

Bab 5

Praktikum

Pemrograman Widget



Flutter

Dosen : Bambang Sugiarto, ST, MT

**Program Studi S1 Teknik Informatika
Fakultas Teknik
Universitas Sangga Buana YPKP Bandung**

*Dirangkum dari berbagai sumber referensi
(hanya untuk penggunaan internal/tidak untuk dipublikasikan)*

Tujuan Praktikum

- Mahasiswa mengetahui pemrograman widget pada Flutter.
- Mahasiswa mampu membuat pemrograman widget pada Flutter.
- Mahasiswa mampu membuat implementasi pemrograman widget dengan membuat aplikasi mobile sederhana pada Flutter.



Pendahuluan

- Sesuai dengan materi yang disampaikan pada kuliah teori tentang kosep dasar widget pada flutter, maka pada praktikum kita kali ini akan mencoba untuk mengaplikasikan widget pada pemrograman Flutter.
- Praktikum kita yang pertama adalah mengaplikasikan stateless widget yang hanya bertugas untuk menampilkan sesuatu secara statis tanpa melakukan tracking perubahan data dari waktu ke waktu.
- Sedangkan pada praktikum yang kedua kita akan mencoba mengaplikasikan stateful widget yang bisa merespon terhadap interaksi user, dan bisa *tracking* perubahan data, lalu merender ulang layout sesuai perubahan data, maka ia adalah stateful.



Praktikum Stateless Widget

<https://jagongoding.com/android/flutter/dasar/stateless-widget/>

- Buatlah project baru pada Flutter dan beri nama StatelessProject
- Sesuai dengan praktikum sebelumnya, dimana kita menampilkan “HelloWorld”
- Untuk membuat stateless widget, kita harus membuat sebuah *class* baru yang meng-*extends class* StatelessWidget.
- Sebagai contoh, di sini kita akan membuat widget baru dengan nama TeksUtama.
 - Langkah-langkahnya:
 1. Buka file main.dart
 2. Tambahkan kode berikut di bagian paling atas (setelah import) atau di paling bawah :



```
class TeksUtama extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    // TODO: implement build  
  }  
}
```

- Lalu pada fungsi build, return widget Column yang berisi dua buah teks :

- Teks **Halo**

- Dan teks **Dunia**

Seperti pada gambar berikut ini :



```
...
Widget build(BuildContext context) {
  return Column(
    children: [
      Text(
        'Halo',
        textDirection: TextDirection.ltr,
      ),
      Text(
        'Dunia',
        textDirection: TextDirection.ltr,
        style: TextStyle(
          fontSize: 45,
          fontWeight: FontWeight.bold,
          color: Colors.red
        ),
      ),
    ],
  );
}
```



Pada fungsi *main()*, kita ganti atribut/argumen body dari *Text* menjadi *TeksUtama()* (yaitu widget berbasis stateless yang telah kita buat sendiri).

```
...  
  
body: Center(  
  child: TeksUtama(),  
),  
  
...
```

- Coba jalankan aplikasi, jika tidak ada error harusnya sekarang kita mendapatkan output seperti berikut :

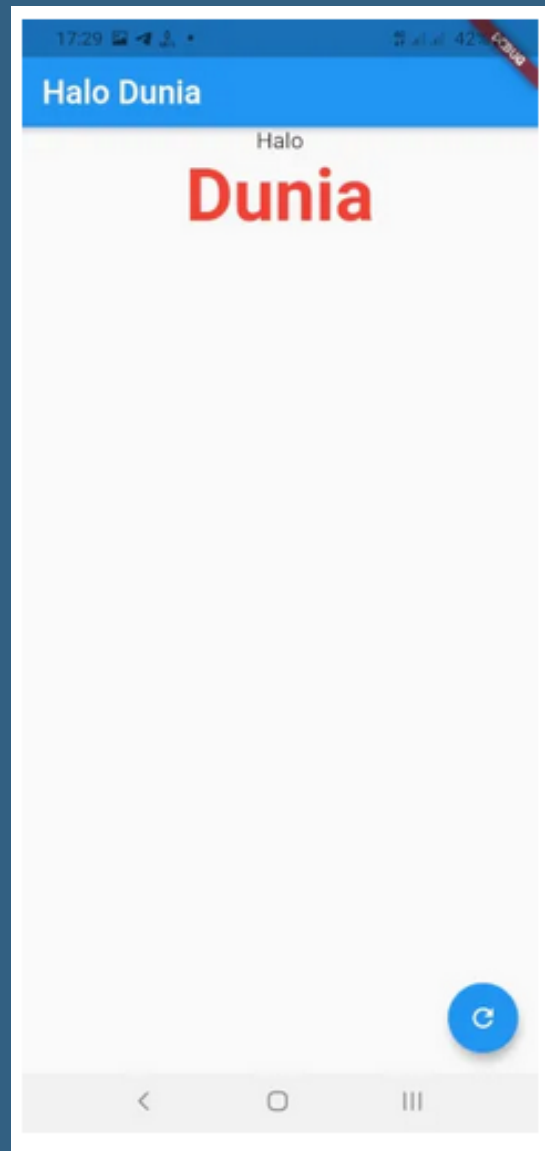


Pada fungsi *main()*, kita ganti atribut/argumen body dari *Text* menjadi *TeksUtama()* (yaitu widget berbasis stateless yang telah kita buat sendiri).

```
...  
  
body: Center(  
    child: TeksUtama(),  
),  
  
...
```

- Coba jalankan aplikasi, jika tidak ada error harusnya sekarang kita mendapatkan output seperti berikut :





Parameter Widget :

- Kembali lagi ke class TeksUtama.
- Sebelum fungsi build, tambahkan kode program berikut:

```
...  
class TeksUtama extends StatelessWidget {  
  
    final String teks1;  
    final String teks2;  
  
    TeksUtama({ this.teks1, this.teks2 });  
  
    @override  
    Widget build(BuildContext context) {  
        ...  
    }  
}
```



Lalu pada fungsi build(), ganti:

1. string "Apa kabar" menjadi teks1
2. dan string "Dunia" menjadi teks2.

```
...  
@override  
Widget build(BuildContext context) {  
  return Column(  
    children: [  
      Text(  
        teks1,  
        textDirection: TextDirection.ltr,  
      ),  
      Text(  
        teks2,  
        textDirection: TextDirection.ltr,  
        style: TextStyle(  
          fontSize: 45,  
          fontWeight: FontWeight.bold,  
          color: Colors.red  
        ),  
      ),  
    ],  
  );  
}
```



- Pada fungsi main(), widget Scaffold, pada argumen/atribut body, kita ubah pemanggilan TeksUtama() dengan menambahkan dua parameter, yaitu parameter teks1 dan teks2.

```
...  
body: Center(  
  child: TeksUtama(  
    teks1: 'Halo',  
    teks2: 'Dunia',  
  ),  
,  
...  

```



- Ketika kita jalankan, harusnya tampilannya akan sama saja dengan sebelumnya.
- Bedanya, sekarang kode program kita menjadi lebih rapi karena widget-nya sudah terkelompokkan.
- Kita juga bisa menampilkan widget TeksUtama berkali-kali seperti berikut :

```
...  
body: Center(  
  child: Column(  
    children: [  
      TeksUtama(  
        teks1: 'Halo',  
        teks2: 'Dunia',  
      ),  
      TeksUtama(  
        teks1: 'Apa kabar',  
        teks2: 'Budi',  
      ),  
    ],  
  ),  
)
```



```
TeksUtama(  
    teks1: 'Apa kabar',  
    teks2: 'Andi',  
),  
TeksUtama(  
    teks1: 'Apa kabar',  
    teks2: 'Santi',  
),  
]  
,  
,  
...
```

- Jika kita jalankan, maka outputnya adalah sebagai berikut:





Praktikum Stateful Widget

<https://jagongoding.com/android/flutter/dasar/stateful-widget/>

- Langkah pertama ini kita akan membuat sebuah state baru. Langkah-langkahnya:
 1. Buka file lib/main.dart
 2. Rename class TeksUtama menjadi StateTeksUtama
 3. Ubah extends class dari StatelessWidget menjadi State<TeksUtama>
 4. Hapus kode program di bawah:

```
final String teks1;  
final String teks2;  
  
TeksUtama({ this.teks1, this.teks2 });
```



- Lalu ganti dengan kode berikut:

```
var listNama = [  
    'Budi', 'Andi', 'Sari', 'Rudi',  
    'Nurul', 'Huda', 'Lendis', 'Fabri'  
];  
var listWarna = [  
    Colors.red, Colors.purple, Colors.teal, Colors.lime,  
    Colors.indigo, Colors.deepPurple, Colors.cyan,  
    Colors.blue, Colors.yellow  
];  
int index = 0;
```

- Pada kode di atas, kita membuat variabel baru pada kelas StateTeksUtama untuk menyimpan list nama, list warna, dan index yang akan digunakan untuk menentukan mana nama dan warna yang aktif.



Update tampilan Widget :

- Update fungsi build pada kelas StateTeksUtama dengan mengubah:
 1. teks1 menjadi "Apa kabar"
 2. teks2 menjadi listNama[index % listNama.length]
 3. dan Colors.red menjadi listWarna[index % listWarna.length]
- Perhatikan kode yang ter-*highlight* di bawah ini:

```
@override
Widget build(BuildContext context) {
  return Column(
    children: [
      Text(
        'Apa kabar',
        textDirection: TextDirection.ltr,
      ),
      Text(
        listNama[index % listNama.length],
        textDirection: TextDirection.ltr,
```



```
style: TextStyle(  
  fontSize: 45,  
  fontWeight: FontWeight.bold,  
  color: listWarna[index % listWarna.length]  
)  
)  
],  
);  
}
```

- Kita telah membuat sebuah state baru dengan nama StateTeksUtama.
- Kelas StateTeksUtama ini memiliki 3 buah variabel:
 - listNama
 - listWarna
 - dan index



- Kelas tersebut juga mengembalikan sebuah widget (pada fungsi build()) yang mana tampilannya masih sama dengan tampilan pada pertemuan sebelumnya.
- Bedanya, teks untuk nama sekarang diambil dari indeks yang aktif.
- Begitu juga untuk warna, juga diambil dari indeks yang aktif
- Selanjutnya, kita akan membuat sebuah Widget baru (stateful) berdasarkan state yang sudah kita buat (yaitu kelas StateTeksUtama).
- Langsung saja, kita beri nama widget yang baru ini dengan nama TeksUtama (sama seperti nama sebelumnya).
- Berikut ini kode programnya:



```
class TeksUtama extends StatefulWidget {  
  
    final state = StateTeksUtama();  
  
    @override  
    State<StatefulWidget> createState() => state;  
}
```

- Pada kode program di atas, kita membuat sebuah variabel dengan nama state yang merupakan *instance* dari kelas StateTeksUtama().
- Saat pertama kali dibangun, sebuah StatefulWidget akan membuat sebuah state dengan memanggil fungsi createState().
 - Kita langsung saja kembalikan variabel state yang telah kita inisiasi.



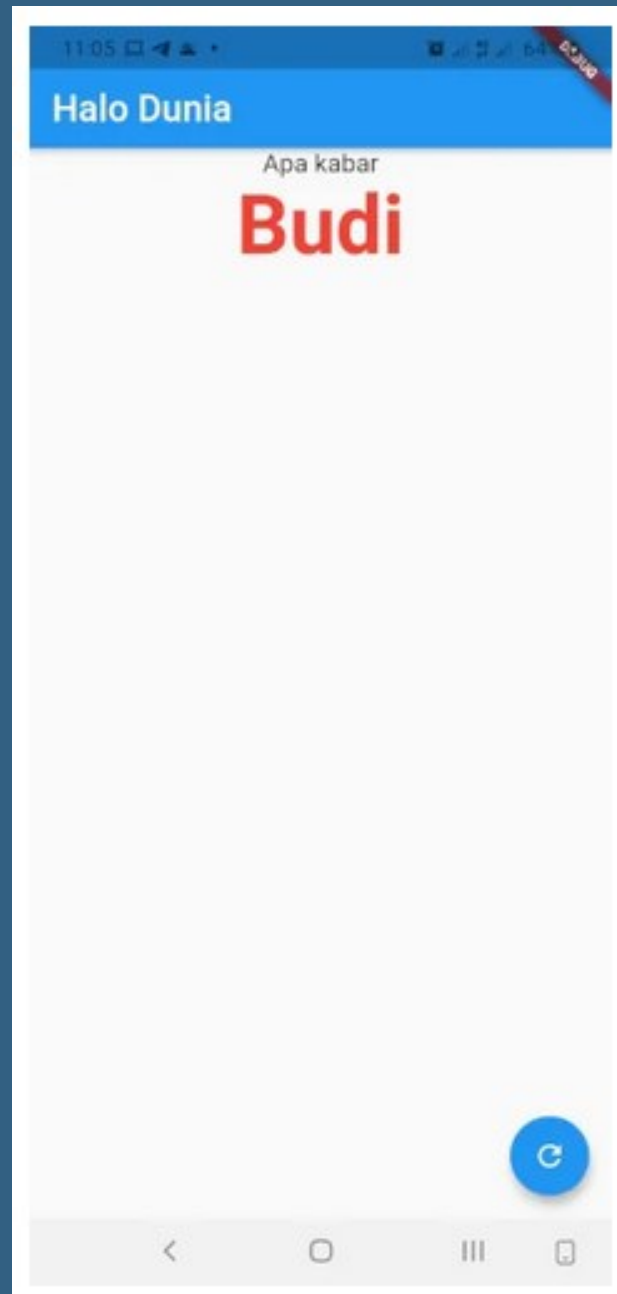
- Sampai di sini kode program kita sampai sekarang masih belum bisa dijalankan karena error.
- Oleh karena itu, mari kita perbaiki dengan mengubah body dari Scaffold pada fungsi main() menjadi seperti ini :
 - **Lihat baris yang ter-highlight**

```
void main() {  
    TeksUtama teksUtama = TeksUtama();  
  
    runApp(  
        MaterialApp(  
            title: 'Halo Dunia',  
            home: Scaffold(  
                ...  
                body: Center(  
                    child: teksUtama,  
                ),  
                ...  
            )  
        )  
    );  
}
```



- Pada kode di atas, kita membuat sebuah variabel teksUtama, yang merupakan instance dari widget TeksUtama.
- Kenapa kita inisiasi dulu seperti ini?
- Tidak wajib seperti ini, tapi bisa dicoba untuk memberikan variasi lain bahwasanya untuk menyusun UI pada flutter itu tidak kaku.
- Karena untuk menghasilkan output yang sama, kita bisa berkreatifitas sebebas kita.
- Sekarang, coba jalankan aplikasi kita, harusnya kita akan mendapatkan output sebagai berikut:





- Kita bisa bereksperimen dengan mengubah-ubah nilai index menjadi angka berapapun. Angka 1, 2, 10, 35, 95, 99.
- Lalu jalankan lagi aplikasi, anda akan menemukan bahwa teks nama dan warnanya akan berubah.
- Kita bisa juga menambahkan fungsi increment seperti berikut :
 - Mengubah state (variabel index) ketika tombol refresh diklik. Stepnya:
 1. Pada kelas StateTeksUtama, tambahkan fungsi incrementIndex() seperti di bawah:

```
void incrementIndex () {  
    setState(() {  
        index++;  
    });  
}
```



- Dan pada fungsi main(), atribut Scaffold.floatingActionButton, isi nilai onPressed menjadi teksUtama.state.incrementIndex.
- Perhatikan kode program yang ter-*highlight* berikut ini:

```
void main() {  
  TeksUtama teksUtama = TeksUtama();  
  
  runApp(  
    MaterialApp(  
      title: 'Halo Dunia',  
      home: Scaffold(  
        ...  
        body: Center(  
          child: teksUtama,  
        ),  
        floatingActionButton: FloatingActionButton(  
          tooltip: 'Refresh',  
          child: Icon(Icons.refresh),  
          onPressed: teksUtama.state.incrementIndex,  
        ),  
      ),  
    ),  
  );  
}
```



- Jalankan lagi aplikasi kita.
- Lalu coba tekan **tombol random** dan tulisan nama dan warnanya akan berubah-ubah

