

LAPORAN PRAKTIKUM
MATA KULIAH PEMROGRAMAN MOBILE



**UNIVERSITAS
AMIKOM PURWOKERTO**

Oleh :
NAMA: FAJAR RAMADHAN
NIM: 23SA31A065

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM
PURWOKERTO
2025

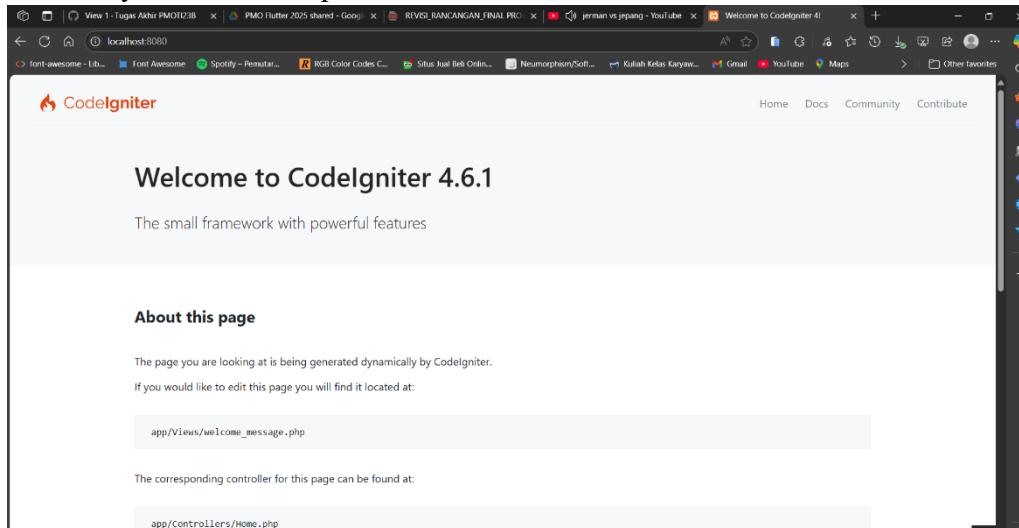
PRAKTIKUM 11 IMPLEMENTASI NGROK DAN API CALLS HTTP DI FLUTTER

- Menjalankan projek rest api pada praktikum sebelumnya dan membuatnya open publik dengan ngrok "ngrok http localhost:8080"

The screenshot shows the ngrok application window with the following details:

```
Session Status      online
Account            fajaramadhan.college@gmail.com (Plan: Free)
Version           3.23.0
Region            Asia Pacific (ap)
Web Interface     http://127.0.0.1:4040
Forwarding        https://5c5a-103-185-27-58.ngrok-free.app -> http://localhost:8080/
Connections       ttl     opn     rtl     rt5     p50     p90
                  0       0       0.00   0.00   0.00   0.00
```

- Maka hasilnya bisa diakses publik

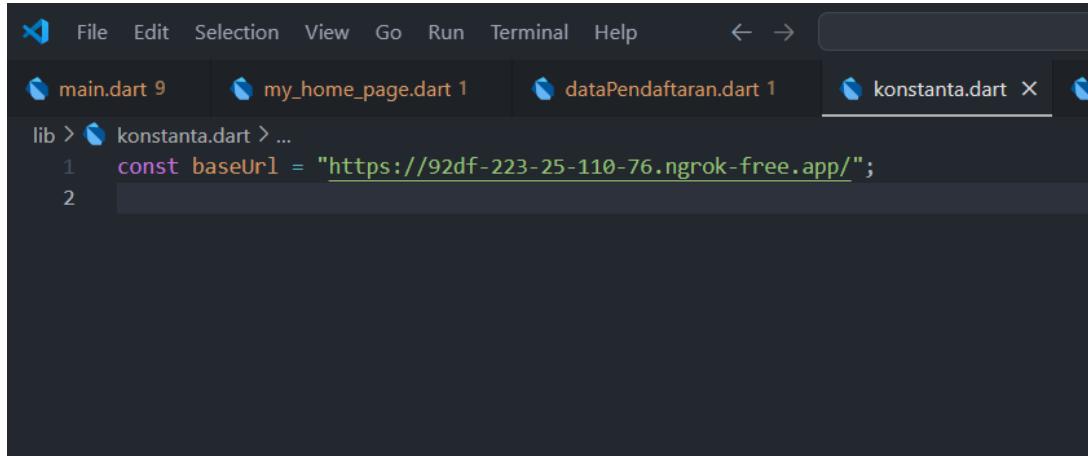


- install http dengan "flutter pub add http" setelah itu "flutter pub get"

```
Pajar@Pajar MINGW64 /d/colleges/Semester 4/PMD/PRAKTIKUM 6/praktikum_6
$ flutter pub add http
Resolving dependencies...
Downloading packages...
async 2.12.0 (2.13.0 available)
fake_async 1.3.2 (1.3.3 available)
flutter_lints 5.0.0 (6.0.0 available)
http 1.4.0
leak_tracker 10.0.8 (11.0.1 available)
leak_tracker_flutter_testing 3.0.9 (3.0.10 available)
leak_tracker_testing 3.0.1 (3.0.2 available)
lints 5.1.1 (6.0.0 available)
material_color_utilities 0.11.1 (0.13.0 available)
meta 1.16.0 (1.17.0 available)
test_api 0.7.4 (0.7.6 available)
typed_data 1.4.0
vector_math 2.1.4 (2.2.0 available)
vm_service 14.3.1 (15.0.2 available)
web 1.1.1
Changed 4 dependencies!
12 packages have newer versions incompatible with dependency constraints.
Try 'flutter pub outdated' for more information.

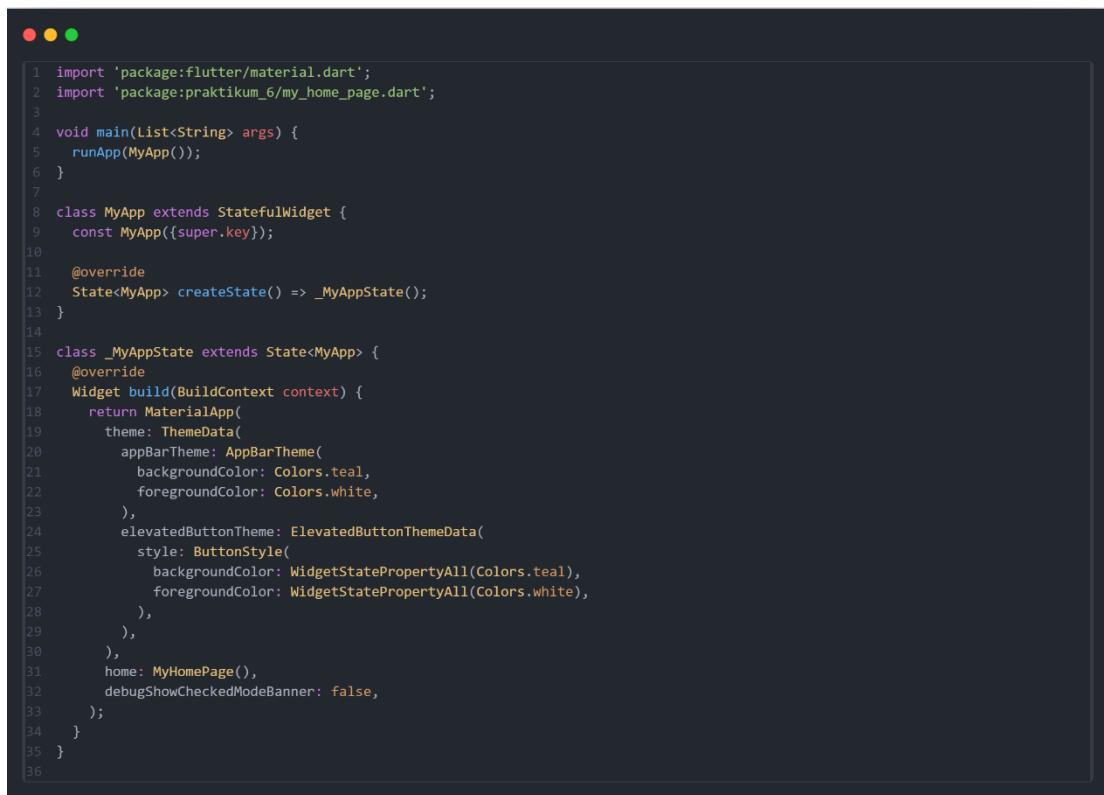
Pajar@Pajar MINGW64 /d/colleges/Semester 4/PMD/PRAKTIKUM 6/praktikum_6
$ flutter pub get
Resolving dependencies... (1.0s)
Downloading packages...
async 2.12.0 (2.13.0 available)
fake_async 1.3.2 (1.3.3 available)
flutter_lints 5.0.0 (6.0.0 available)
leak_tracker 10.0.8 (11.0.1 available)
leak_tracker_flutter_testing 3.0.9 (3.0.10 available)
leak_tracker_testing 3.0.1 (3.0.2 available)
lints 5.1.1 (6.0.0 available)
material_color_utilities 0.11.1 (0.13.0 available)
meta 1.16.0 (1.17.0 available)
test_api 0.7.4 (0.7.6 available)
vector_math 2.1.4 (2.2.0 available)
vm_service 14.3.1 (15.0.2 available)
Got dependencies!
```

4. Membuat baseurl memisahkannya pada file konstanta.dart untuk dapat digunakan secara reusable



```
File Edit Selection View Go Run Terminal Help ← →
main.dart 9 | my_home_page.dart 1 | dataPendaftaran.dart 1 | konstanta.dart X
lib > konstanta.dart > ...
1 const baseUrl = "https://92df-223-25-110-76.ngrok-free.app/";
```

5. kode main.dart menampilkan extracted widget form MyHomepage agar bisa reusable untuk submit data dan update data



```
import 'package:flutter/material.dart';
import 'package:praktikum_6/my_home_page.dart';

void main(List<String> args) {
  runApp(MyApp());
}

class MyApp extends StatefulWidget {
  const MyApp({super.key});

  @override
  State<MyApp> createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(
        appBarTheme: AppBarTheme(
          backgroundColor: Colors.teal,
          foregroundColor: Colors.white,
        ),
        elevatedButtonTheme: ElevatedButtonThemeData(
          style: ButtonStyle(
            backgroundColor: WidgetStatePropertyAll(Colors.teal),
            foregroundColor: WidgetStatePropertyAll(Colors.white),
          ),
        ),
      ),
      home: MyHomePage(),
      debugShowCheckedModeBanner: false,
    );
  }
}
```

6. Kode extracted widget pada my_homepage.dart, saya buat beberapa kondisional dimana memiliki satu argumen jika argumen/parameter terisi maka teridentifikasi sebagai update data jika tidak maka post untuk api calls nya, dan pula kondisi tersebut digunakan untuk navigatior push jika submit data dan navigator pop jika update

```
 1 import 'dart:convert';
 2 import 'package:flutter/material.dart';
 3 import 'package:http/http.dart' as http;
 4 import 'package:praktikum_6/agama.dart';
 5 import 'package:praktikum_6/dataPendaftaran.dart';
 6 import 'package:praktikum_6/jamDaftar.dart';
 7 import 'package:praktikum_6/kemampuanBahasa.dart';
 8 import 'package:praktikum_6/konstanta.dart';
 9 import 'package:praktikum_6/model.dart';
10 import 'package:praktikum_6/tanggalDaftar.dart';
11
12 class MyHomePage extends StatefulWidget {
13   final Pendaftaran? pendaftaran;
14   const MyHomePage({super.key, this.pendaftaran});
15
16   @override
17   State<MyHomePage> createState() => _MyHomePageState();
18 }
19
20 class _MyHomePageState extends State<MyHomePage> {
21   TextEditingController namaController = TextEditingController();
22   TextEditingController emailController = TextEditingController();
23   TextEditingController noTelpController = TextEditingController();
24
25   String? jenisKelamin;
26
27   @override
28   void initState() {
29     super.initState();
30     if (widget.pendaftaran != null) {
31       namaController.text = widget.pendaftaran!.nama ?? '';
32       emailController.text = widget.pendaftaran!.email ?? '';
33       noTelpController.text = widget.pendaftaran!.noTelp ?? '';
34       jenisKelamin = widget.pendaftaran!.jenisKelamin ?? '';
35     }
36   }
37
38   @override
39   Widget build(BuildContext context) {
40     return Scaffold(
41       appBar: AppBar(title: Text("Latihan Formulir")),
42       body: Padding(
43         padding: const EdgeInsets.all(8.0),
44         child: ListView(
45           children: [
46             TextField(
47               controller: namaController,
48               decoration: InputDecoration(label: Text("Nama Lengkap")),
49             ),
50             TextField(
51               controller: emailController,
52               keyboardType: TextInputType.emailAddress,
53               decoration: InputDecoration(label: Text("Email")),
54             ),
55             TextField(
56               controller: noTelpController,
57               keyboardType: TextInputType.phone,
58               decoration: InputDecoration(label: Text("Nomor Telpon")),
59             ),
60             Row(
61               mainAxisAlignment: MainAxisAlignment.spaceEvenly,
62               children: [
63                 Expanded(
64                   child: RadioListTile(
65                     title: Text("Pria"),
66                     value: "Pria",
67                     groupValue: jenisKelamin,
68                     onChanged: (value) {
69                       setState(() {
70                         jenisKelamin = value;
71                       });
72                     },
73                   ),
74                 ),
75                 Expanded(
76                   child: RadioListTile(
77                     title: Text("Wanita"),
78                     value: "Wanita",
79                     groupValue: jenisKelamin,
80                     onChanged: (value) {
81                       setState(() {
82                         jenisKelamin = value;
83                       });
84                     },
85                   ),
86                 ),
87               ],
88             ),
89           ],
90         ),
91       ),
92     );
93   }
94 }
```

```

54 ),
55     TextField(
56       controller: noelpController,
57       keyboardType: TextInputType.phone,
58       decoration: InputDecoration(label: Text("Nomor Telpon")),
59     ),
60   Row(
61     mainAxisAlignment: MainAxisAlignment.spaceEvenly,
62     children: [
63       Expanded(
64         child: RadioListTile(
65           title: Text("Pria"),
66           value: "Pria",
67           groupValue: jenisKelamin,
68           onChanged: (value) {
69             setState(() {
70               jenisKelamin = value;
71             });
72           },
73         ),
74       ),
75       Expanded(
76         child: RadioListTile(
77           title: Text("Wanita"),
78           value: "Wanita",
79           groupValue: jenisKelamin,
80           onChanged: (value) {
81             setState(() {
82               jenisKelamin = value;
83             });
84           },
85         ),
86       ),
87     ],
88   ),
89   Kemampuanbahasa(),
90   Agama(),
91   Tanggaldaftar(),
92   Jandaftar(),
93   Row(
94     mainAxisAlignment: MainAxisAlignment.spaceEvenly,
95     children: [
96       ElevatedButton(
97         onPressed: () async {
98           Pendaftaran dataPendaftaran = Pendaftaran(
99             id: widget.pendaftaran?.id,
100            nama: namaController.text,
101            email: emailController.text,
102            noelpon: noelpController.text,
103            jenisKelamin: jenisKelamin.toString(),
104            bahasa: bahasaDipilihList.toString(),
105            agama: agamaDipilih.toString(),
106            tanggalDaftar: tanggalDaftarDaftarController.text,
107            jamDaftar: jamDaftarDaftarController.text,
108          );
109          http.Response response;
110          if (widget.pendaftaran == null) {
111            var url = Uri.parse(baseUrl + 'pendaftaran');
112            respons = await http.post(
113              url,
114              headers: {
115                'Content-Type': 'application/json; charset=UTF-8',
116              },
117              body: jsonEncode(dataPendaftaran.toJson()),
118            );
119          } else {
120            var url = Uri.parse(
121              baseUrl + 'pendaftaran/${dataPendaftaran.id}',
122            );
123            respons = await http.put(
124              url,
125              headers: {
126                'Content-type': 'application/json; charset=UTF-8',
127              },
128              body: jsonEncode(dataPendaftaran.toJson()),
129            );
130          }
131        if (respons.statusCode == 200 || respons.statusCode == 201) {
132          if (widget.pendaftaran == null) {
133            Navigator.push(
134              context,
135              MaterialPageRoute(
136                builder: (context) => Datapendaftaran(),
137              ),
138            );
139          } else {
140            Navigator.pop(context, true);
141          }
142        } else {
143          Map<String, dynamic> responsDecode =
144            jsonDecode(respons.body) as Map<String, dynamic>;
145          ScaffoldMessenger.of(context).showSnackBar(
146            SnackBar(
147              content: Text(
148                responsDecode['messages'].toString().replaceAll(
149                  "\r",
150                  "\n",
151                  "\n\n",
152                ),
153              ),
154            ),
155          );
156        }
157      },
158    ),
159    child: Text("Simpan"),
160  ),
161  ElevatedButton(
162    onPressed: () {
163      Navigator.push(
164        context,
165        MaterialPageRoute(
166          builder: (context) => Datapendaftaran(),
167        ),
168      );
169    },
170    child: Text("Lihat Data"),
171  ),
172 ],
173 ),
174 ],
175 ),
176 );
177 );
178 }
179 }
180

```

7. Kode pada data pendaftaran untuk menampilkan list data sesuai struktur pada model dengan api calls get

```
1 // ignore: file_names
2 import 'dart:convert';
3
4 import 'package:flutter/material.dart';
5 import 'package:praktikum_6/constantes.dart';
6 import 'package:praktikum_6/model.dart';
7 import 'package:http/http.dart' as http;
8 import 'package:praktikum_6/my_home_page.dart';
9
10 class Datapendaftaran extends StatefulWidget {
11   const Datapendaftaran({super.key});
12
13   @override
14   State<Datapendaftaran> createState() => _DatapendaftaranState();
15 }
16
17 class _DatapendaftaranState extends State<Datapendaftaran> {
18   List<Pendaftaran> pendaftaranList = [];
19
20   Future ambilDataPendaftaran() async {
21     var url = Uri.parse(baseUrl + 'pendaftaran');
22     var respons = await http.get(url);
23
24     List responsDecode = jsonDecode(respons.body);
25     pendaftaranList.clear();
26     for (var element in responsDecode) {
27       pendaftaranList.add(Pendaftaran.fromJson(element));
28     }
29   }
30
31   @override
32   Widget build(BuildContext context) {
33     return Scaffold(
34       appBar: AppBar(title: Text("Data Pendaftaran")),
35       body: FutureBuilder(
36         future: ambilDataPendaftaran(),
37         builder: (context, snapshot) {
38           if (snapshot.connectionState == ConnectionState.waiting) {
39             return Center(child: CircularProgressIndicator());
40           } else {
41             if (pendaftaranList.length == 0) {
42               return Center(child: Text("Data tidak ditemukan"));
43             }
44           }
45           return ListView.builder(
46             itemCount: pendaftaranList.length,
47             itemBuilder:
48               (context, index) => Card(
49                 child: ListTile(
50                   title: Text(pendaftaranList[index].nama!),
51                   subtitle: Column(
52                     crossAxisAlignment: CrossAxisAlignment.start,
53                     children: [
54                       Text(pendaftaranList[index].email!),
55                       Text(pendaftaranList[index].noTelp!),
56                       Text(pendaftaranList[index].jenisKelamin!),
57                       Text(pendaftaranList[index].bahasa!),
58                       Text(pendaftaranList[index].agama!),
59                       Text(pendaftaranList[index].tanggalDaftar!),
60                       Text(pendaftaranList[index].jamDaftar!),
61                     ],
62                   ),
63                 ),
64               );
65         },
66       ),
67     );
68   }
69 }
```

```
63     trailing: Container(
64       width: 100,
65       child: Row(
66         children: [
67           IconButton(
68             onPressed: () async {
69               final result = await Navigator.push(
70                 context,
71                 MaterialPageRoute(
72                   builder:
73                     (context) => MyHomePage(
74                       pendaftaran: pendaftaranList[index],
75                     ),
76                   ),
77                 );
78               if (result == true) {
79                 setState(() {
80                   ambilDataPendaftaran();
81                 });
82               }
83             },
84             icon: Icon(Icons.edit),
85           ),
86           IconButton(
87             onPressed: () {
88               hapusData(index);
89             },
90             icon: Icon(Icons.delete),
91           ),
92           ],
93         ),
94       ),
95     ),
96   );
97 },
98 ),
99 ),
100 );
101 }
102 void hapusData(int index) {
103   showDialog(
104     context: context,
105     builder:
106       (context) => AlertDialog(
107         title: Text("Hapus Data"),
108         content: Text(
109           "Apakah anda yakin ingin menghapus data ${pendaftaranList[index].nama}",
110         ),
111         actions: [
112           ElevatedButton(
113             onPressed: () async {
114               setState(() {
115                 var url = Uri.parse(
116                   baseUrl + 'pendaftaran/${pendaftaranList[index].id!}',
117                 );
118                 var respons = http.delete(url);
119                 Navigator.pop(context);
120               });
121             },
122             child: Text("Ya"),
123           ),
124           ElevatedButton(
125             onPressed: () {
126               setState(() {
127                 Navigator.pop(context);
128               });
129             },
130             child: Text("Tidak"),
131           ),
132           ],
133         ),
134       ),
135     );
136   }
137 }
```

8. Model.dart yang mengatur struktur data

```
1 class Pendaftaran {
2   String? id;
3   String? nama;
4   String? email;
5   String? noTelpon;
6   String? jenisKelamin;
7   String? bahasa;
8   String? agama;
9   String? tanggalDaftar;
10  String? jamDaftar;
11
12  Pendaftaran({
13    this.id,
14    this.nama,
15    this.email,
16    this.noTelpon,
17    this.jenisKelamin,
18    this.bahasa,
19    this.agama,
20    this.tanggalDaftar,
21    this.jamDaftar,
22  });
23
24  Pendaftaran.fromJson(Map<String, dynamic> json) {
25    id = json['id'];
26    nama = json['nama'];
27    email = json['email'];
28    noTelpon = json['no_telpon'];
29    jenisKelamin = json['jenis_kelamin'];
30    bahasa = json['bahasa'];
31    agama = json['agama'];
32    tanggalDaftar = json['tanggal_daftar'];
33    jamDaftar = json['jam_daftar'];
34  }
35
36  Map<String, dynamic> toJson() {
37    final Map<String, dynamic> data = <String, dynamic>{};
38    data['id'] = id;
39    data['nama'] = nama;
40    data['email'] = email;
41    data['no_telpon'] = noTelpon;
42    data['jenis_kelamin'] = jenisKelamin;
43    data['bahasa'] = bahasa;
44    data['agama'] = agama;
45    data['tanggal_daftar'] = tanggalDaftar;
46    data['jam_daftar'] = jamDaftar;
47    return data;
48  }
49 }
50
51 // List<Pendaftaran> pendaftaranList = [];
52 }
```

9. Hasil kode program

The screenshots show a mobile application interface for a registration form and a list of registered users.

Screenshot 1: Latihan Formulir

This screen shows a form for user input:

- Email:
- Nomor Telpon:
- Gender selection: Pria, Wanita
- Kemampuan Berbahasa: Batak, Madura, Jepang, Korea
- Silahkan Pilih Agama:
- Tanggal Daftar:
- Jam Daftar:
- Action buttons: and

Screenshot 2: Data Pendaftaran

This screen lists registered users:

User	Email	Gender	Language	Date	Action
Fajar Ramadhan	fajar@gmail.com	08961	Pria [Indonesia, Jawa, Indonesia, Jawa]	2025-06-15 22:42:00	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
John Wick N	kohn@gmail.com	075248	Pria [Sunda, Inggris, Indonesia, Jawa]	2025-06-15 22:50:00	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Guthur abdel	gau@gmail.com				<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Screenshot 3: Confirmation Dialog

A confirmation dialog box is displayed:

Hapus Data

Apakah anda yakin ingin menghapus data Fajar saja?

Ya Tidak

The background shows the same list of users as Screenshot 2.