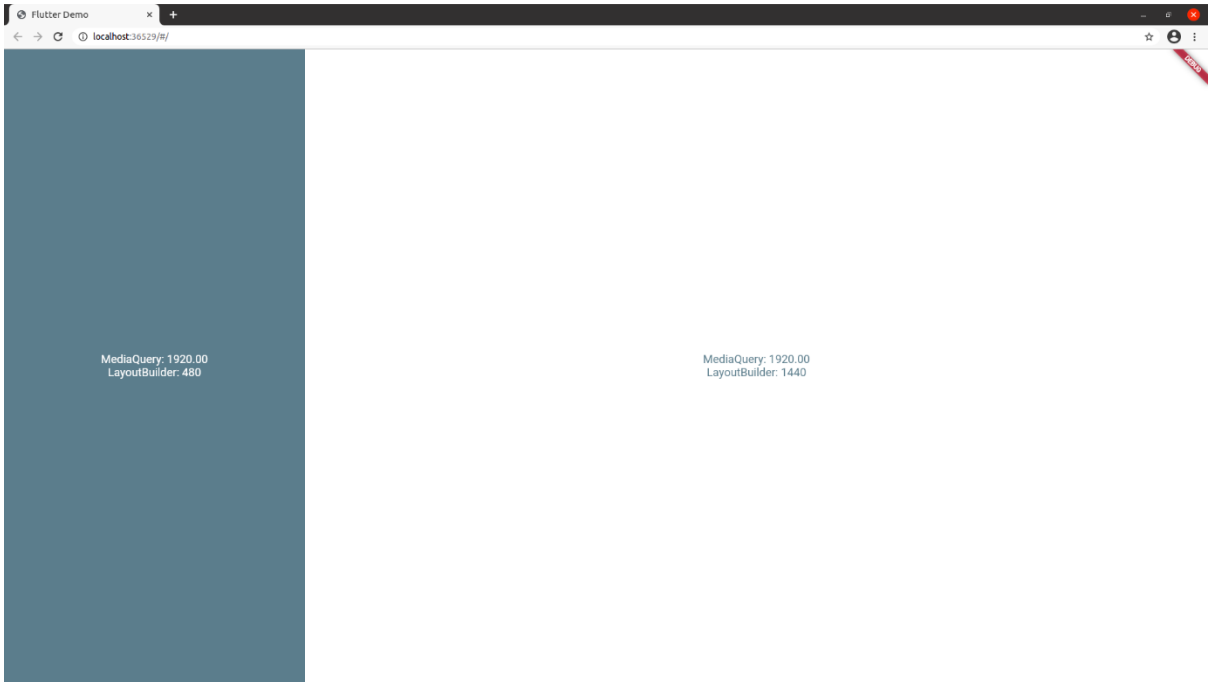
Cara lain yang bisa kita gunakan adalah dengan widget LayoutBuilder. Perbedaan umum antara MediaQuery dan Layout Builder adalah MediaQuery akan mengembalikan ukuran layar yang digunakan, sedangkan LayoutBuilder mengembalikan ukuran maksimum dari widget tertentu.

Berikut ini adalah contoh kode yang menunjukkan perbedaan antara MedaQuery dan LayoutBuilder:

```
1. class HomePage extends StatelessWidget {
2.   @override
3.   Widget build(BuildContext context) {
4.     Size screenSize = MediaQuery.of(context).size;
5.
6.     return Scaffold(
7.       backgroundColor: Colors.blueGrey,
8.       body: Row(
9.         children: [
10.           Expanded(
11.             child: LayoutBuilder(
12.               builder: (BuildContext context, BoxConstraints constr
   aints) {
13.                 return Column(
14.                   mainAxisAlignment: MainAxisAlignment.center,
15.                   crossAxisAlignment: CrossAxisAlignment.stretch,
16.                   children: [
17.                     Text(
18.                       'MediaQuery: ${screenSize.width.toStringAsFix
   ed(2)}',
19.                       style: TextStyle(color: Colors.white, fontSiz
   e: 18),
20.                       textAlign: TextAlign.center,
21.                     ),
22.                     Text(
23.                       'LayoutBuilder: ${constraints.maxWidth}',
24.                       style: TextStyle(color: Colors.white, fontSiz
   e: 18),
25.                       textAlign: TextAlign.center,
26.                     ),
27.                   ],
28.                 );
29.               },
30.             ),
31.           ),
32.           Expanded(
33.             flex: 3,
34.             child: LayoutBuilder(
```

```
                builder: (BuildContext context, BoxConstraints constr
aints) {
                  return Container(
                    color: Colors.white,
                    child: Column(
                      mainAxisAlignment: MainAxisAlignment.center,
                      crossAxisAlignment: CrossAxisAlignment.stretch,
                      children: [
                        Text(
                          'MediaQuery: ${screenSize.width.toStringAsF
ixed(2)}',
                          style: TextStyle(color: Colors.blueGrey, fo
ntSize: 18),
                          textAlign: TextAlign.center,
                        ),
                        Text(
                          'LayoutBuilder: ${constraints.maxWidth}',
                          style: TextStyle(color: Colors.blueGrey, fo
ntSize: 18),
                          textAlign: TextAlign.center,
                        ),
                      ],
                    ),
                  );
                },
              ),
            ),
          ],
        ),
      );
    }
  }
```

Hasil ketika dijalankan pada browser akan seperti ini:



Ubahlah ukuran jendela browser untuk melihat perubahan ukuran layar atau media yang digunakan.

Dengan mendapatkan ukuran lebar dan tinggi layar seperti di atas, kita bisa menentukan tampilan konten berdasarkan ukuran layar yang digunakan. Dalam responsive design, terdapat breakpoint yang merupakan “titik” di mana layout akan beradaptasi untuk memberikan pengalaman pengguna sebaik mungkin.

Dengan kode di bawah ini berarti akan terdapat tiga model tampilan berdasarkan ukuran layar:

```
1. class ResponsivePage extends StatelessWidget {
2.   @override
3.   Widget build(BuildContext context) {
4.     return Scaffold(
5.       appBar: AppBar(),
6.       body: LayoutBuilder(
7.         builder: (BuildContext context, BoxConstraints constraints)
   {
8.           if (constraints.maxWidth < 600) {
9.             return ListView(
10.                children: _generateContainers(),
11.              );
12.            } else if (constraints.maxWidth < 900) {
13.              return GridView.count(
14.                crossAxisCount: 2,
15.                children: _generateContainers(),
16.              );
```

```
            } else {
              return GridView.count(
                crossAxisCount: 6,
                children: _generateContainers(),
              );
            }
          },
        ),
      );
    }

    List<Widget> _generateContainers() {
      return List<Widget>.generate(20, (index) {
        return Container(
          margin: const EdgeInsets.all(8),
          color: Colors.blueGrey,
          height: 200,
        );
      });
    }
  }
```

Berikut adalah tampilan dari kode di atas ketika dijalankan: