

Modul 14

Komponen dalam mendesain Aplikasi

DatePicker dan TimePicker

Module Overview

Mengenal dan mengimplementasikan widget DatePicker dan TimePicker untuk memudahkan pengguna dalam memilih waktu dengan widget yang telah disediakan oleh material UI.

Module Objectives

Setelah mempelajari dan mempraktikkan modul ini, mahasiswa diharapkan dapat:

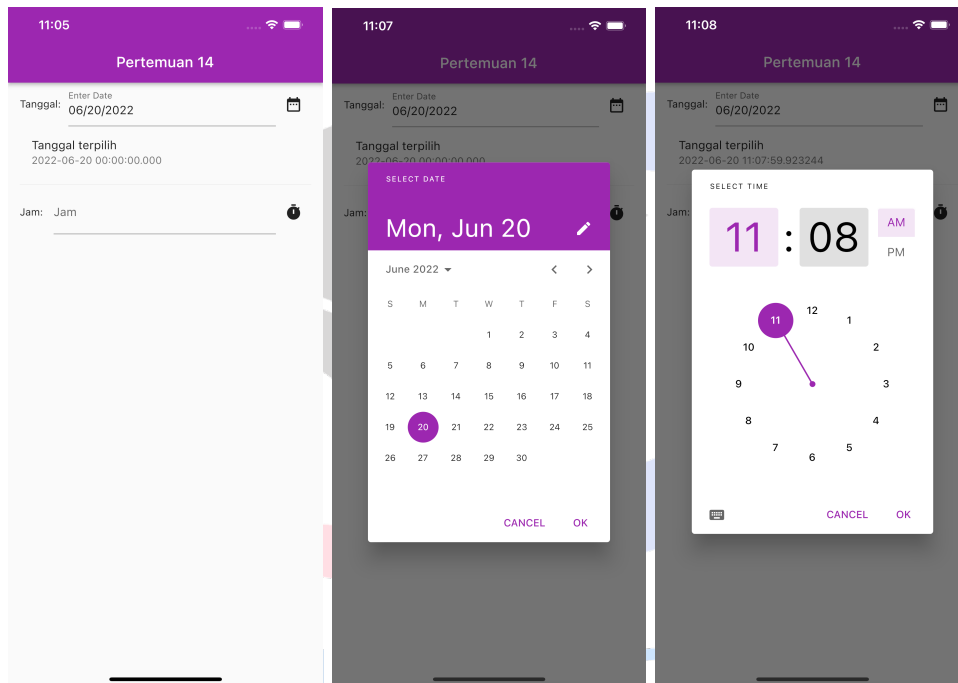
- Mengetahui kegunaan dari widget DatePicker dan TimePicker.
- Mampu menerapkan widget DatePicker dan TimePicker pada aplikasi flutter.

Baners, Dialog dan Snackbars

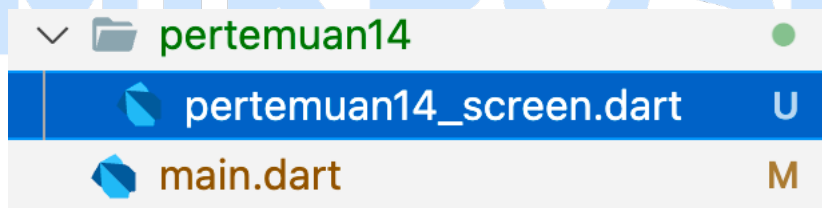
Menerapkan DatePicker dan TimePicker

Keterangan:

Aplikasi ini menerapkan DatePicker dan TimePicker menggunakan widget **showDatePicker()** dan **showTimePicker()** untuk memudahkan pengguna dalam memilih waktu.



1. Berikut adalah struktur folder untuk pertemuan 14.



Keterangan:

- pertemuan14_screen.dart merupakan file untuk screen UI flutter (root home)

2. Buatlah screen baru dengan nama file **pertemuan14_screen.dart** pada folder pertemuan14. Import **material**, Gunakan **StatefulWidget**, namai class **statefulWidget** dengan **Pertemuan14Screen** dan gunakan **Scaffold** untuk memulai mendesain komponen.

```
1 import 'package:flutter/material.dart';
2
3 class Pertemuan14Screen extends StatefulWidget {
4   const Pertemuan14Screen({Key? key}) : super(key: key);
5
6   @override
7   State<Pertemuan14Screen> createState() => _Pertemuan14ScreenState();
8 }
9
10 class _Pertemuan14ScreenState extends State<Pertemuan14Screen> {
11   @override
12   Widget build(BuildContext context) {
13     return Scaffold(
14       appBar: AppBar(title: const Text('Pertemuan 14')),
15       body: const Text('Pertemuan14'));
16   }
17 }
```

Keterangan:

- Update **main.dart** menjadi "home: Pertemuan14Screen()"
- Pada terminal root folder ketik "**flutter run**" untuk menjalankan aplikasi.

3. **Menggunakan showDatePicker().** Pada body Pertemuan13Screen, tambahkan widget **Padding** dan **Column**, kemudian children dari column sebagai berikut:

```

1  body: Padding(
2    padding: const EdgeInsets.symmetric(horizontal: 16.0),
3    child: Column(
4      children: [
5        //DatePicker
6        Row(
7          children: [
8            const Text('Tanggal:'),
9            const SizedBox(width: 10),
10           Expanded(
11             child: InputDatePickerFormField(
12               initialDate: _date,
13               firstDate: DateTime(1990),
14               lastDate: DateTime(2250),
15               onPressed: (date) {
16                 setState(() {
17                   _date = date;
18                   print(_date);
19                 });
20               },
21             ),
22           ),
23           IconButton(
24             onPressed: () async {
25               var res = await showDatePicker(
26                 context: context,
27                 initialDate: _date,
28                 firstDate: DateTime(2000),
29                 lastDate: DateTime(2500));
30
31               if (res != null) {
32                 setState(() {
33                   _date = res;
34                 });
35               }
36             },
37             icon: const Icon(Icons.date_range))
38           ],
39         ),
40         ListTile(
41           title: const Text('Tanggal terpilih'),
42           subtitle: Text(_date.toString()),
43         ),
44         const Divider(),
45       ],
46     ),
47   );

```

Tambahkan lokal variabel **_date** yang berisi waktu saat ini dengan memanggil **DateTime.now()**. Letakkan ini **sebelum** Widget build.



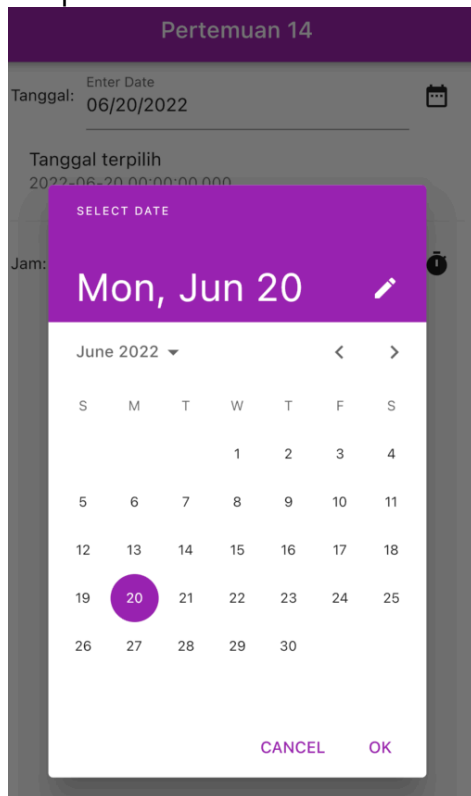
```
1 class _Pertemuan14ScreenState extends State<Pertemuan14Screen> {  
2   DateTime _date = DateTime.now();  
3  
4   @override  
5   Widget build(BuildContext context) {
```

Keterangan:

- **Baris 11-21, InputDatePickerFormField** adalah sebuah widget form field yang dapat menerima inputan tanggal. Umumnya widget ini adalah opsi jika pengguna ingin mengetik / isi waktu sendiri sesuai format waktu yang ditentukan.
- **Baris 12, initialDate** harus diisi dan berada direntang waktu **firstDate** dan **lastDate**. Pada source code di atas initialDate di dapat dari waktu saat ini yang telah diisialisasi melalui **DateTime.now()** pada tahapan sebelumnya.
- **Baris 15, onSubmit** adalah sebuah property untuk menerima inputan waktu oleh pengguna. Pada fungsi ini maka di setlah **_date** dengan **date**.
- **Baris 23-39**, adalah sebuah **IconButton()** yang jika diklik maka akan menampilkan sebuah **showDatePicker()**. sama seperti widget sebelumnya, maka pada **showDatePicker** harus diisi **initialDate**, **firstDate** dan **lastDate**. Ini akan membatasi tanggal yang bisa dipilih oleh pengguna.
- **showDatePicker()** merupakan dialog yang mengembalikan nilai (**future**) atau sebuah fungsi **Future<DateTime>** artinya mengembalikan nilai **DateTime** jika pengguna telah memilih waktu atau mengembalikan nilai **null** jika tidak ada memilih waktu. Jadi pastikan menerapkan **async-await** pada fungsi ini.

Hasil:

Simpan dan reload flutter untuk melihat hasilnya.



UNIVERSITAS
MIKROSKIL

4. **Menggunakan showTimePicker.** Tambahkan widget berikut sebagai children dari **column!** letakkan setelah widget sebelumnya. Seperti tampilan berikut ini:



```

1  //Time Picker
2  Row(
3    children: [
4      const Text('Jam: '),
5      const SizedBox(width: 10),
6      Expanded(
7        child: TextField(
8          enabled: false,
9          controller: _time,
10         decoration: const InputDecoration(labelText: 'Jam'),
11       )),
12     IconButton(
13       onPressed: () async {
14         var res = await showTimePicker(
15           context: context,
16           initialTime: TimeOfDay.now(),
17         );
18
19         if (res != null) {
20           setState(() {
21             _time =
22               TextEditingController(text: res.format(context));
23           });
24         }
25       },
26       icon: const Icon(Icons.timer))
27   ],
28 )

```

Tambahkan TextEditingController `_time` di dalam class, seperti gambar berikut.



```

1  class _Pertemuan14ScreenState extends State<Pertemuan14Screen> {
2    DateTime _date = DateTime.now();
3    TextEditingController? _time;
4
5    @override
6    Widget build(BuildContext context) {

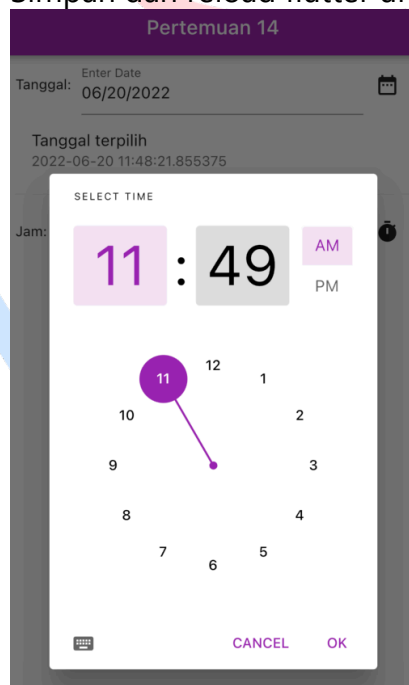
```

Keterangan:

- **Baris 14 - 17**, sebuah fungsi untuk memanggil dialog timePicker menggunakan fungsi `showTimePicker` yang akan return sebuah `Future<TimeOfDay>`. Oleh karena itu gunakan `async - await`.
- **Baris 7 - 11**, adalah sebuah `textField` untuk menerima nilai waktu yang dipilih oleh pengguna.
- **Baris 16**, `TimeOfDay.now()` akan menghasilkan waktu saat ini.
- **Baris 19 - 24**, Jika waktu yang dipilih tidak null maka perlu di set waktu pada variabel `_time` dengan format **`res.format(context)`**, tujuannya agar `TimeOfDay` memiliki format yang mudah dibaca pengguna, contoh: 11.00 AM, dari pada `TimeOfDay(11:00)`.

Hasil

Simpan dan reload flutter untuk melihat perubahan.



Latihan Praktek

Menggunakan **showDateRangePicker()**, tambahkan sebuah `iconButton` yang akan menampilkan dialog Date Range Picker. Kemudian tampilkan tanggal-tanggal terpilih sebagai list.

