

Pemrograman User Interface dengan Flutter:

Input Widgets

Fadjar Fathurrahman

1 Tujuan

- mengenal dan menggunakan beberapa input widgets yang sering ditemukan pada aplikasi Flutter

2 Input Widgets

Sebagian besar aplikasi didesain tidak hanya untuk menyampaikan atau menampilkan informasi, namun juga berinteraksi dengan pengguna. Salah satu interaksi dengan pengguna adalah melalui gestur (salah satunya adalah klik) dan input data seperti yang kita temui pada form HTML.

Flutter memiliki sangat banyak input widgets yang dapat digunakan. Pada praktikum ini kita akan mengenal beberapa input widgets yang sering digunakan:

- text field
- checkbox
- radio button
- slider
- dropdown

3 Text field

Text field pada Flutter direpresentasikan dengan widget `TextField`. Contoh penggunaan:

```
TextField(  
  controller: _controller,  
  decoration: InputDecoration(  
    labelText: 'Nama'  
    hintText: 'Ketik nama Anda di sini'  
  )  
);
```

Parameter controller memiliki banyak kegunaan, salah satunya adalah untuk menangani input dari pengguna. Contoh inisialisasi controller untuk `TextField`:

```
final _controller = TextEditingController();
```

Berikut ini adalah contoh program yang akan kita gunakan. Program ini merupakan potongan program, Anda harus membuat membuat program utama untuk memanggil program ini.

```

import 'package:flutter/material.dart';

class ExampleInput01 extends StatefulWidget {
  ExampleInput01({Key key}) : super(key: key);

  @override
  _ExampleInput01State createState() => _ExampleInput01State();
}

class _ExampleInput01State extends State<ExampleInput01> {

  final _inpNamaCtrl = TextEditingController();
  String _nama = '';

  void dispose() {
    _inpNamaCtrl.dispose();
    super.dispose();
  } // dispose

  void _handleSetName(String inpstr) {
    setState( () {
      _nama = inpstr;
    });
  }

  Widget build(BuildContext context) {

    // TextField yang kita gunakan dibungkus dengan Container agar lebih
    // mudah untuk mengatur padding
    final inpNama = Container(
      alignment: Alignment.center,
      padding: const EdgeInsets.all(6),
      child: TextField(
        controller: _inpNamaCtrl,
        decoration: InputDecoration(
          labelText: 'Nama',
          hintText: 'Ketik nama Anda di sini',
        ),
        onChanged: _handleSetName,
      )
    );

    final outNama = Container(
      alignment: Alignment.centerLeft,
      padding: EdgeInsets.all(10),
      child: Text(
        'Anda memasukkan nama: $_nama',
        style: TextStyle(fontSize: 15,)
      ),
    );

    return Scaffold(
      body: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [

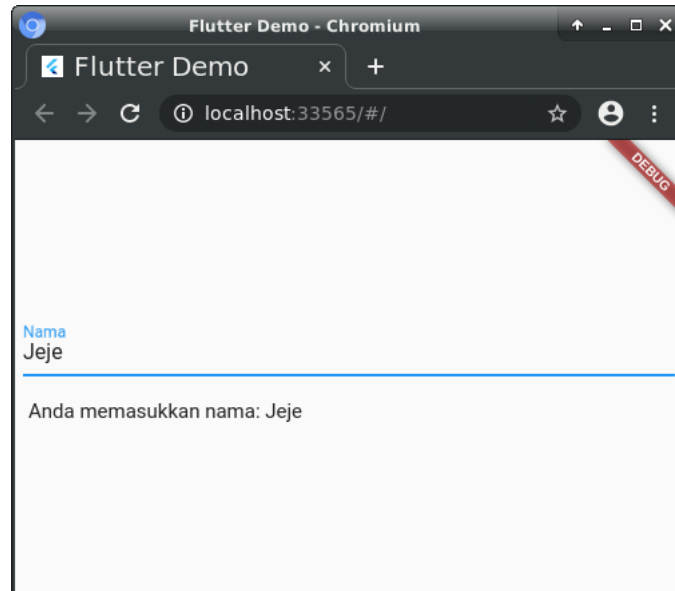
```

```

        inpNama,
        outNama,
    ],
)
);
}
}

```

Contoh tampilan program:



Untuk membatasi karakter apa saja yang dapat diketikkan pada suatu `TextField`, Anda dapat menggunakan property `inputFormatters` dari `TextField`. Beberapa kelas yang dapat digunakan (built-in) pada Flutter:

- `LengthLimitingTextInputFormatter`
- `WhitelistingTextInputFormatter`
- `BlacklistingTextInputFormatter`

Contoh:

```

TextField(
  inputFormatters: [
    // Hanya karakter angka, spasi, atau strip
    WhitelistingTextInputFormatter(RegExp('[0-9 -]')),
    // Panjang maksimal 16
    LengthLimitingTextInputFormatter(16),
  ]
)

```

4 Checkbox

Checkbox digunakan pada Flutter untuk mendapat input bernilai boolean. Contoh penggunaan (dengan label):

```

Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Checkbox(
      value: _setuju,

```

```

        onChanged: _handleSetuju,
      ),
      Text('Setuju'),
    ],
  );

```

Variabel `_setuju` adalah suatu variabel state pada suatu `StatefulWidget` dan `_handleSetuju` digunakan sebagai callback ketika nilai suatu `Checkbox` berubah.

```

bool _setuju = false;

// Callback untuk checkbox
void _handleSetuju(bool val) {
  setState( () {
    _setuju = val;
  });
}

```

5 Radio

Widget `Radio` biasa digunakan untuk menerima input yang berupa enumeration. Contoh:

```

enum NilaiKepuasan {
  baik,
  jelek,
  biasa,
}

NilaiKepuasan _nilaiKepuasan = NilaiKepuasan.baik;

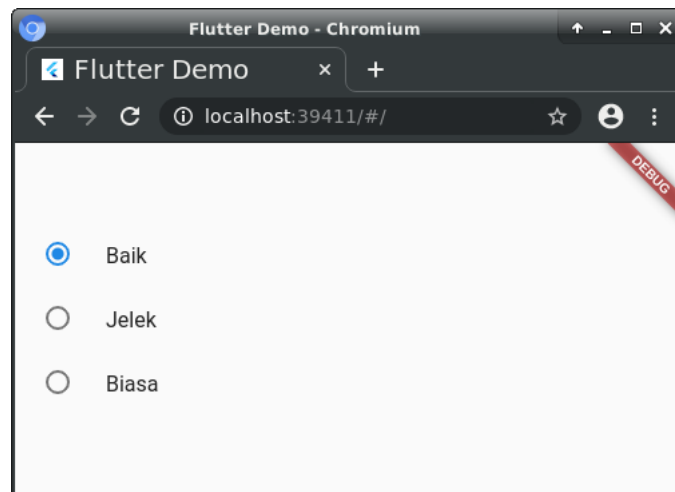
final radioPuasBaik = ListTile(
  title: Text('Baik'),
  leading: Radio(
    groupValue: _nilaiKepuasan,
    value: NilaiKepuasan.baik,
    onChanged: (NilaiKepuasan val) {
      setState( () {
        _nilaiKepuasan = val;
      });
    },
  ),
);

final radioPuasJelek = ListTile(
  title: Text('Jelek'),
  leading: Radio(
    groupValue: _nilaiKepuasan,
    value: NilaiKepuasan.jelek,
    onChanged: (NilaiKepuasan val) {
      setState( () {
        _nilaiKepuasan = val;
      });
    },
  ),
);

```

```
final radioPuasBiasa = ListTile(
  title: Text('Biasa'),
  leading: Radio(
    groupValue: _nilaiKepuasan,
    value: NilaiKepuasan.biasa,
    onChanged: (NilaiKepuasan val) {
      setState( () {
        _nilaiKepuasan = val;
      });
    },
  ),
);
```

Contoh tampilan:



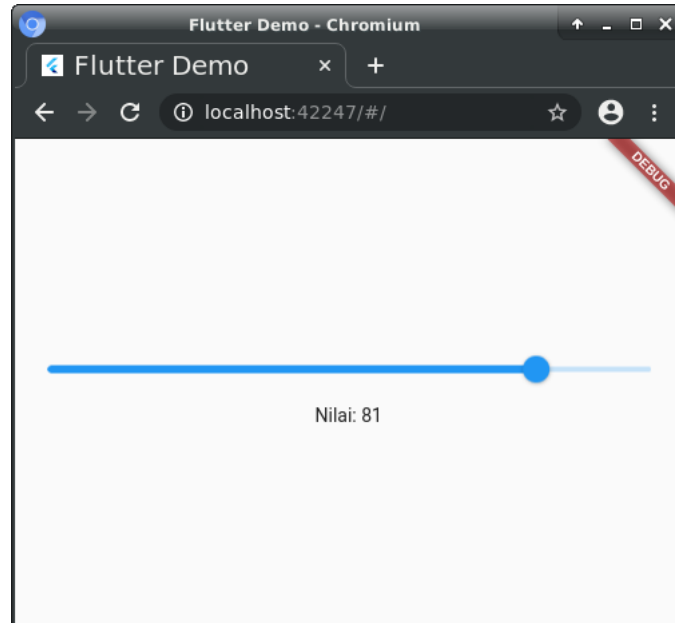
6 Slider

Slider biasa digunakan untuk menampilkan nilai numerik. Contoh:

```
double _nilaiSlider = 0;

final sliderNilai = Slider(
  label: _nilaiSlider.toString(),
  min: 0, max: 100,
  divisions: 100,
  value: _nilaiSlider,
  onChanged: (double val) {
    setState( () {
      _nilaiSlider = val;
    });
  }
);
```

Contoh tampilan:

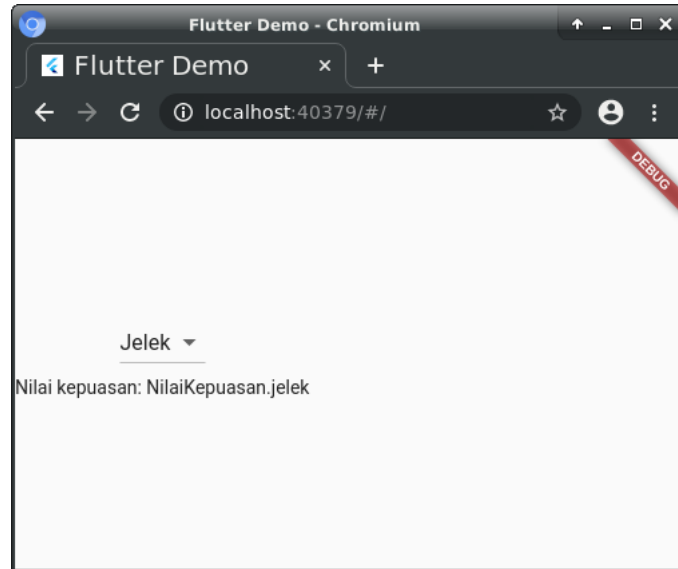


7 Dropdown

Kegunaan widget ini mirip dengan Radio. Contoh penggunaan:

```
final db1 = DropdownButton<NilaiKepuasan>(  
  value: _nilaiKepuasan,  
  items: const <DropdownMenuItem<NilaiKepuasan>>[  
    DropdownMenuItem<NilaiKepuasan>(  
      child: Text('Baik'),  
      value: NilaiKepuasan.baik,  
    ),  
    DropdownMenuItem<NilaiKepuasan>(  
      child: Text('Jelek'),  
      value: NilaiKepuasan.jelek,  
    ),  
    DropdownMenuItem<NilaiKepuasan>(  
      child: Text('Biasa'),  
      value: NilaiKepuasan.biasa,  
    ),  
  ],  
  onChanged: (NilaiKepuasan val) {  
    setState( () {  
      _nilaiKepuasan = val;  
    });  
  },  
);
```

Contoh tampilan:



8 Tugas

Lengkapi potongan program di atas sehingga dapat menjadi program Flutter yang dapat dijalankan.