

Pemrograman User Interface dengan Flutter:

Form

Fadjar Fathurrahman

1 Tujuan

- mengimplementasikan form dan validasi data sederhana pada aplikasi Flutter

2 Form

Pada minggu lalu, kita telah mengenal beberapa widget yang dapat digunakan untuk menerima input dari user. Widget input ini dapat digabungkan ke dalam suatu widget Form. Widget Form biasa digunakan jika kita ingin mendapatkan input data yang saling terkait. Misalnya kita ingin membuat suatu fitur untuk menambahkan user baru pada aplikasi. Biasanya kita memerlukan input nama pengguna (nama asli), username, password, email, tempat dan tanggal lahir, dan sebagainya. Pada kasus tersebut Form dapat digunakan untuk membungkus semua widget input yang diperlukan beserta data yang ada menjadi suatu kesatuan. Pada Flutter, Form dapat melakukan hal ini dengan menggunakan suatu key, yang dapat diinisialisasi menggunakan sintaks berikut:

```
GlobalKey<FormState> _key = GlobalKey<FormState>();
```

Variabel `_key` ini kemudian dapat digunakan pada Form sebagai berikut:

```
Widget build(BuildContext context) {  
  return Form(  
    key: _key,  
    child: ... //  
  );  
}
```

Pada dasarnya, setiap input widget dapat dibungkus dalam suatu Form. Flutter menyediakan widget `FormField` yang dapat membungkus sembarang widget sehingga dapat digunakan pada suatu Form. Salah satu keuntungannya adalah `FormField` menyediakan beberapa metode yang berguna dalam memproses input dari Form:

- `onSaved`
- `validator`

Pada Flutter, ada satu widget yang sering digunakan pada Form yang sudah dibungkus dengan `FormField`, yaitu `TextFormField`. Widget ini tidak lain adalah `TextField` yang sudah dibungkus dengan `FormField` sehingga dapat langsung digunakan pada Form.

Contoh:

```
// Gunakan widget ini sebagai home pada MaterialApp
class ContohFormPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Contoh Form')),
      body: ContohForm(),
    );
  }
}

class ContohForm extends StatefulWidget {
  @override
  ContohFormState createState() {
    return ContohFormState();
  }
}

class ContohFormState extends State<ContohForm> {

  final _formKey = GlobalKey<FormState>();
  String _nama = '';
  String _nim = '';

  @override
  Widget build(BuildContext context) {

    final _buttonSubmit = Padding(
      padding: const EdgeInsets.symmetric(vertical: 16.0),
      child: ElevatedButton(
        child: Text('Daftar'),
        onPressed: () {
          Scaffold.of(context).showSnackBar(
            SnackBar(
              content: Text('Terimakasih data sedang diproses')
            )
          );
          // Tulis ke terminal
          print('Nama = $_nama');
          print('NIM = $_nim');
        },
      ),
    );

    final _inputNama = TextFormField(
      decoration: InputDecoration(
        labelText: 'Nama',
        hintText: 'Ketik nama Anda di sini',
      ),
      onChanged: ..., // lengkapi
    );

    final _inputNim = TextFormField(
      decoration: InputDecoration(
        labelText: 'NIM',

```

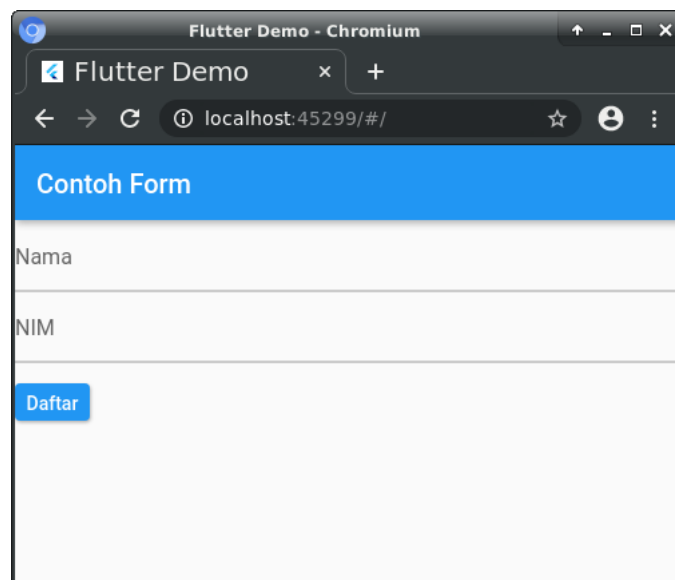
```

        hintText: 'Ketik nama Anda di sini',
      ),
      onChanged: ..., // lengkapi
    );

    return Form(
      key: _formKey,
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: <Widget>[
          _inputNama,
          _inputNim,
          _buttonSubmit,
        ],
      ),
    );
  }
}

```

Contoh tampilan:



Silakan lengkapi kode di atas sehingga ketika button Daftar ditekan, program akan menampilkan nama dan NIM pada terminal atau konsol.

3 Validasi

Pada beberapa form, kita seringkali perlu melakukan validasi dari input yang diterima oleh user. Misalkan kita ingin agar input nama tidak boleh kosong ketika user mengklik tombol Daftar. Hal ini dapat dilakukan dengan menggunakan properti validator pada `_inputNama`:

```

final _inputNama = TextFormField(
  decoration: InputDecoration(
    labelText: 'Nama',
    hintText: 'Ketik nama Anda di sini',
  ),
  onChanged: ..., // silakan isi
  validator: (value) {

```

```

    if (value.isEmpty) {
      return 'Bagian ini harus diisi';
    }
    return null;
  }
};

```

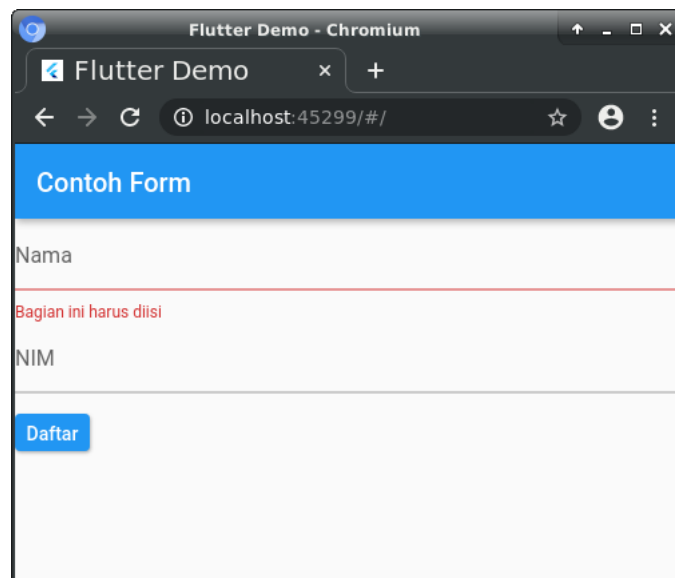
Untuk _buttonSubmit:

```

final _buttonSubmit = Padding(
  padding: const EdgeInsets.symmetric(vertical: 16.0),
  child: ElevatedButton(
    child: Text('Daftar'),
    onPressed: () {
      if(_formKey.currentState.validate()) {
        Scaffold.of(context).showSnackBar(
          SnackBar(
            content: Text('Terimakasih data sedang diproses')
          )
        );
        print('nama = $_nama');
        print('NIM = $_nim');
      }
    },
  ),
);

```

Tampilan aplikasi:



4 Tugas

- Coba lakukan validasi untuk `_inputNim` sehingga input ini harus diisi.
- Buat suatu data mahasiswa beserta NIM, misalkan untuk 10 mahasiswa. Lakukan validasi untuk input nama dan NIM yang diberikan sedemikian rupa sehingga hanya nama dan NIM yang ada pada data yang dapat diterima. Periksa juga sehingga pasangan nama dan NIM sesuai.