

**LAPORAN PRAKTIKUM**  
**MATA KULIAH PEMROGRAMAN MOBILE**  
**PERTEMUAN KE-6**



**U N I V E R S I T A S**  
**AMIKOM PURWOKERTO**

Wasis Rifky Pamuji

22SA11A236

IF 22 F

## A. Tampilan Utama

### Latihan Formulir

Nama Lengkap  
1222

email  
wa

Nomor Telpn  
qwqwwqqwwqw

☐ Pria

☒ Wanita

Silakan Pilih Agama Yang Anda Anut ▾

Kemampuan Berbahasa

☒ indonesia

☒ inggris

☐ arab

☒ sunda

☒ jawa

☐ madura

☒ jepang

☒ korea

☐ mandarin

Tanggal Daftar  
2024-05-22

Jam Daftar  
10:06

Simpan

Lihat Data

Pada praktikum ke 6 ini kita akan membuat sebuah tampilan berbentuk formulir dengan beberapa widget diantaranya text field, dropdown button, option button, checkbox, text field dengan format tanggal dan jam, serta terdapat tombol simpan untuk menyimpan data yang diinputkan, terdapat juga tombol lihat data untuk menampilkan data yang telah disimpan.

## B. Kode Program

- File model.dart

```
1 import 'package:flutter/material.dart';
2
3 class Pendaftaran {
4   BuildContext context;
5   String nama = "";
6   String email = "";
7   String noTelp = "";
8   String jeniskelamin = "";
9   String bahasa = "";
10  String agama = "";
11  String tanggalDaftar = "";
12  String jamDaftar = "";
13
14  Pendaftaran(
15    this.context,
16    this.nama,
17    this.email,
18    this.noTelp,
19    this.jeniskelamin,
20    this.bahasa,
21    this.agama,
22    this.tanggalDaftar,
23    this.jamDaftar,
24  );
```

Mendeklarasikan variabel nama, email, notelp dan yang lainnya yang akan digunakan untuk menyimpan dan mengelola data

```

26  bool sudahValid() {
27      if (nama.isEmpty) {
28          tampilkanPesan("Nama Wajib Diisi");
29          return false;
30      }
31      if (email.isEmpty) {
32          tampilkanPesan("Email Wajib Diisi");
33          return false;
34      }
35      if (noTelp.isEmpty) {
36          tampilkanPesan("noTelp Wajib Diisi");
37          return false;
38      }
39      if (jeniskelamin.isEmpty) {
40          tampilkanPesan("jeniskelamin Wajib Diisi");
41          return false;
42      }
43      if (bahasa.isEmpty) {
44          tampilkanPesan("bahasa Wajib Diisi");
45          return false;
46      }
47      if (agama.isEmpty) {
48          tampilkanPesan("agama Wajib Diisi");
49          return false;
50      }
51      if (tanggalDaftar.isEmpty) {
52          tampilkanPesan("tanggalDaftar Wajib Diisi");
53          return false;
54      }
55      if (jamDaftar.isEmpty) {
56          tampilkanPesan("jamDaftar Wajib Diisi");
57          return false;
58      }
59
60      return true;
61  }

```

Membuat fungsi dengan nama 'sudahValid' dengan tipe boolean yang digunakan untuk memeriksa apakah semua data yang diperlukan dalam proses pendaftaran, apakah sudah terisi atau tidak. Fungsi ini mengembalikan nilai true jika semua data sudah terisi dan false jika ada setidaknya satu data yang masih kosong.

```

63  void tampilkanPesan(String s) {
64      ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text(s)));
65  }
66  }
67
68  List<Pendaftaran> pendaftaranList = [];

```

Membuat fungsi untuk menampilkan pesan yang akan digunakan untuk menampilkan pesan notifikasi atau peringatan kepada pengguna dalam bentuk snackbar

- Fiel agama.dart

```

1  import 'package:flutter/material.dart';
2
3  String? agamaDipilih;
4
5  class Agama extends StatefulWidget {
6    const Agama({super.key});
7
8    @override
9    State<Agama> createState() => _AgamaState();
10 }
11
12 class _AgamaState extends State<Agama> {
13   List<String> agamaList = [
14     "Islam",
15     "Katolik",
16     "Protestan",
17     "Hindu",
18     "Budha",
19     "Konghucu",
20     "Penganut Kepercayaan",
21   ];

```

Mendeklarasikan variabel 'agamaDipilih' tipe data string dengan nullable. Terdapat class statefulwidget yang didalamnya terdapat list berisi daftar agama.

```

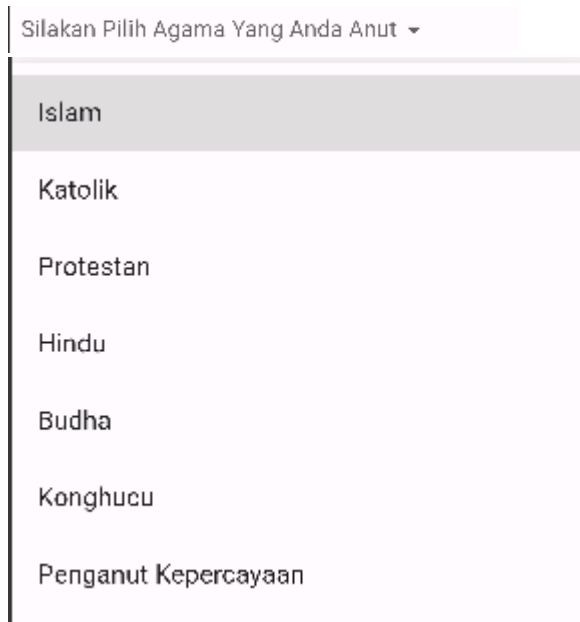
23   @override
24   Widget build(BuildContext context) {
25     return DropdownButton(
26       hint: Text("Silakan Pilih Agama Yang Anda Anut"),
27       value: agamaDipilih,
28       items: agamaList
29         .map((e) => DropdownMenuItem(
30           child: Text(e),
31           value: e,
32         )) // DropdownMenuItem
33         .toList(),
34       onChanged: (value) {
35         setState(() {
36           agamaDipilih = value;
37         });
38       },
39     ); // DropdownButton
40   }
41 }

```

Pada bagian return deklarasikan widget dropdown button, hint digunakan untuk menampilkan teks panduan, value digunakan untuk menentukan nilai yang dipiih, items digunakan untuk menentukan daftar item yang akan ditampilkan dalam dropdown button

dan item ini diambil dari list agamaList, onChange adalah callback yang dipanggil ketika pengguna memilih salah satu item. Didalamnya terdapat setState() digunakan untuk memperbarui nilai agamaDipilih.

Output:



- File agama.dart

```
1  import 'package:flutter/material.dart';
2
3  List<String> bahasaDipilihList = [];
4
5  class KemampuanBerbahasa extends StatefulWidget {
6    const KemampuanBerbahasa({super.key});
7
8    @override
9    State<KemampuanBerbahasa> createState() => _KemampuanBerbahasaState();
10 }
11
12 class _KemampuanBerbahasaState extends State<KemampuanBerbahasa> {
13   List<String> bahasaList = [
14     "indonesia",
15     "inggris",
16     "arab",
17     "sunda",
18     "jawa",
19     "madura",
20     "jepang",
21     "korea",
22     "mandarin",
23   ];
```

Mendeklarasikan list kosong dengan nama 'BahasaDipilihList' yang akan digunakan untuk menyimpan nilai. Terdapat class statefulwidget yang didalamnya terdapat list berisi daftar bahasa.

```
24  @override
25  Widget build(BuildContext context) {
26    return Column(
27      crossAxisAlignment: CrossAxisAlignment.start,
28      children: [
29        Text("Kemampuan Berbahasa"),
30        Container(
31          height: 90,
32          child: GridView(
33            gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
34              crossAxisCount: 3, childAspectRatio: 4 / 1), // Sliv
35            children: bahasaList
36              .map((e) => OpsiBahasa(
37                strBahasa: e,
38              )) // OpsiBahasa
39              .toList(),
40            ), // GridView
41          ), // Container
42        ],
43      ); // Column
44  }
45  }
```

Pada bagian return diatur dengan bentuk column dimulai dari sisi kiri, terdapat text judul 'Kemampuan Berbahasa'. Didalam container terdapat widget GridView untuk membuat grid yang akan menampilkan daftar bahasa. crossAxisCount : 3 untuk mengatur menjadi 3 kolom dan childAspectRatio : 4/1 untuk mengatur rasio setiap item didalam grid. Kemudian bahasaList diubah menjadi Opsibahasa.

```

47 class OpsiBahasa extends StatefulWidget {
48   final String strBahasa;
49   const OpsiBahasa({super.key, required this.strBahasa});
50
51   @override
52   State<OpsiBahasa> createState() => _OpsiBahasaState();
53 }
54
55 class _OpsiBahasaState extends State<OpsiBahasa> {
56   bool? opsiDipilih = false;
57   @override
58   Widget build(BuildContext context) {
59     return Row(
60       children: [
61         Checkbox(
62           value: opsiDipilih,
63           onChanged: (value) {
64             setState(() {
65               opsiDipilih = value;
66               if (value == true) {
67                 bahasaDipilihList.add(widget.strBahasa);
68               } else {
69                 bahasaDipilihList.remove(widget.strBahasa);
70               }
71             });
72           },
73         ), // Checkbox
74         Text(widget.strBahasa)
75       ],
76     ); // Row
77   }
78 }

```

Mendeklarasikan class statefulWidget untuk mengatur tampilan setiap opsi bahasa. variabel opsiDipilih bertipe data bool? yang berarti bisa memiliki nilai true, false, atau null karena menggunakan tanda tanya (?). Nilai awal variabel ini diatur menjadi false. Pada bagian return mengembalikan widget row untuk mengatur elemen secara horizontal. Didalamnya terdapat widget checkbox yang menampilkan kotak centang dengan value dari opsiBahasa. Kemudian terdapat onChange yang didalamnya terdapat logika untuk mengubah nilai variabel opsiDipilih.

#### Output

Kemampuan Berbahasa

<input checked="" type="checkbox"/> indonesia	<input type="checkbox"/> inggris	<input type="checkbox"/> arab
<input checked="" type="checkbox"/> sunda	<input type="checkbox"/> jawa	<input type="checkbox"/> madura
<input type="checkbox"/> jepang	<input type="checkbox"/> korea	<input type="checkbox"/> mandarin



- File tglDaftar.dart

```

1  import 'package:flutter/material.dart';
2
3  TextEditingController tanggalDaftarController = TextEditingController();
4
5  class TanggalDaftar extends StatefulWidget {
6    const TanggalDaftar({super.key});
7
8    @override
9    State<TanggalDaftar> createState() => _tanggalDaftarState();
10 }
11
12 class _tanggalDaftarState extends State<TanggalDaftar> {
13   @override
14   Widget build(BuildContext context) {
15     return TextField(
16       controller: tanggalDaftarController,
17       decoration: InputDecoration(
18         label: Text("Tanggal Daftar"),
19         suffixIcon: IconButton(
20           onPressed: () {
21             pilihTanggal();
22           },
23           icon: Icon(Icons.date_range),
24         ), // IconButton
25       ), // InputDecoration
26     ); // TextField
27   }
28

```

Mendeklarasikan text field dengan label 'Tanggal Daftar', suffixIcon digunakan untuk menambahkan ikon, kemudian terdapat IconButton yang berisi Icone.date\_range berfungsi ketika ikon diklik akan memanggil fungsi pilihtanggal

```

29  Future<void> pilihTanggal() async {
30    DateTime? tglDidapat;
31
32    tglDidapat = await showDatePicker(
33      context: context,
34      firstDate: DateTime(2024),
35      lastDate: DateTime(2025),
36    );
37
38    if (tglDidapat.toString() != "null") {
39      tanggalDaftarController.text = tglDidapat.toString().substring(0, 10);
40    }
41  }
42

```

mendeklarasikan fungsi pilihTanggal. Di dalamnya terdapat await showDatePicker untuk menampilkan dialog pemilih tanggal dan menunggu pengguna memilih tanggal. Properti firstDate dan lastDate digunakan untuk menentukan rentang tanggal yang dapat dipilih. Lalu terdapat logika, jika tanggal yang dipilih tidak null, maka akan mengubah teks pada tanggalDaftarController dengan tanggal yang dipilih. kemudian pada bagian tglDidapat.toString().substring(0, 10) untuk mendapatkan string representasi tanggal dalam format "yyyy-MM-dd".

Output:



- File jamDaftar.dart

```
1  import 'package:flutter/material.dart';
2
3  TextEditingController jamDaftarController = TextEditingController();
4
5  class JamDaftar extends StatefulWidget {
6    const JamDaftar({super.key});
7
8    @override
9    State<JamDaftar> createState() => _JamDaftarState();
10 }
11
12 class _JamDaftarState extends State<JamDaftar> {
13   @override
14   Widget build(BuildContext context) {
15     return TextField(
16       controller: jamDaftarController,
17       decoration: InputDecoration(
18         label: Text("Jam Daftar"),
19         suffixIcon: IconButton(
20           onPressed: () {
21             pilihTanggal();
22           },
23           icon: Icon(Icons.timer),
24         ), // IconButton
25       ), // InputDecoration
26     ); // TextField
27   }
```

```

29  Future<void> pilihTanggal() async {
30      TimeOfDay? tglDidapat;
31
32      tglDidapat =
33      |   await showTimePicker(context: context, initialTime: TimeOfDay.now());
34
35  ~   if (tglDidapat.toString() != "null") {
36      |   jamDaftarController.text = tglDidapat
37      |   |   .toString()
38      |   |   .replaceAll('TimeOfDay(', '')
39      |   |   .replaceAll(')', '');
40      |   }
41      }
42  }

```

Hampir sama seperti file `tglDaftar.dart` hanya pada fungsi `pilihTanggal` yang berbeda. Mendeklarasikan variabel `tglDidapat` dengan tipe `TimeOfDay` untuk menyimpan waktu yang dipilih. Di dalam fungsi, terdapat `await showTimePicker()` untuk menampilkan dialog pemilih waktu dan menunggu pengguna memilih waktu. Properti `initialTime` digunakan untuk menentukan waktu awal yang ditampilkan di dialog pemilih waktu. Kemudian terdapat logika, jika waktu yang dipilih tidak null, maka akan mengubah teks pada `jamDaftarController` dengan waktu yang dipilih. Pada bagian `tglDidapat.toString().replaceAll('TimeOfDay(', '').replaceAll(')', '')` untuk mendapatkan string representasi waktu dalam format "HH:mm".

Output:

Jam Daftar



- File `dataPendaftar.dart`

```

14  ~   return Scaffold(
15  ~   |   appBar: AppBar(
16      |   |   title: Text("Data Pendaftaran"),
17      |   |   ), // AppBar
18  ~   |   body: ListView.builder(
19      |   |   itemCount: pendaftaranList.length,
20  ~   |   |   itemBuilder: (context, index) => Card(
21  ~   |   |   |   child: ListTile(
22      |   |   |   |   title: Text(pendaftaranList[index].nama),
23  ~   |   |   |   |   subtitle: Column(
24      |   |   |   |   |   crossAxisAlignment: CrossAxisAlignment.start,
25  ~   |   |   |   |   |   children: [
26      |   |   |   |   |   |   Text(pendaftaranList[index].email),
27      |   |   |   |   |   |   Text(pendaftaranList[index].noTelp),
28      |   |   |   |   |   |   Text(pendaftaranList[index].jeniskelamin),
29      |   |   |   |   |   |   Text(pendaftaranList[index].bahasa),
30      |   |   |   |   |   |   Text(pendaftaranList[index].agama),
31      |   |   |   |   |   |   Text(pendaftaranList[index].tanggalDaftar),
32      |   |   |   |   |   |   Text(pendaftaranList[index].jamDaftar),
33      |   |   |   |   |   ],

```

Terdapat appBar dengan text 'Data Pendaftar'. Pada bagian body terdapat ListView.builder untuk menampilkan daftar data pendaftaran. Itemcount untuk menentukan jumlah item yang akan ditampilkan, dihitung berdasarkan panjang pendaftaranList. Kemudian terdapat ListTile dengan title nama, kemudian subtitle berisi data yang lainnya.

```
35 | trailing: IconButton(  
36 | | onPressed: () {  
37 | | hapusData(index);  
38 | | },  
39 | | icon: Icon(  
40 | | Icons.delete,
```

Menambahkan IconButton dengan properti onPressed berfungsi untuk memanggil fungsi hapusdata(index).

```
49 | void hapusData(int index) {  
50 | | showDialog(  
51 | | | context: context,  
52 | | | builder: (context) => AlertDialog(  
53 | | | | title: Text("Hapus Data"),  
54 | | | | content: Text(  
55 | | | | | "Apakah anda yakin akan hapus data ${pendaftaranList[index].nama} ?"),  
56 | | | | actions: [  
57 | | | | | ElevatedButton(  
58 | | | | | | onPressed: () {  
59 | | | | | | | setState(() {  
60 | | | | | | | | pendaftaranList.removeAt(index);  
61 | | | | | | | | Navigator.pop(context);  
62 | | | | | | | | });  
63 | | | | | | },  
64 | | | | | | child: Text("Ya"),  
65 | | | | | ), // ElevatedButton  
66 | | | | | ElevatedButton(  
67 | | | | | | onPressed: () {  
68 | | | | | | | Navigator.pop(context);  
69 | | | | | | },  
70 | | | | | | child: Text("Tidak"),  
71 | | | | | ), // ElevatedButton  
72 | | | | ],  
73 | | | ), // AlertDialog  
74 | | );  
75 | }  
76 | }
```

Mendeklarasikan fungsi hapusData, didalamnya terdapat widget showDialog untuk menampilkan dialog konfirmasi dengan AlertDialog. Kemudian terdapat title hapus data lalu content untuk menampilkan pesan konfirmasi. Action digunakan untuk menambahkan tombol aksi pada dialog. Kemudian ElevatedButton membuat tombol dengan fungsi menghapus data dalam list pendaftaranList dan tombol dengan fungsi untuk membatalkan aksi dengan nama tidak.

- File main.dart

```

1  import 'package:flutter/material.dart';
2  import 'package:tugas6/agama.dart';
3  import 'package:tugas6/bahasa.dart';
4  import 'package:tugas6/datapendaftar.dart';
5  import 'package:tugas6/jamDaftar.dart';
6  import 'package:tugas6/model.dart';
7  import 'package:tugas6/tglDaftar.dart';
8
  Run | Debug | Profile
9  void main(List<String> args) {
10 |   runApp(MyApp());
11 | }
12

```

Mengimport package yang diperlukan. Kemudian membuat method untuk menjalankan aplikasi

```

13 class MyApp extends StatelessWidget {
14   const MyApp({super.key});
15
16   @override
17   Widget build(BuildContext context) {
18     return MaterialApp(
19       theme: ThemeData(
20         appBarTheme: AppBarTheme(
21           backgroundColor: Colors.blue,
22           foregroundColor: Colors.white,
23         ), // AppBarTheme
24         elevatedButtonTheme: ElevatedButtonThemeData(
25           style: ButtonStyle(
26             backgroundColor: MaterialStatePropertyAll(Colors.blue),
27             foregroundColor: MaterialStatePropertyAll(Colors.white),
28           ), // ButtonStyle
29         ), // ElevatedButtonThemeData
30       ), // ThemeData
31       debugShowCheckedModeBanner: false,
32       home: MyHomePage(),
33     ); // MaterialApp
34   }
35 }

```

Deklarasikan class statelesswidget, pada bagian return terdapat appBarthem untuk mengatur warna background dan foreground pada appbar. Kemudian buttonstyle untuk mengatur warna background dan foreground tampilan tombol.

```

37 class MyHomePage extends StatefulWidget {
38   const MyHomePage({super.key});
39
40   @override
41   State<MyHomePage> createState() => _MyHomePageState();
42 }
43
44 class _MyHomePageState extends State<MyHomePage> {
45   TextEditingController namaController = TextEditingController();
46   TextEditingController emailController = TextEditingController();
47   TextEditingController noTelpController = TextEditingController();
48
49   String? jeniskelamin;

```

Pada class StatefulWidget mendeklarasikan TextEditingController untuk menampung

```

51   @override
52   Widget build(BuildContext context) {
53     return Scaffold(
54       appBar: AppBar(
55         title: Text("Latihan Formulir"),
56       ), // AppBar
57       body: Padding(
58         padding: const EdgeInsets.all(8.0),
59         child: ListView(
60           children: [
61             TextField(
62               controller: namaController,
63               decoration: InputDecoration(
64                 label: Text("Nama Lengkap"),
65               ), // InputDecoration
66             ), // TextField
67             TextField(
68               controller: emailController,
69               keyboardType: TextInputType.emailAddress,
70               decoration: InputDecoration(
71                 label: Text("email"),
72               ), // InputDecoration
73             ), // TextField
74             TextField(
75               controller: noTelpController,
76               keyboardType: TextInputType.phone,
77               decoration: InputDecoration(
78                 label: Text("Nomor Telpon"),
79               ), // InputDecoration
80             ), // TextField

```

Membuat text field untuk menginputkan nama, email, dan nomor telpon

```

81 Row(
82   mainAxisAlignment: MainAxisAlignment.spaceEvenly,
83   children: [
84     Expanded(
85       child: RadioListTile(
86         title: Text("Pria"),
87         value: "Pria",
88         groupValue: jeniskelamin,
89         onChanged: (value) {
90           setState(() {
91             jeniskelamin = value;
92           });
93         },
94       ), // RadioListTile
95     ), // Expanded
96     Expanded(
97       child: RadioListTile(
98         title: Text("Wanita"),
99         value: "Wanita",
100        groupValue: jeniskelamin,
101        onChanged: (value) {
102          setState(() {
103            jeniskelamin = value;
104          });
105        },
106      ), // RadioListTile
107    ), // Expanded
108  ],
109 ), // Row

```

Kemudian terdapat row yang berisi radioListTile untuk memilih jenis kelamin

Output:

☐ Pria
 ☐ Wanita

```

110 Agama(),
111 KemampuanBerbahasa(),
112 TanggalDaftar(),
113 JamDaftar(),

```

Memanggil class yang sudah dibuat

```

114         Padding(
115             padding: const EdgeInsets.all(8.0),
116             child: Row(
117                 mainAxisAlignment: MainAxisAlignment.spaceEvenly,
118                 children: [
119                     ElevatedButton(
120                         onPressed: () {
121                             setState(() {
122                                 Pendaftaran pendaftaran = Pendaftaran(
123                                     context,
124                                     namaController.text,
125                                     emailController.text,
126                                     noTelpController.text,
127                                     jeniskelamin.toString(),
128                                     bahasaDipilihList.toString(),
129                                     agamaDipilih.toString(),
130                                     tanggalDaftarController.text,
131                                     jamDaftarController.text,
132                                 ); // Pendaftaran
133
134                                 if (pendaftaran.sudahValid()) {
135                                     pendaftaranList.add(pendaftaran);
136
137                                     Navigator.push(
138                                         context,
139                                         MaterialPageRoute(
140                                             builder: (context) => DataPendaftaran(),
141                                         ), // MaterialPageRoute
142                                     );
143                                 }
144                             });
145                 ],

```

Membuat tombol simpan dengan elevated button, kemudian didalamnya terdapat logika jika pendaftaran sudahValid maka tambahkan ke list pendaftaranList. Navigator.push untuk melakukan navigasi ke halaman DataPelanggan.

```

148         ),
149         ElevatedButton(
150             onPressed: () {
151                 Navigator.push(
152                     context,
153                     MaterialPageRoute(
154                         builder: (context) => DataPendaftaran(),
155                     ), // MaterialPageRoute
156                 );
157             },
158             child: Text("Lihat Data"),

```



Membuat tombol Lihat Data dengan fungsi membuka halaman DataPendaftar