

**LAPORAN PRAKTIKUM**  
**MATA KULIAH PEMROGRAMAN MOBILE**



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

**Oleh :**

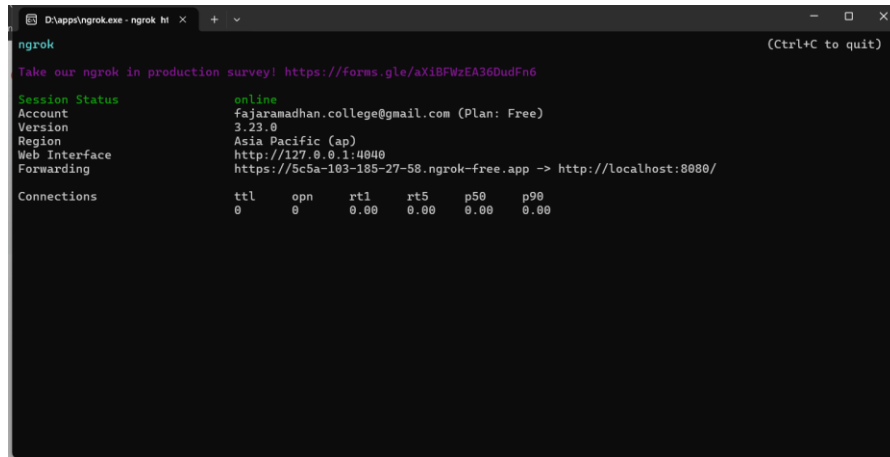
**NAMA: FAJAR RAMADHAN**

**NIM: 23SA31A065**

**PROGRAM STUDI TEKNOLOGI INFORMASI**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS AMIKOM**  
**PURWOKERTO**  
**2025**

## PRAKTIKUM 11 IMPLEMENTASI NGROK DAN API CALLS HTTP DI FLUTTER

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



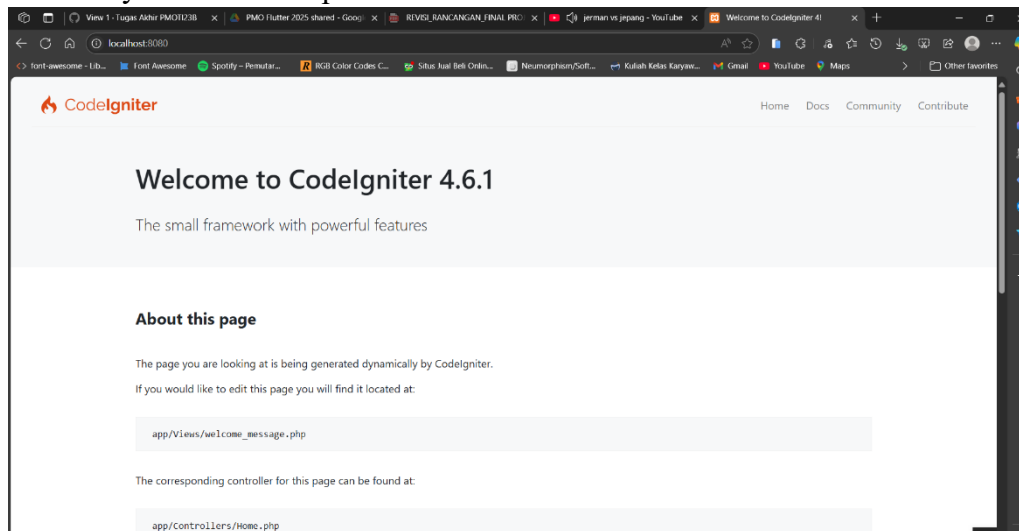
```
ngrok

Take our ngrok in production survey! https://forms.gle/aXiBFwzEA36DudFn6

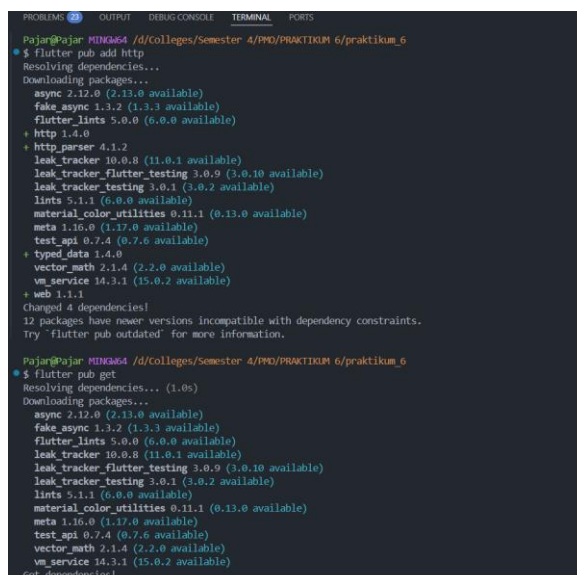
Session Status      online
Account             fajaramadhan.college@gmail.com (Plan: Free)
Version             3.23.0
Region              Asia Pacific (ap)
Web Interface        http://127.0.0.1:4040
Forwarding           https://5c5a-103-185-27-58.ngrok-free.app -> http://localhost:8080/

Connections
  ttl    opn    rt1    rt5    p50    p90
   0     0      0.00   0.00   0.00   0.00
```

2. Maka hasilnya bisa diakses publik



3. install http dengan "flutter pub add http" setelah itu "flutter pub get"

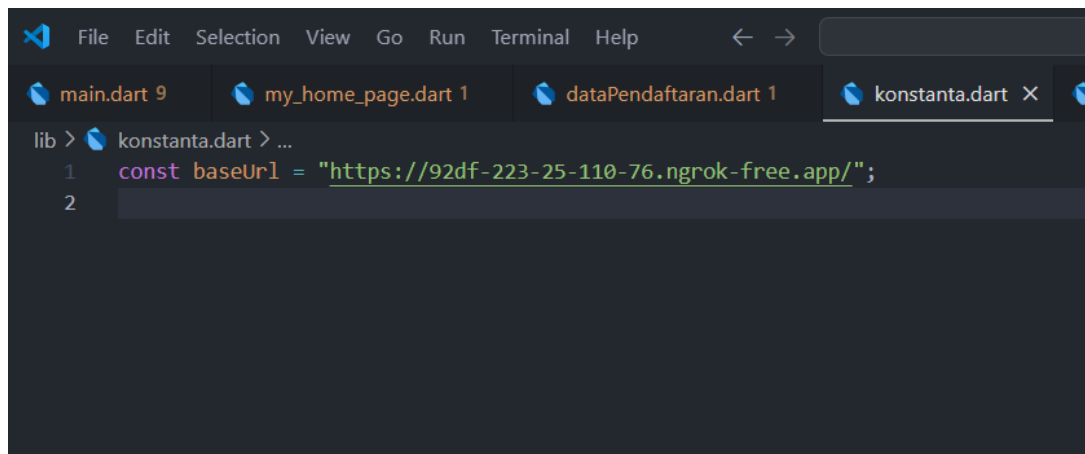


```
PROBLEMS OUTPUT DEBUG-CONSOLE TERMINAL PORTS

Pajir@Pajir-MINGW64 /d/Colleges/Semester 4/PMO/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
+ http_parser 4.1.2
+ 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.

Pajir@Pajir-MINGW64 /d/Colleges/Semester 4/PMO/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



The screenshot shows an IDE window with several tabs: main.dart 9, my\_home\_page.dart 1, dataPendaftaran.dart 1, and konstanta.dart X. The active tab is konstanta.dart, which contains the following code:

```
lib > konstanta.dart > ...  
1  const baseUrl = "https://92df-223-25-110-76.ngrok-free.app/";  
2
```

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



The screenshot shows an IDE window with a single tab for main.dart. The code in the file is as follows:

```
1  import 'package:flutter/material.dart';  
2  import 'package:praktikum_6/my_home_page.dart';  
3  
4  void main(List<String> args) {  
5    runApp(MyApp());  
6  }  
7  
8  class MyApp extends StatefulWidget {  
9    const MyApp({super.key});  
10  
11    @override  
12    State<MyApp> createState() => _MyAppState();  
13  }  
14  
15  class _MyAppState extends State<MyApp> {  
16    @override  
17    Widget build(BuildContext context) {  
18      return MaterialApp(  
19        theme: ThemeData(  
20          appBarTheme: AppBarTheme(  
21            backgroundColor: Colors.teal,  
22            foregroundColor: Colors.white,  
23          )),  
24          elevatedButtonTheme: ElevatedButtonThemeData(  
25            style: ButtonStyle(  
26              backgroundColor: WidgetStatePropertyAll(Colors.teal),  
27              foregroundColor: WidgetStatePropertyAll(Colors.white),  
28            )),  
29          ),  
30        ),  
31        home: MyHomePage(),  
32        debugShowCheckedModeBanner: false,  
33      );  
34    }  
35  }  
36
```

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 navigator 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 Formlir")),
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 Telp")),
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             ),

```

```

54     ),
55     TextField(
56       controller: noTelpController,
57       keyboardType: TextInputType.phone,
58       decoration: InputDecoration(label: Text("Nomor Telpn")),
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             noTelp: noTelpController.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 respons;
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
132           if (respons.statusCode == 200 ||
133               respons.statusCode == 201) {
134             if (widget.pendaftaran == null) {
135               Navigator.push(
136                 context,
137                 MaterialPageRoute(
138                   builder: (context) => Datapendaftaran(),
139                 ),
140               );
141             } else {
142               Navigator.pop(context, true);
143             }
144           } else {
145             Map<String, dynamic> responsDecode =
146               jsonDecode(respons.body) as Map<String, dynamic>;
147             ScaffoldMessenger.of(context).showSnackBar(
148               SnackBar(
149                 content: Text(
150                   responsDecode['messages'].toString().replaceAll(
151                     ",",
152                     "\n\n",
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/konstanta.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].noTelpn!),
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
103 void hapusData(int index) {
104     showDialog(
105         context: context,
106         builder:
107             (context) => AlertDialog(
108                 title: Text("Hapus Data"),
109                 content: Text(
110                     "Apakah anda yakin ingin menghapus data ${pendaftaranList[index].nama}?",
111                 ),
112                 actions: [
113                     ElevatedButton(
114                         onPressed: () async {
115                             setState(() {
116                                 var url = Uri.parse(
117                                     baseUrl + 'pendaftaran/${pendaftaranList[index].id!}',
118                                 );
119                                 var respons = http.delete(url);
120                                 Navigator.pop(context);
121                             });
122                         },
123                         child: Text("Ya"),
124                     ),
125                     ElevatedButton(
126                         onPressed: () {
127                             setState(() {
128                                 Navigator.pop(context);
129                             });
130                         },
131                         child: Text("Tidak"),
132                     ),
133                 ],
134             ),
135     );
136 }
137 }
138

```

## 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 = [];
```



## 9. Hasil kode program

23108RN04Y 22:56

### Latihan Formulir

Email

Nomor Telpun

☐ Pria ☐ Wanita

Kemampuan Berbahasa

☐ Batak ☐ Madura

☐ Jepang ☐ Korea

Silahkan Pilih Agama

Tanggal Daftar

Jam Daftar

Simpan Lihat Data

23108RN04Y 22:58

### Data Pendaftaran

Fajar Ramadhan  
fajar@gmail.com  
08961  
Pria  
[Indonesia, Jawa, Indonesia, Jawa]  
Kristen  
2025-06-15  
22:42:00

John Wick N  
kohn@gmail.com  
075248  
Pria  
[Sunda, Inggris, Indonesia, Jawa]  
ateis  
2025-06-15  
22:50:00

Guthur abdel  
gau@gmail.com

23108RN04Y 22:58

### Data Pendaftaran

Fajar saja  
fajar@gmail.com  
08961  
Pria  
[Batak, Indonesia]  
Ka  
20  
22

Jd  
kd  
07  
Pr  
[S  
Indonesia, Jawa]  
ateis  
2025-06-15  
22:50:00

Guthur abdel  
gau@gmail.com  
087248

**Hapus Data**

Apakah anda yakin ingin menghapus data Fajar saja?

Ya Tidak