

| | |
|----------------------|--|
| Universitas | : Universitas Logistik dan Bisnis Internasional (ULBI) |
| Program Studi | : D4 Teknik Informatika |
| Semester | : 5 (Lima) |
| Mata Kuliah | : Pemrograman IV (Pemrograman Mobile) |
| Topik | : Praktikum 7 Membuat Input Widget Lanjutan |

ADVANCED INPUT WIDGET

Masih sama seperti di Pertemuan ke-6 kali ini kita akan membahas bagaimana cara membuat:

- Date Picker
- Color Picker
- File Picker

Sebelum masuk ke pembuatan widget, gunakan project di minggu ke-6. Selanjutnya yang harus kalian buat adalah buat file baru bernama **advanced_form.dart** kemudian buat sebuah StatefulWidget dengan nama **AdvancedForm** berikut

```

1  import 'package:flutter/material.dart';
2
3  class AdvancedForm extends StatefulWidget {
4    const AdvancedForm({super.key});
5
6    @override
7    State<AdvancedForm> createState() => _AdvancedFormState();
8  }
9
10 class _AdvancedFormState extends State<AdvancedForm> {
11   @override
12   Widget build(BuildContext context) {
13     return const Placeholder();
14   }
15 }

```

Kemudian ubah kode pada **widget build** seperti berikut

```

Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Interactive Widget'),
      centerTitle: true,
    ),
    body: Container(
      padding: const EdgeInsets.all(16),
      child: ListView(
        children: [
          // buildDatePicker,
          // buildColorPicker,
          // buildFilePicker,
        ],
      ),
    ),
  );
}

```

Nanti ketiga widget yang akan dibuat (Date Picker, Color Picker, File Picker) tinggal kalian panggil pada struktur yang berwarna merah pada code di atas.

A. Date Picker

Widget **DatePicker** pada Flutter adalah sebuah widget yang digunakan untuk menampilkan antarmuka pemilihan tanggal (date picker) kepada pengguna. Widget ini memungkinkan pengguna untuk memilih tanggal dari sebuah kalender dan biasanya digunakan dalam aplikasi yang memerlukan pengaturan tanggal, seperti aplikasi perencanaan atau pengelolaan jadwal. Dengan 'DatePicker', pengguna dapat memilih tanggal dengan mudah, dan nilai tanggal yang dipilih dapat digunakan dalam kode aplikasi untuk berbagai tujuan, seperti penjadwalan, penghitungan, atau penyimpanan data terkait tanggal.

- **Menambahkan packages intl**

Cara yang pertama adalah menambah packages **intl** di **pubspec.yaml** ([INTL PACKAGE](#))

```
dependencies:  
  flutter:  
    sdk: flutter  
  intl: ^0.20.2
```

Ingat posisi intl harus sejajar dengan flutter.

- **Mempersiapkan variabel**

Selanjutnya adalah kita inisialisasi dan deklarasikan 2 variabel berikut

```
class _AdvancedFormState extends State<AdvancedForm> {  
  DateTime _dueDate = DateTime.now();  
  final currentDate = DateTime.now();  
}
```

- **Membangun UI**

Selanjutnya adalah membuat user interface dari Date Picker, disini kita coba buatlah sebuah method dengan nama **buildDatePicker** pada class **_AdvancedFormState** seperti berikut

```
Widget buildDatePicker(BuildContext context) {
  return Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
          const Text('Date'),
          TextButton(
            child: const Text('Select'),
            onPressed: () {},
          ), // TextButton
        ],
      ), // Row
      Text(DateFormat('dd-MM-yyyy').format(_dueDate))
    ],
  ); // Column
}
```

Kemudian panggil method buildDatePicker yang sudah dibuat pada **Widget build** di atas

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Interactive Widget'),
      centerTitle: true,
    ), // AppBar
    body: Container(
      padding: const EdgeInsets.all(16),
      child: ListView(
        children: [
          buildDatePicker(context),
        ],
      ), // ListView
    ), // Container
  ); // Scaffold
}
```

Kemudian buka file **bottom_navbar.dart** kemudian panggil class AdvancedForm yang sudah dibuat

```
14 class _DynamicBottomNavbarState extends State<DynamicBottomNavbar> {
15   int _currentPageIndex = 0;
16
17   final List<Widget> _pages = <Widget>[
18     const MyInput(),
19     const MyFormValidation(),
20     const MyInputForm(),
21     const AdvancedForm(),
22   ]; // <Widget>[]
```

Kemudian pada Widget build bottom navbar tambah kode berikut

```

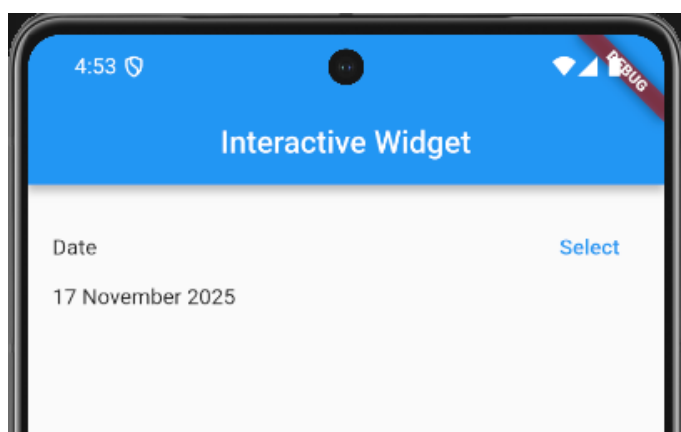
@override
Widget build(BuildContext context) {
  return Scaffold(
    body: _pages[_currentPageIndex],
    bottomNavigationBar: BottomNavigationBar(
      type: BottomNavigationBarType.fixed,
      currentIndex: _currentPageIndex,
      onTap: onTabTapped,
      items: const [
        BottomNavigationBarItem(
          icon: Icon(Icons.task_alt_outlined),
          label: 'Latihan',
        ), // BottomNavigationBarItem
        BottomNavigationBarItem(
          icon: Icon(Icons.input_outlined),
          label: 'Form Validation',
        ), // BottomNavigationBarItem
        BottomNavigationBarItem(
          icon: Icon(Icons.dynamic_form_rounded),
          label: 'Form Input',
        ), // BottomNavigationBarItem
        BottomNavigationBarItem(
          icon: Icon(Icons.phone_iphone_rounded),
          label: 'Advanced Form',
        ), // BottomNavigationBarItem
      ],
    ),
  );
}

```

Pada kode **BottomNavigationBarType.fixed** digunakan untuk:

- Mempertahankan semua item navigasi ditempatkan sama secara permanen di bagian bawah.
- Cocok jika kalian memiliki lebih dari tiga item

Coba jalankan aplikasi seharusnya sudah tampil seperti gambar berikut



Namun fungsional dari widgetnya belum bisa, karena kita belum membuatnya.

- **Menambahkan fungsi showDatePicker**

Selanjutnya adalah kita menambahkan fungsi ShowDatePicker di dalam onPressed

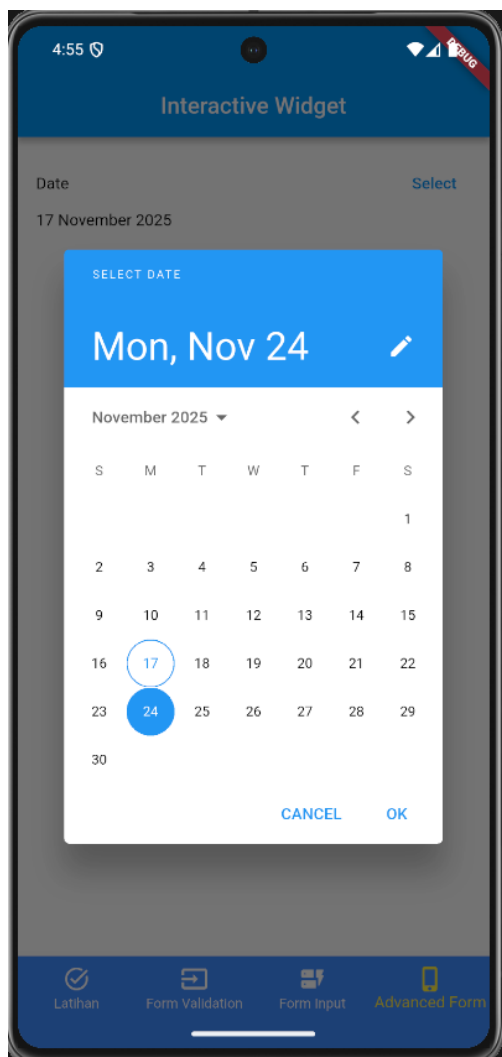
```
onPressed: () async {
  final selectDate = await showDatePicker(
    context: context,
    initialDate: currentDate,
    firstDate: DateTime(1990),
    lastDate: DateTime(currentDate.year + 5),
  );
}
```

- **Memanggil fungsi setState**

Kemudian masih di onPressed, kita buat fungsi setState

```
setState(() {
  if (selectDate != null) {
    _dueDate = selectDate;
  }
});
```

- **Hasil Akhir**



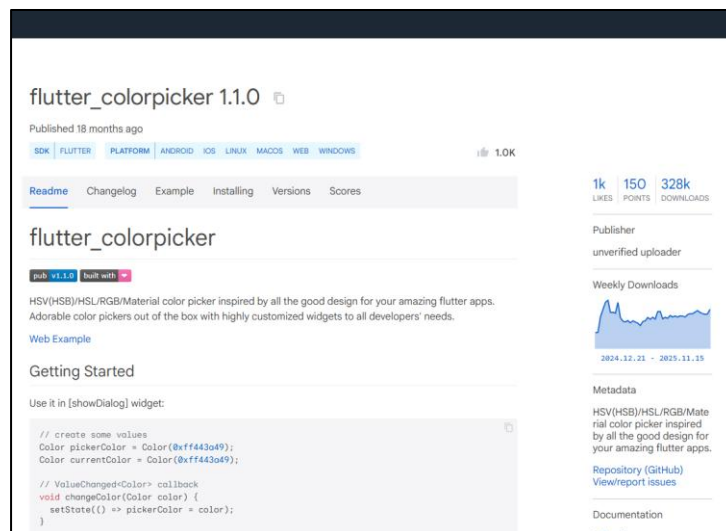
Bisa dilihat sekarang DatePicker sudah berjalan sempurna. Saat select button ditekan akan muncul dialog date picker, tanggal yang ada di UI akan berubah sesuai dengan tanggal yang dipilih oleh user.

B. Color Picker

Widget **ColorPicker** pada Flutter adalah sebuah komponen yang memungkinkan pengguna untuk memilih warna dengan mudah. Widget ini menyediakan berbagai cara bagi pengguna untuk memilih warna, seperti pemilihan warna langsung, pemilihan warna melalui roda warna, atau bahkan melalui kode warna. Hal ini sangat berguna dalam aplikasi yang memerlukan personalisasi warna, seperti aplikasi cat atau desain, dan memudahkan pengguna untuk menyesuaikan tampilan sesuai preferensi mereka. Widget color picker tersedia dalam berbagai paket dan dapat disesuaikan sesuai kebutuhan pengembang aplikasi.

- **Menggunakan packages flutter_colorpicker**

Pada praktikum pembuatan color picker kita akan menggunakan packages flutter_colorpicker. Bisa akses [disini](#)



- **Menambahkan packages flutter_collorpicker**

Cara selanjutnya adalah menambah packages flutter_collorpicker di pubspec.yaml

```
dependencies:  
  flutter:  
    sdk: flutter  
  intl: ^0.19.0  
  flutter_colorpicker: ^1.1.0
```

Ingat posisi flutter_colorpicker harus sejajar dengan intl.

*Note jika kalian sudah mengubah file pubspec.yaml tapi project flutter kalian masih error coba lakukan stop proses running aplikasi kemudian coba running kembali aplikasinya

- **Menambahkan variabel**

Selanjutnya adalah kita inisialisasi dan deklarasikan 1 variabel (**_currentColor**) berikut pada class **_AdvancedFormState**

```
class _AdvancedFormState extends State<AdvancedForm> {
  DateTime _dueDate = DateTime.now();
  final currentDate = DateTime.now();
  Color _currentColor = Colors.orange;
}
```

- **Membangun UI**

Selanjutnya adalah membuat user interface dari Color Picker, disini kita coba buatlah sebuah method dengan nama **buildColorPicker** seperti berikut

```
69 Widget buildColorPicker(BuildContext context) {
70   return Column(
71     crossAxisAlignment: CrossAxisAlignment.start,
72     children: [
73       const Text('Color'),
74       const SizedBox(height: 10),
75       Container(
76         height: 100,
77         width: double.infinity,
78         color: _currentColor,
79       ), // Container
80       const SizedBox(height: 10),
81       Center(
82         child: ElevatedButton(
83           style: ElevatedButton.styleFrom(
84             backgroundColor: _currentColor,
85           ),
86           onPressed: () {},
87           child: const Text('Pick Color'),
88         ), // ElevatedButton
89       ), // Center
90     ],
91   ); // Column
92 }
```

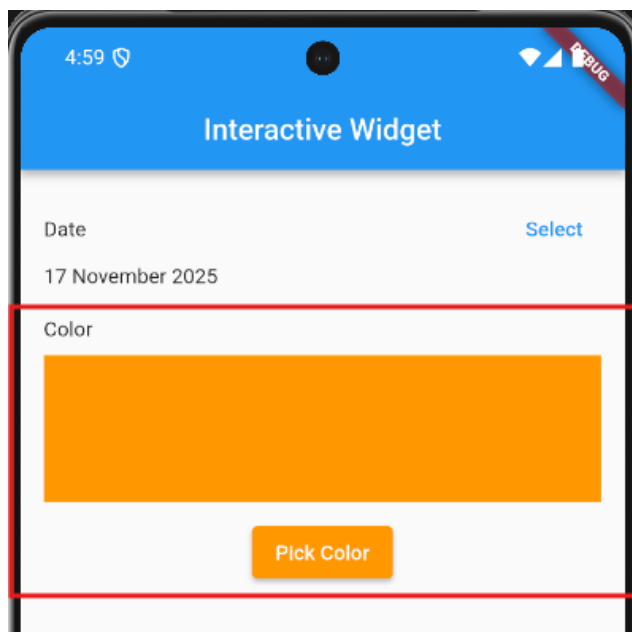
Kemudian panggil method buildDatePicker yang sudah dibuat, tapi sebelum itu tambahkan juga widget SizedBox agar jarak DatePicker dan ColorPicker tidak terlalu berdekatan.

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Interactive Widget'),
      centerTitle: true,
    ), // AppBar
    body: Container(
      padding: const EdgeInsets.all(16),
      child: ListView(
        children: [
          buildDatePicker(context),
          const SizedBox(height: 20),
          buildColorPicker(context),
        ],
      ), // ListView
    ), // Container
  ); // Scaffold
}

```

Coba jalankan aplikasi seharusnya sudah tampil seperti gambar berikut



Namun fungsional dari widgetnya belum bisa, karena kita belum membuatnya.

- **Menambahkan fungsi showDialog**

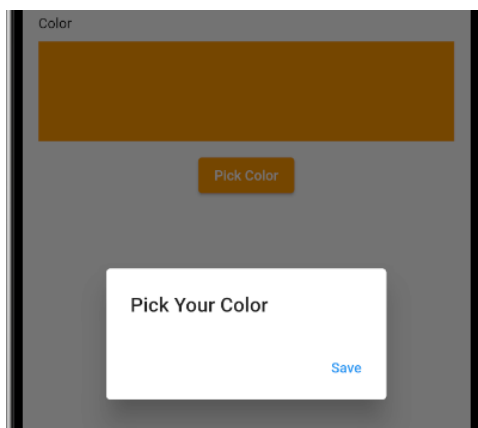
Selanjutnya disini kita akan menambahkan showDialog di dalam onPressed widget buildColorPicker dengan return widget AlertDialog


```

onPressed: () {
  showDialog(
    context: context,
    builder: (context) {
      return AlertDialog(
        title: const Text('Pick Your Color'),
        actions: [
          TextButton(
            onPressed: () {
              return Navigator.of(context).pop();
            },
            child: const Text('Save'),
          ), // TextButton
        ],
      ); // AlertDialog
    });
},

```

Saat dijalankan masih hanya seperti ini saja yaitu menampilkan AlertDialog



- **Import packages**

Lakukan import package flutter_colorpicker

```
import 'package:flutter_colorpicker/flutter_colorpicker.dart';
```

- **Membuat kode**

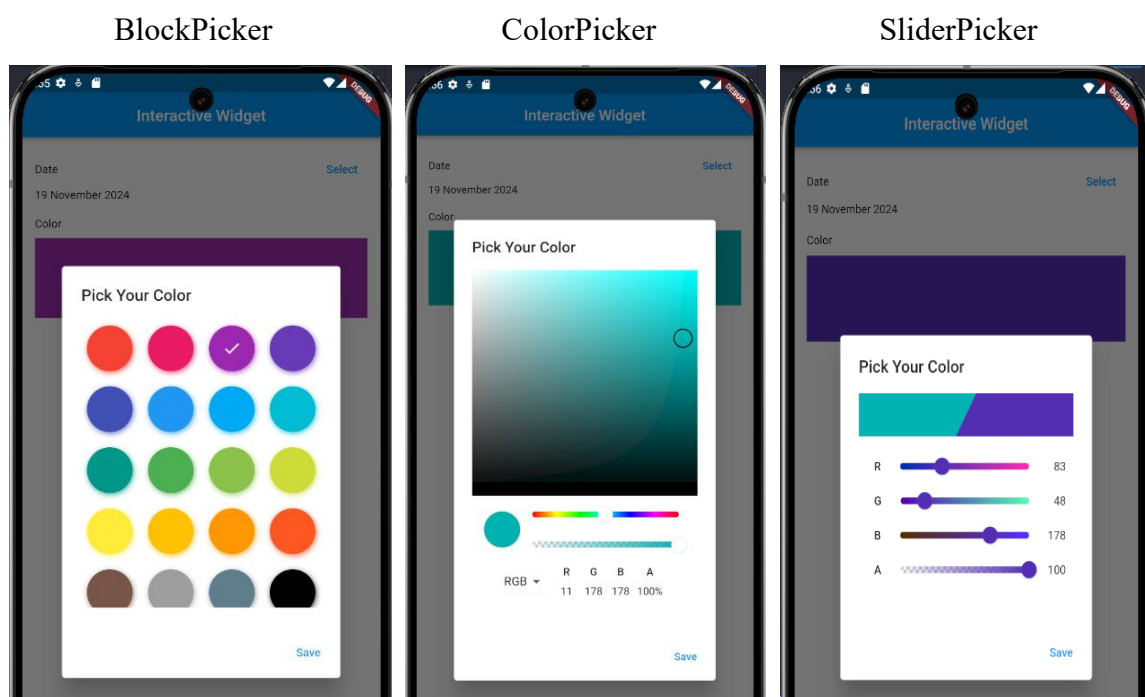
Selanjutnya masih pada onPressed kita akan membuat kode untuk penggunaan packages flutter_colorpicker, tempatkan kode berikut pada Widget AlertDialog setelah parameter title seperti berikut

```
onPressed: () {
  showDialog(
    context: context,
    builder: (context) {
      return AlertDialog(
        title: const Text('Pick Your Color'),
        content: Column(
          mainAxisAlignment: MainAxisAlignment.min,
          children: [
            BlockPicker(
              pickerColor: _currentColor,
              onColorChanged: (color) {
                setState(() {
                  _currentColor = color;
                });
              }, // BlockPicker
            ),
          ], // Column
        ),
        actions: [
```

Contoh di atas kita menggunakan **BlockPicker**, namun kalian juga bisa mencoba selain **BlockPicker**, kalian juga bisa menggantinya dengan **ColorPicker** dan **SliderPicker**.

- **Hasil akhir**

Packages flutter_colorpicker memiliki custom widget yang dapat digunakan, seperti **BlockPicker**, **ColorPicker** dan **SliderPicker**

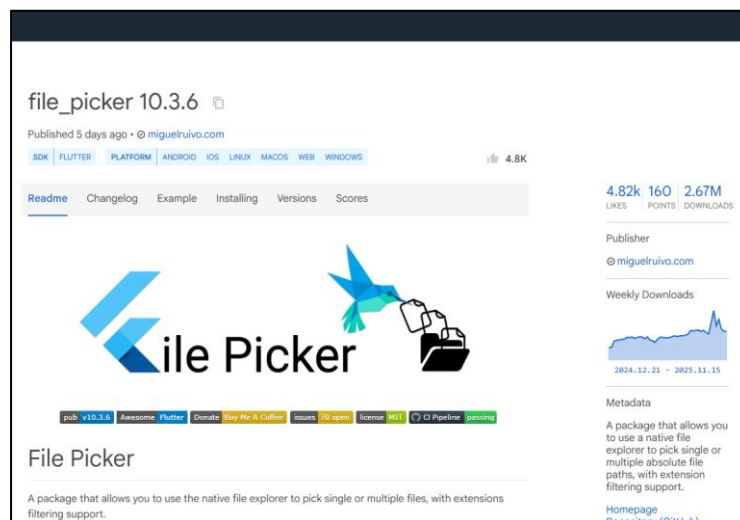


C. File Picker

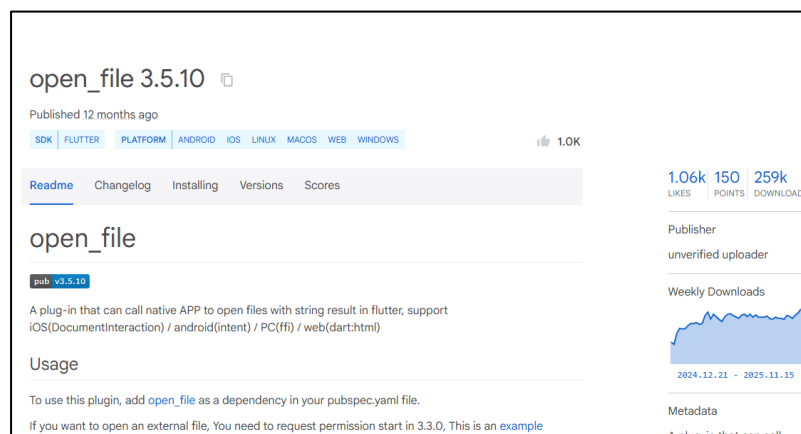
Widget **FilePicker** pada Flutter adalah alat yang memungkinkan pengguna untuk memilih file dari sistem file perangkat mereka dengan mudah. Widget ini memfasilitasi pengambilan berbagai jenis file, seperti gambar, dokumen, audio, dan lainnya. FilePicker memungkinkan pengguna untuk menjelajah dan memilih file yang diperlukan, dan pengembang dapat menggunakannya untuk mengakses dan mengelola file yang diperlukan dalam aplikasi Flutter mereka. Widget ini sangat berguna dalam berbagai jenis aplikasi, seperti aplikasi pengeditan dokumen, pengunggahan gambar, dan berbagai aplikasi lain yang memerlukan akses file dari perangkat pengguna.

- **Menggunakan packages file_picker dan open_file**

Packages file_picker. Bisa akses [disini](#)



Packages open_file. Bisa akses [disini](#)



- **Menambahkan packages file_picker dan open_file di pubspec.yaml**

Cara selanjutnya adalah menambah packages file_picker dan open_file di pubspec.yaml

```
dependencies:
  flutter:
    sdk: flutter
  intl: ^0.20.2
  flutter_colorpicker: ^1.1.0
  file_picker: ^10.3.6
  open_file: ^3.5.10
```

Jalankan perintah **flutter pub get** pada terminal vscode

- **Import packages**

Lakukan import package `file_picker` dan `open_file`

```
import 'package:file_picker/file_picker.dart';
import 'package:open_file/open_file.dart';
```

- **Menambahkan variabel**

Selanjutnya adalah kita inisialisasi 1 variabel berikut untuk menyimpan file

```
class _AdvancedFormState extends State<AdvancedForm> {
  DateTime _dueDate = DateTime.now();
  final currentDate = DateTime.now();
  Color _currentColor = Colors.orange;
  String? _dataFile;
```

- **Membangun UI**

Selanjutnya adalah membuat user interface dari File Picker, disini kita coba buatlah sebuah method dengan nama **buildFilePicker** seperti berikut

```
124 Widget buildFilePicker(BuildContext context) {
125   return Column(
126     crossAxisAlignment: CrossAxisAlignment.start,
127     children: [
128       const Text('Pick File'),
129       const SizedBox(height: 10),
130       Center(
131         child: ElevatedButton(
132           onPressed: () {},
133           child: const Text('Pick and Open File'),
134         ), // ElevatedButton
135       ), // Center
136       if (_dataFile != null) Text('File Name: $_dataFile'),
137     ],
138   ); // Column
139 }
```

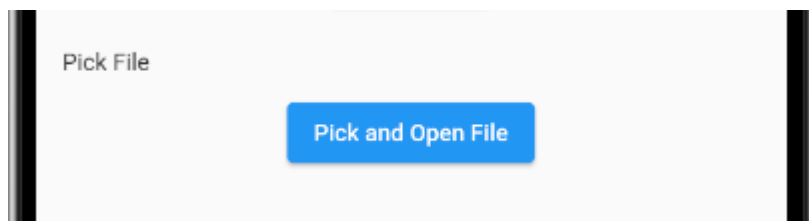
Kemudian panggil method `buildFilePicker` yang sudah dibuat, dan juga tambahkan widget `SizedBox` agar jarak `ColorPicker` dan `FilePicker` tidak terlalu berdekatan.

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Interactive Widget'),
      centerTitle: true,
    ), // AppBar
    body: Container(
      padding: const EdgeInsets.all(16),
      child: ListView(
        children: [
          buildDatePicker(context),
          const SizedBox(height: 20),
          buildColorPicker(context),
          const SizedBox(height: 20),
          buildFilePicker(context),
        ],
      ), // ListView
    ), // Container
  ); // Scaffold
}

```

Coba jalankan aplikasi seharusnya sudah tampil seperti gambar berikut



Namun fungsional dari widgetnya belum bisa, karena kita belum membuatnya.

- **Menambahkan fungsi untuk explore dan pilih files**

Selanjutnya adalah kita akan membuat fungsi untuk mengeksplor dan memilih files dari storage. Buatlah method bernama `_pickFile` dan tetap di dalam class `_AdvancedFormState`

```

void _pickFile() async {
  final result = await FilePicker.platform.pickFiles();
  if (result == null) return;
}

```

- **Memanggil fungsi `_pickFile`**

Langkah berikutnya kita panggil method `_pickFile` yang sudah dibuat di dalam `onPressed` pada widget `buildFilePicker`

```
Center(
  child: ElevatedButton(
    onPressed: () {
      _pickFile();
    },
    child: const Text('Pick and Open File'),
  ), // ElevatedButton
), // Center
```

- **Membuat fungsi untuk membuka files**

Selanjutnya membuat sebuah fungsi untuk membuka files yang telah dipilih

```
void _openFile(PlatformFile file) {
  OpenFile.open(file.path);
}
```

- **Mengambil file dari object result**

Tambah beberapa baris kode pada method _pickFile, karena disini akan mengambil file dari object result dan memanggil method _openFile di dalam method _pickFile

```
void _pickFile() async {
  final result = await FilePicker.platform.pickFiles();
  if (result == null) return;

  //Mendapatkan file dari object result
  final file = result.files.first;
  _openFile(file);
}
```

- **Memanggil fungsi setState**

Kemudian masih di method _pickFile, kita panggil fungsi setState

```
void _pickFile() async {
  final result = await FilePicker.platform.pickFiles();
  if (result == null) return;

  //Mendapatkan file dari objek result
  final file = result.files.first;

  //Memanggil setState untuk memperbarui tampilan UI
  setState(() {
    //Memperbarui nilai _dataFile dengan nama file yang dipilih
    _dataFile = file.name;
  });

  _openFile(file);
}
```

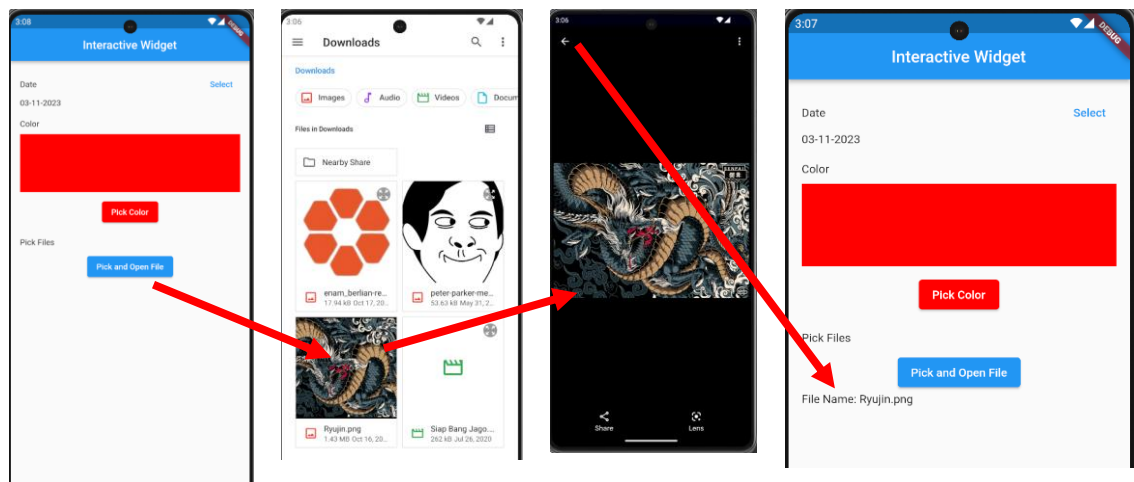
- **Jalankan aplikasi**

Stop emulator, kemudian running lagi.

- **Hasil**

Ketika button ditekan maka akan membuka storage device, kemudian file yang dipilih akan terbuka, kemudian nama file yang dipilih akan tampil di UI.

Untuk menyimpan file foto, pdf, video, dll pada device emulator coba lakukan drag & drop file dari laptop kalian ke emulator, dan file tersebut biasanya akan ada pada folder Downloads di emulator



- **Menampilkan gambar**

Selanjutnya kita akan membuat bagaimana caranya menampilkan gambar yang dipilih. Yang pertama dilakukan adalah kita inisialisasi dan deklarasi variabel pada file `advancaed_form` berikut

```

lib > advanced_form.dart > _AdvancedFormState
1  import 'package:flutter/material.dart';
2  import 'package:intl/intl.dart';
3  import 'package:flutter_colorpicker/flutter_colorpicker.dart';
4  import 'package:file_picker/file_picker.dart';
5  import 'package:open_file/open_file.dart';
6  import 'dart:io'; // Import untuk File
7
8  class AdvancedForm extends StatefulWidget {
9    const AdvancedForm({super.key});
10
11    @override
12    State<AdvancedForm> createState() => _AdvancedFormState();
13  }
14
15  class _AdvancedFormState extends State<AdvancedForm> {
16    DateTime _dueDate = DateTime.now();
17    final currentDate = DateTime.now();
18    Color _currentColor = Colors.orange;
19    String? dataFile;
20    File? _imageFile;

```

Kemudian pada widget **buildFilePicker** tambahkan kode berikut untuk menampilkan gambar

```

Widget buildFilePicker(BuildContext context) {
  return Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      const Text('Pick File'),
      const SizedBox(height: 10),
      Center(
        child: ElevatedButton(
          onPressed: () {
            _pickFile();
          },
          child: const Text('Pick and Open File'),
        ), // ElevatedButton
      ), // Center
      if (dataFile != null) Text('File Name: $dataFile'),
      const SizedBox(height: 10),
      // Menampilkan gambar jika ada
      if (_imageFile != null)
        Image.file(
          _imageFile!,
          height: 200, // Ukuran gambar
          width: double.infinity,
          fit: BoxFit.cover, // Menyesuaikan gambar
        ), // Image.file
    ],
  ); // Column
}

```

Kemudian pada method **_pickFile** tambah kode berikut


```

void _pickFile() async {
  final result = await FilePicker.platform.pickFiles();
  if (result == null) return;

  //Mendapatkan file dari objek result
  final file = result.files.first;

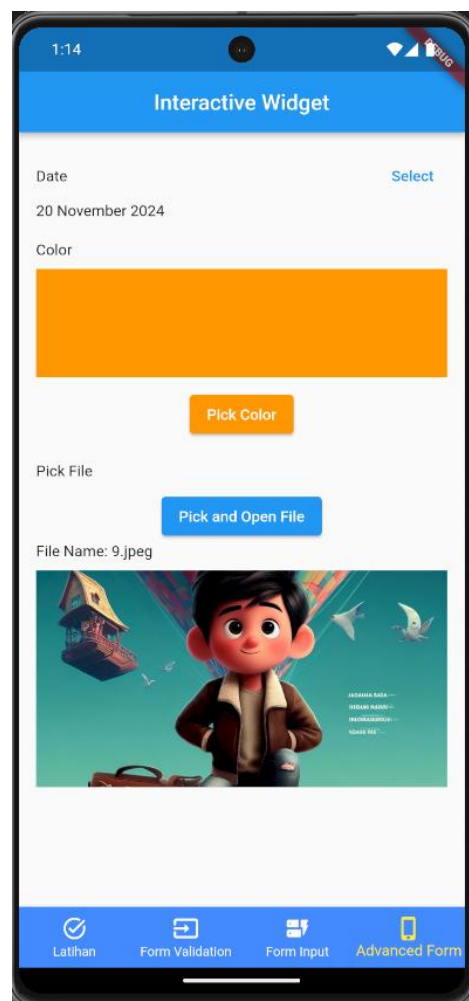
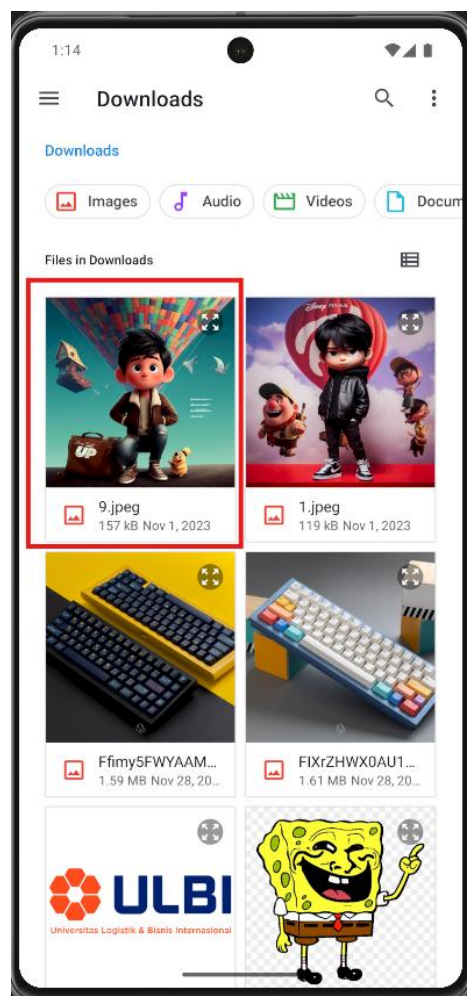
  // Mengecek apakah file yang dipilih adalah gambar
  if (file.extension == 'jpg' ||
      file.extension == 'png' ||
      file.extension == 'jpeg') {
    setState(() {
      _imageFile = File(file.path!); // Mengambil file gambar
    });
  }

  //Memanggil setState untuk memperbarui tampilan UI
  setState(() {
    //Memperbarui nilai _dataFile dengan nama file yang dipilih
    _dataFile = file.name;
  });

  _openFile(file);
}

```

Kemudian sekarang pilih file yang ada pada device emulator kalian



SOAL LATIHAN :

1. Apa perbedaan antara variabel `_dueDate` dan `currentDate` dalam kode di atas?
2. Lihat kode berikut, maksud dari kode tersebut apa dan apa yang terjadi dengan `+5`?

```
lastDate: DateTime(currentDate.year + 5),
```

3. Apa yang dimaksud dengan `_currentColor` dalam kode pada praktikum dan bagaimana nilainya diubah?
4. Tuliskan kode mana yang menampilkan dialog dengan opsi pemilihan warna dalam Flutter!
5. Apa peran `setState` dalam kode ini? Berikan contoh penerapannya!
6. Mengapa memanggil `setState` setelah memilih dan membuka file dalam `_pickFile`?
7. Sebutkan package apa saja yang digunakan dalam proyek ini dan masing-masing kegunaannya!
8. Bagaimana cara menampilkan gambar yang dipilih pengguna? Jelaskan alur kodenya!
9. Jelaskan bagaimana kode ini memanfaatkan package `open_file` dan apa tujuannya!
10. Jelaskan fungsi dari `buildFilePicker` dan bagaimana cara kerjanya!

Pengumpulan Hasil :

Jika sudah berhasil, hasil praktikum dimasukkan ke dalam github masing-masing. Buat folder dengan nama **Pertemuan07** kemudian di dalamnya buat 2 folder yaitu **folder Praktikum** dan **Output**.

Isi folder **Praktikum** adalah aplikasi flutter yang sudah dibuat, tetapi **sebelum melakukan push ke github** wajib jalankan perintah “**flutter clean**” pada VSCode. Isi folder Output adalah hasil dari aplikasi flutter yang sudah dibuat dengan format gif.

Setelah melakukan perintah **flutter clean** kalian bisa langsung push ke dalam repository kalian.

Yang dikumpulkan di **Edlink** adalah Jawaban dari soal latihan di atas.