Widget Playground: Eksplorasi Widget Dasar Flutter

Pendahuluan

Dalam dunia Flutter, *widget* adalah raja! Mereka adalah elemen fundamental yang membentuk tampilan antarmuka aplikasi. Bayangkan *widget* seperti balok-balok Lego yang bisa kamu susun untuk membuat berbagai macam bentuk, dari yang sederhana hingga yang kompleks. Setiap elemen yang kamu lihat di aplikasi Flutter, seperti teks, gambar, tombol, dan layout, semuanya terbuat dari *widget*.

Flutter menyediakan banyak *widget* siap pakai yang bisa langsung kamu gunakan. Di minggu ini, kita akan bermain-main dengan beberapa *widget* dasar dan belajar bagaimana menyusunnya untuk membuat tampilan yang menarik.

1. Mengenal Widget

- **Apa itu Widget?**

  Widget adalah komponen UI yang mendeskripsikan bagaimana tampilannya harus terlihat berdasarkan konfigurasi dan *state* saat ini. Mereka seperti *blueprint* yang memberi tahu Flutter bagaimana cara merender elemen di layar. *Widget* disusun secara hierarkis, membentuk struktur seperti pohon yang disebut *widget tree*.

- **Stateless vs. Stateful Widget**

  Ada dua jenis utama *widget* di Flutter:

  1. **Stateless Widget:** *Widget* yang tampilannya tidak berubah seiring waktu. Contohnya: Text, Icon, Image.
  2. **Stateful Widget:** *Widget* yang tampilannya bisa berubah berdasarkan interaksi pengguna atau data yang diterima. Contohnya: Checkbox, TextField, Slider.

- **Contoh Kode:**

```dart
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('Contoh Widget'),
        ),
        body: Center(
          child: Text('Hello World!'),
        ),
      ),
    );
  }
}
```

2. Widget Text dan Image

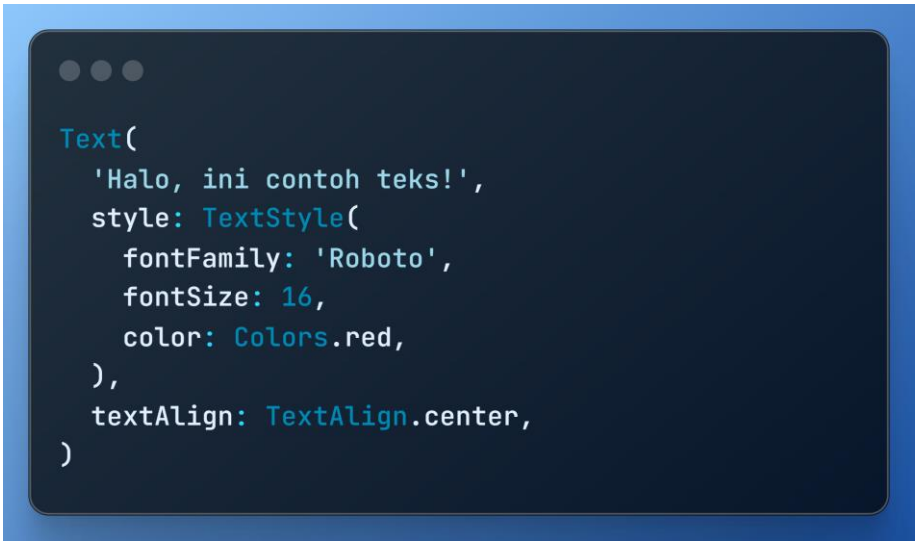
Dua widget dasar yang penting di Flutter: Text dan Image. Keduanya adalah kunci untuk menampilkan informasi dan visual di aplikasi kamu.

- **Widget Text**

  Sesuai namanya, widget Text digunakan untuk menampilkan teks di aplikasi Flutter. Sederhana, kan? Tapi jangan salah, widget ini punya banyak fitur untuk mengatur tampilan teks sesuai keinginanmu.

  - **Style:** Kamu bisa atur jenis font, ukuran, warna, ketebalan, dan *style* lainnya dengan properti style. Misalnya, TextStyle(fontSize: 20, color: Colors.blue, fontWeight: FontWeight.bold) akan membuat teks berukuran 20, berwarna biru, dan tebal.

- **TextAlign:** Properti ini untuk mengatur perataan teks, mau rata kiri, kanan, tengah, atau *justify*.
- **MaxLines & Overflow:** Kalau teksnya kepanjangan, kamu bisa batasi jumlah baris dengan maxLines dan atur tampilan teks yang kepanjangan dengan overflow (misalnya, menampilkan titik-titik di akhir).
- **Contoh Kode:**



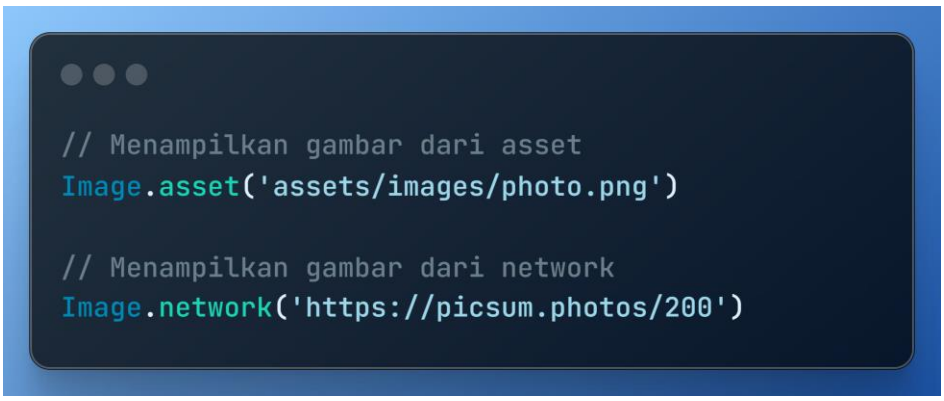
```
Text(
  'Halo, ini contoh teks!',
  style: TextStyle(
    fontFamily: 'Roboto',
    fontSize: 16,
    color: Colors.red,
  ),
  textAlign: TextAlign.center,
)
```

- **Widget Image**

  Widget Image digunakan untuk menampilkan gambar di aplikasi. Kamu bisa menampilkan gambar dari berbagai sumber:

  - **Asset:** Gambar yang disimpan di dalam project Flutter. Gunakan Image.asset('nama_file.jpg').
  - **Network:** Gambar yang diambil dari internet. Gunakan Image.network('url_gambar').
  - **File:** Gambar yang diambil dari *storage* perangkat. Gunakan Image.file(File('path_gambar')).
  - Kamu juga bisa atur ukuran dan tampilan gambar dengan properti width, height, dan fit.

- **Contoh Kode:**



```
// Menampilkan gambar dari asset
Image.asset('assets/images/photo.png')

// Menampilkan gambar dari network
Image.network('https://picsum.photos/200')
```

3. Container dan Styling

Container ini seperti kotak ajaib di Flutter! Dia bisa jadi wadah untuk *widget* lain dan kamu bisa kreasikan tampilannya sesukamu.

Container punya banyak properti untuk mengatur tampilan, seperti:

- **color**: Memberi warna latar belakang pada Container. Misalnya, color: Colors.blue akan membuat *container* berwarna biru.
- **padding**: Memberi jarak antara isi *container* dengan tepinya. Misalnya, padding: EdgeInsets.all(20) akan memberi jarak 20 piksel di semua sisi.
- **margin**: Memberi jarak antara *container* dengan *widget* di sekitarnya. Misalnya, margin: EdgeInsets.only(top: 10) akan memberi jarak 10 piksel di atas *container*.
- **width & height**: Mengatur lebar dan tinggi *container*.
- **alignment**: Mengatur posisi isi *container*, mau di tengah, di kiri atas, di kanan bawah, dan sebagainya.

Tapi yang paling seru itu properti decoration! Di sini kamu bisa menambahkan berbagai efek visual, seperti:

- **border**: Memberi garis tepi pada *container*. Kamu bisa atur warna, ketebalan, dan *style* garisnya.
- **borderRadius**: Membuat sudut *container* jadi melengkung. Misalnya, borderRadius: BorderRadius.circular(10) akan membuat sudutnya melengkung dengan radius 10 piksel.

- **boxShadow**: Memberi efek bayangan pada *container*. Kamu bisa atur warna, *offset*, dan *blur* bayangannya.
- **gradient**: Memberi efek gradasi warna pada *container*.

Contoh Kode:

```dart
Container(
  width: 200,
  height: 150,
  color: Colors.amber,
  padding: EdgeInsets.all(16),
  margin: EdgeInsets.only(bottom: 20),
  alignment: Alignment.center,
  child: Text('Container Keren!'),
  decoration: BoxDecoration(
    border: Border.all(color: Colors.black, width: 2),
    borderRadius: BorderRadius.circular(8),
    gradient: LinearGradient(
      colors: [Colors.yellow, Colors.orange],
    ),
    boxShadow: [
      BoxShadow(
        color: Colors.grey.withOpacity(0.5),
        spreadRadius: 3,
        blurRadius: 7,
        offset: Offset(0, 3),
      ),
    ],
  ),
)
```

4. Row dan Column

Dua *widget* ini adalah kunci untuk mengatur tata letak (*layout*) di Flutter. Mereka akan membantumu menyusun *widget-widget* lain secara rapi, baik horizontal maupun vertikal.

- **Widget Row**

  Row digunakan untuk menyusun *widget* secara horizontal, alias berjejer dari kiri ke kanan. Bayangkan seperti barisan orang lagi antre, gitu deh!

  - **mainAxisAlignment**: Properti ini mengatur posisi *widget-widget* di dalam Row. Mau rata kiri, rata kanan, di tengah, atau tersebar rata? Kamu bisa atur dengan properti ini.
  - **crossAxisAlignment**: Properti ini mengatur posisi *widget-widget* di dalam Row secara vertikal. Mau rata atas, rata bawah, atau di tengah? Atur aja pakai properti ini.
  - **Contoh Kode:**

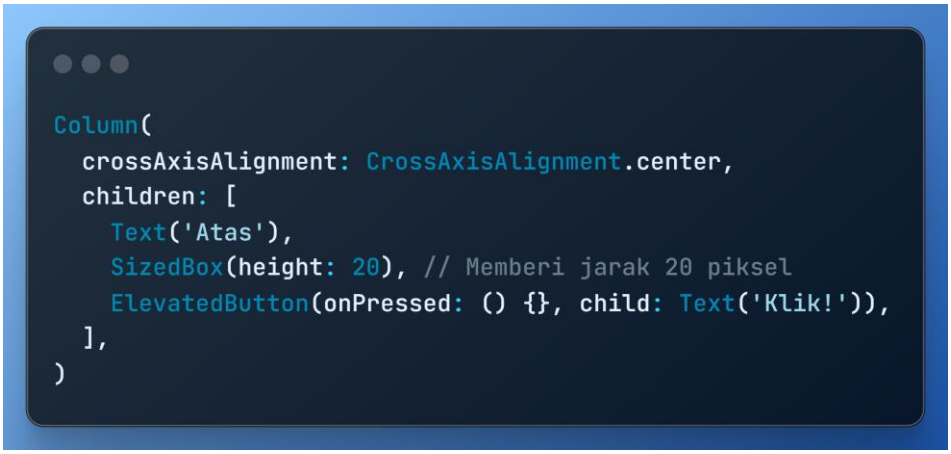
```
Row(
  mainAxisAlignment: MainAxisAlignment.spaceAround,
  children: [
    Text('Satu'),
    Icon(Icons.star),
    Image.asset('assets/images/gambar.png'),
  ],
)
```

- **Widget Column**

  Column digunakan untuk menyusun *widget* secara vertikal, alias bertumpuk dari atas ke bawah. Kayak tumpukan buku, gitu!

  - **mainAxisAlignment**: Sama seperti di Row, properti ini mengatur posisi *widget-widget* di dalam Column. Tapi bedanya, ini mengatur posisi secara vertikal.
  - **crossAxisAlignment**: Properti ini mengatur posisi *widget-widget* di dalam Column secara horizontal.

- **Contoh Kode:**



```
Column(
  crossAxisAlignment: CrossAxisAlignment.center,
  children: [
    Text('Atas'),
    SizedBox(height: 20), // Memberi jarak 20 piksel
    ElevatedButton(onPressed: () {}, child: Text('Klik!')),
  ],
)
```

5. Challenge

Yeay, waktunya tantangan! Setelah belajar tentang berbagai widget dasar, saatnya menguji kemampuanmu dengan tantangan seru di akhir materi Widget Playground. Tujuan dari tantangan ini adalah untuk mengasah kreativitas dan kemampuan problem-solving kamu dalam menyusun widget dan membuat layout.

Kartu Profil

Buat layout kartu profil yang menarik dengan informasi seperti:

- Foto profil
- NPM
- Nama lengkap
- Email
- Bio singkat
- Tombol "Follow"

Gunakan Container untuk membuat *card* dan atur *styling*-nya dengan decoration. Susun elemen-elemen informasi dengan Row dan Column.