

Image

Dalam pengembangan suatu aplikasi kita tidak akan lepas dari image atau gambar untuk membuat tampilan semakin menarik. Pada materi kali ini kita akan belajar bagaimana menampilkan gambar dari internet dan project asset.

Image.network

Untuk menampilkan gambar yang bersumber dari internet, kita akan menggunakan method `Image.network`. Cara penulisan method ini sebagai berikut:

1. `Image.network(url)`

Method ini cukup menambahkan URL gambar dari internet dan kita pun dapat menambahkan `width` dan `height` juga. Di bawah ini adalah contoh penggunaan `Image.network`:

```
1. class FirstScreen extends StatelessWidget {  
2.   @override  
3.   Widget build(BuildContext context) {  
4.     return Scaffold(  
5.       appBar: AppBar(  
6.         title: Text('First Screen'),  
7.       ),  
8.       body: Center(  
9.         child: Image.network(  
10.           'https://picsum.photos/200/300',  
11.           width: 200,  
12.           height: 200,  
13.         ),  
14.       ),  
15.     );  
16.   }  
17. }
```

Pada kode di atas kita panggil method `Image.network` dengan url `https://picsum.photos/200/300` lalu beri `width` dan `height` masing-

masing 200. Sehingga hasilnya seperti berikut:



5:27

DEBUG

First Screen

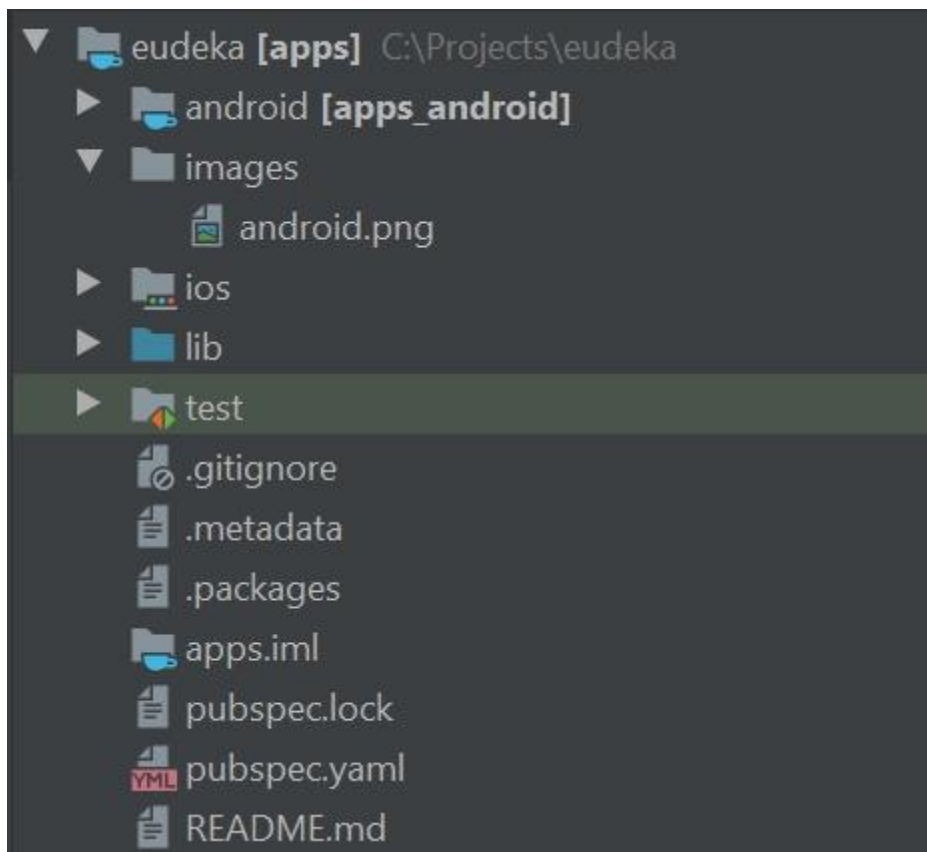


Image.asset

Selain melalui internet, kita juga dapat menampilkan gambar yang bersumber dari asset *project*.. Asset di sini berupa gambar-gambar yang nantinya didaftarkan pada *project*. Untuk mendaftarkan asset gambar pada project kita harus menambahkannya pada berkas **pubspec.yaml**.

Pertama kita harus menambahkan terlebih dahulu gambar yang akan didaftarkan ke dalam folder project kita. Saat ini Flutter mendukung beberapa jenis format gambar, seperti JPEG, PNG, GIF, Animated GIF, WebP, Animated WebP, BMP, dan WBMP. Di luar format tersebut, Flutter akan memanfaatkan API dari masing-masing platform. Jika platform native mendukung format gambar yang digunakan, maka gambar tersebut akan bisa di-render oleh Flutter.

Pada contoh berikut kita menambahkan folder **images/** pada folder project.



Masukkan berkas gambar yang ingin Anda gunakan ke dalam folder **image**. Sebagai contoh kita menggunakan gambar bernama **android.png**.

Setelah menambahkan gambar pada *project*, saatnya kita mendaftarkan gambar tersebut pada **pubspec.yaml**.

Di dalam berkas **pubspec.yaml** kita akan langsung pada bagian Flutter seperti di bawah ini:

```
1. ...
2. flutter:
3.
4.   uses-material-design: true
5.
6.   # To add assets to your application, add an assets section, like this:
7.   # assets:
8.   #   - images/a_dot_burr.jpeg
9.   #   - images/a_dot_ham.jpeg
10.
11. ...
```

Daftarkan asset gambar seperti berikut:

```
1. ...
2. flutter:
3.   uses-material-design: true
4.
5.   assets:
6.     - images/android.png
7. ...
```

Hapus juga tanda pagar (#) atau komentar yang tidak diperlukan. Perhatikan pula indentasi kodenya. **assets:** berada sejajar dengan **uses-material-design:** yaitu berjarak 2 spasi dari ujung dan berada di dalam **flutter:** sedangkan **- images/android.png** berada di dalam **assets:** dan berjarak 4 spasi dari ujung.

Pada contoh di atas kita telah menambahkan asset yang berisi lokasi gambar atau aset yang ingin kita gunakan. Karena kita menambahkan gambar **android.png** pada folder *images*, maka lokasi gambar tersebut adalah **images/android.png**.

Apabila ada banyak gambar yang kita masukkan ke dalam lokasi folder, dibandingkan menuliskan lokasi gambar satu per satu, kita bisa langsung menuliskan folder **images/** seperti berikut:

```
1. ...
2. flutter:
3.
4.   uses-material-design: true
5.
6.   assets:
7.     - images/
8. ...
```

Setelah menambahkan assets kita harus me-*refresh* **pubspec.yaml** dengan cara **save** berkas **pubspec.yaml** bila menggunakan visual studio code atau menekan '**Packages get**' yang ada di pojok kanan atas untuk android studio.

Kita telah mendaftarkan suatu asset. Sekarang kita akan panggil asset tersebut pada kode kita dengan method **Image.asset**. Cara penulisannya seperti berikut:

```
1. Image.asset(lokasi_asset)
```

Contoh dalam kodenya akan seperti berikut:

```
1. class FirstScreen extends StatelessWidget {
2.
3.   @override
4.   Widget build(BuildContext context) {
5.     return Scaffold(
6.       appBar: AppBar(
7.         leading: IconButton(
8.           icon: Icon(Icons.menu, color: Colors.white,)),
9.       ),
10.      title: Text('First Screen'),
11.      actions: <Widget>[
12.        IconButton(
13.          icon: Icon(Icons.search, color: Colors.white,)),
14.      ],
15.    ],
16.  ),
17.  body: Center(
18.    child: Image.asset('images/android.png', width: 200, height
19.      : 200),
20.  ),
21.  floatingActionButton: FloatingActionButton(
22.    child: Icon(Icons.add),
```

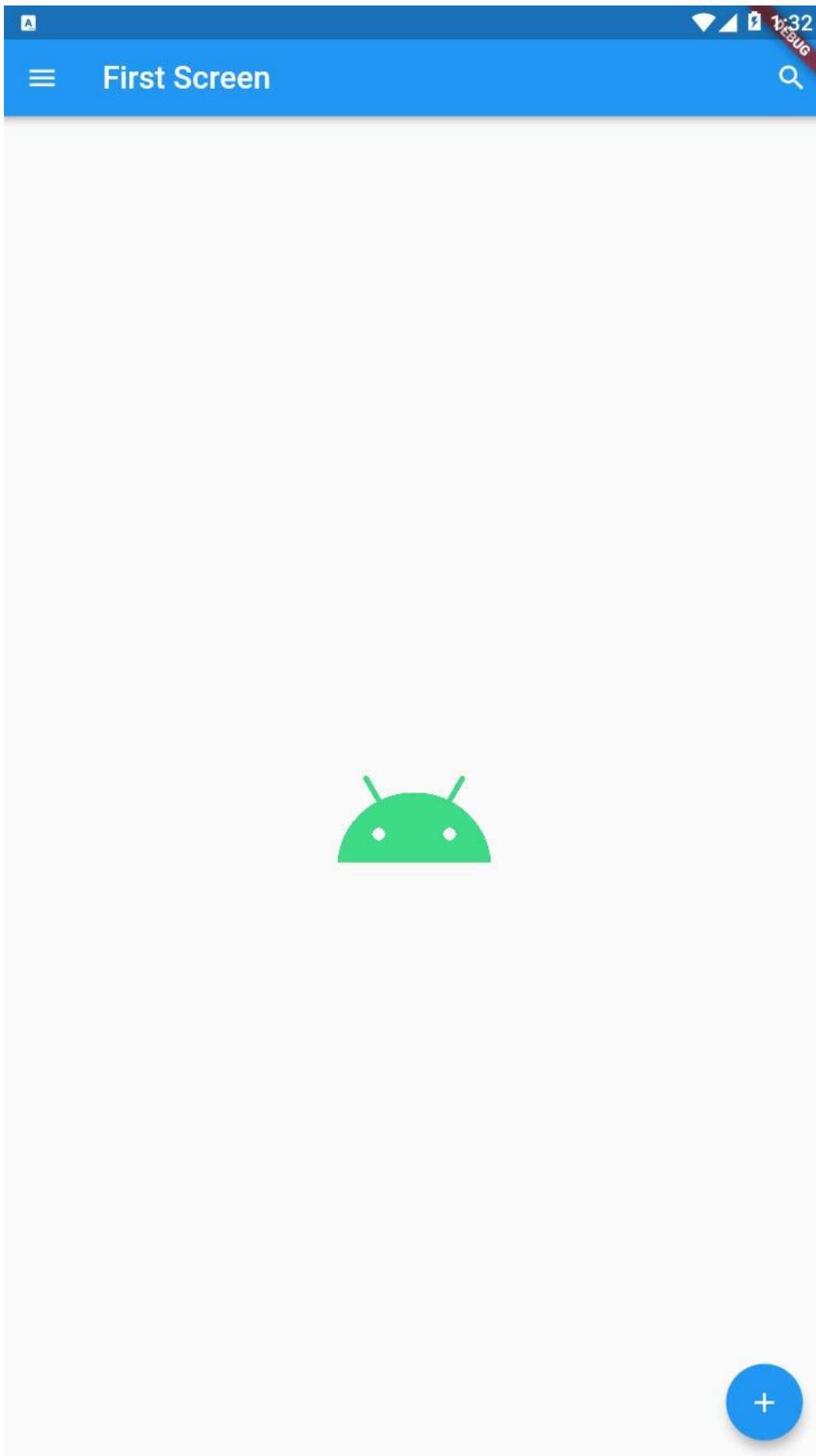
22.),

23.);

24. }

25. }

Jika kita jalankan aplikasi Flutter, maka gambar akan tampil seperti berikut:



Untuk mempelajari widget Image lebih lanjut, Anda dapat membaca dokumentasinya pada tautan [Image Class](#).