

**TUGAS 09**  
**PRAK10 Frontend**



Praktikum Pemrograman Berbasis Web - A

Nama :

Adelia Nurlina Putri

4521210059

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PANCASILA**

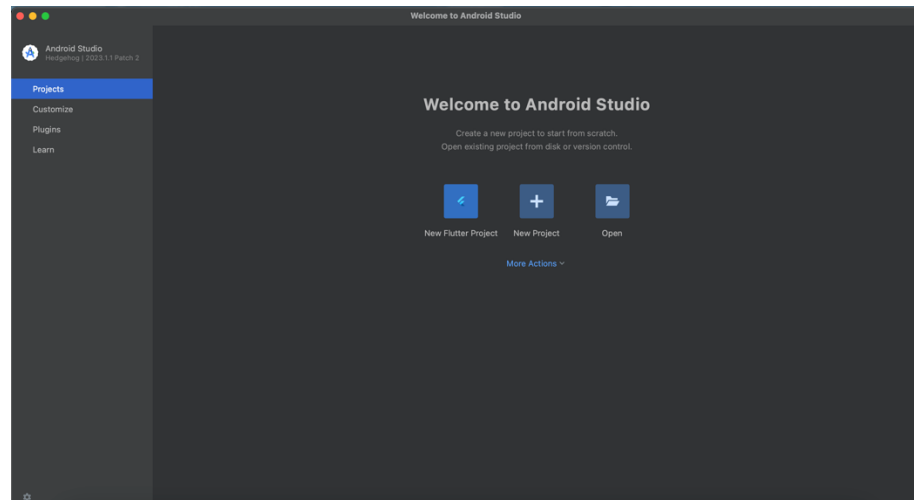
**JAKARTA**

**2024**

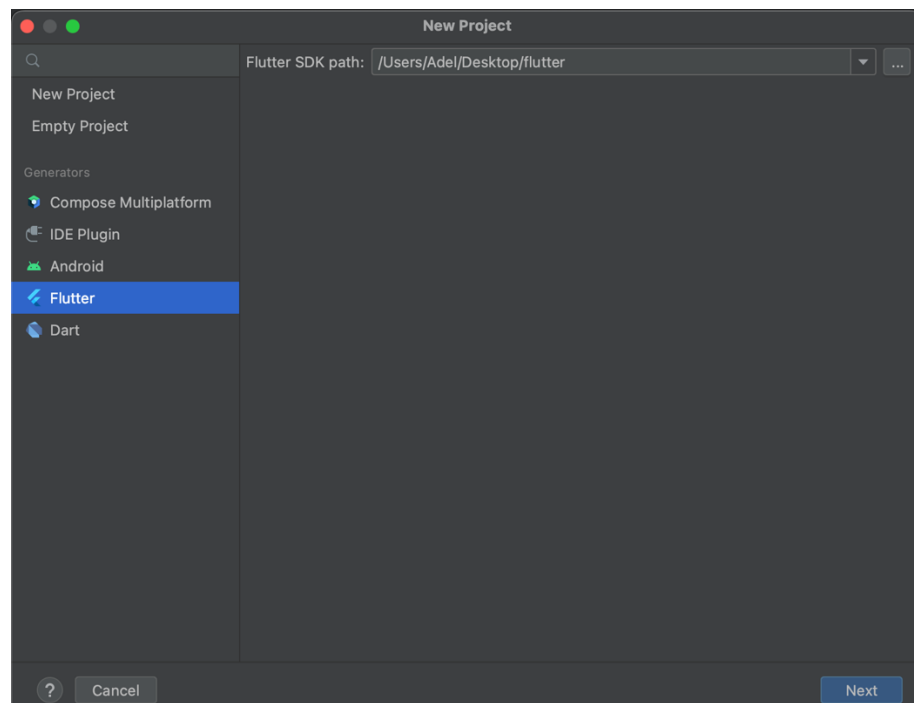
## A. Project prak10

### a. First Initialization

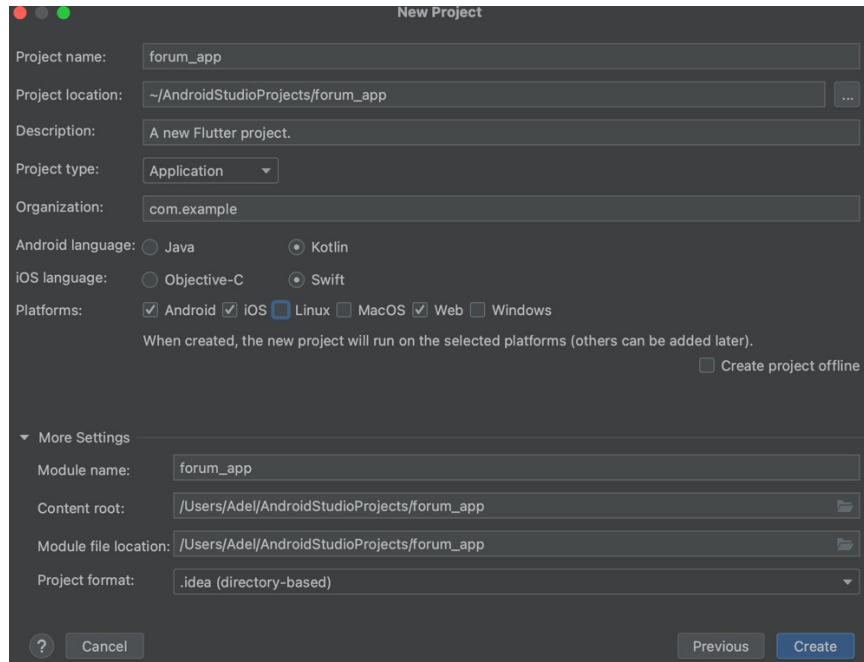
1. Akses web <https://docs.flutter.dev/ui/layout/tutorial>
2. Buka Android Studio



3. Pilih New Flutter Project -> Flutter -> Next



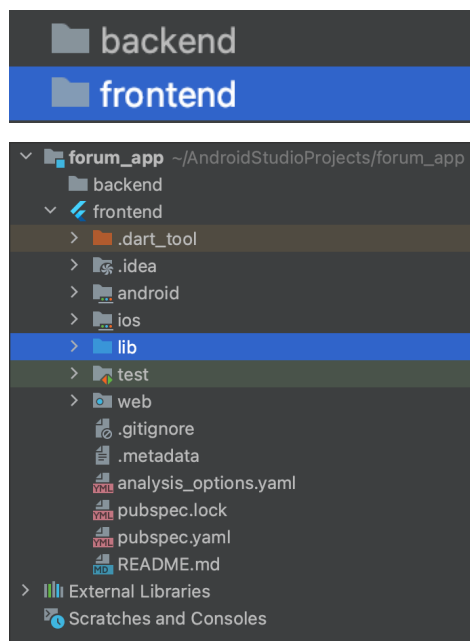
4. Isi form untuk nama project, tempat project, type, bahasa pemrograman, platforms dan lainnya -> Create



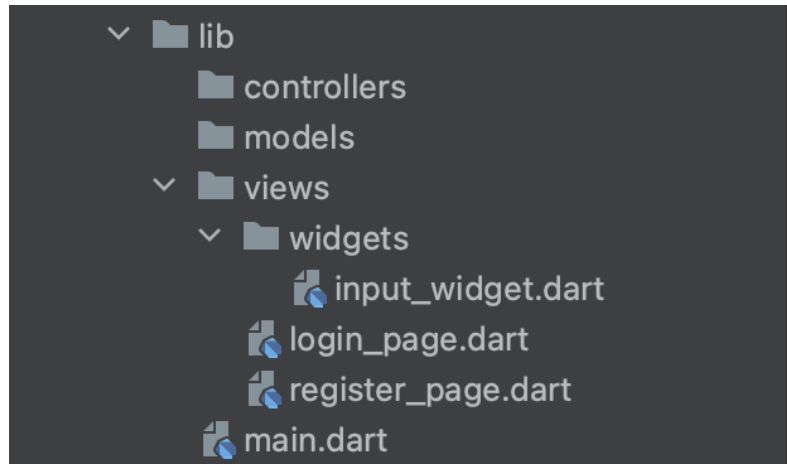
5.

## b. Membuat App

1. Buat folder backend & frontend. Dan masukkan seluruh file project ke frontend folder



## 2. Buat folder dan file dengan susunan seperti berikut



## 3. Add package

```
$ flutter pub add google_fonts
```

```
$ flutter pub add get
```

```

Adel@anps frontend % flutter pub add google_fonts
Resolving dependencies...
+ crypto 3.0.3
+ ffi 2.1.2
  flutter_lints 3.0.2 (4.0.0 available)
+ google_fonts 6.2.0 (6.2.1 available)
+ http 1.2.1
+ http_parser 4.0.2
  leak_tracker 10.0.0 (10.0.5 available)
  leak_tracker_flutter_testing 2.0.1 (3.0.5 available)
  leak_tracker_testing 2.0.1 (3.0.1 available)
  lints 3.0.0 (4.0.0 available)
  material_color_utilities 0.8.0 (0.11.1 available)
  meta 1.11.0 (1.15.0 available)
+ path_provider 2.1.3
+ path_provider_android 2.2.4 (2.2.5 available)
+ path_provider_foundation 2.4.0
+ path_provider_linux 2.2.1
+ path_provider_platform_interface 2.1.2
+ path_provider_windows 2.2.1
+ platform 3.1.4
  
```

```

Adel@anps frontend % flutter pub add get
Resolving dependencies...
  flutter_lints 3.0.2 (4.0.0 available)
+ get 4.6.6
  google_fonts 6.2.0 (6.2.1 available)
  leak_tracker 10.0.0 (10.0.5 available)
  leak_tracker_flutter_testing 2.0.1 (3.0.5 available)
  leak_tracker_testing 2.0.1 (3.0.1 available)
  lints 3.0.0 (4.0.0 available)
  material_color_utilities 0.8.0 (0.11.1 available)
  meta 1.11.0 (1.15.0 available)
  path_provider_android 2.2.4 (2.2.5 available)
  test_api 0.6.1 (0.7.2 available)
  vm_service 13.0.0 (14.2.3 available)
  win32 5.5.0 (5.5.1 available)
Changed 1 dependency!
12 packages have newer versions incompatible with dependency constraints.
Try 'flutter pub outdated' for more information.
Adel@anps frontend %
  
```

#### 4. File main.dart

```
import 'package:flutter/material.dart';
import 'views/login_page.dart';

void main() {
  runApp(const MyApp());
}

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

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Forum Application',
      home: const LoginPage(),
    );
  }
}
```

#### 5. File input\_widget.dart

```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';

class InputWidget extends StatelessWidget {
  const InputWidget({
    super.key,
    required this.hintText,
    required this.controller,
    required this.obscureText,
  });

  final String hintText;
  final TextEditingController controller;
  final bool obscureText;

  @override
  Widget build(BuildContext context) {
    return Container(
      width: double.infinity,
      decoration: BoxDecoration(
        color: Colors.grey[200],
        borderRadius: BorderRadius.circular(10),
      ),
      child: TextField(
        obscureText: obscureText,
        controller: controller,
        decoration: InputDecoration(
          border: InputBorder.none,
          hintText: hintText,
          hintStyle: GoogleFonts.poppins(),
        ),
      ),
    );
  }
}
```

```

        contentPadding: const
EdgeInsets.symmetric(horizontal: 20),
      ),
    ),
  );
}
}

```

## 6. File login\_page.dart

```

import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';

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

  @override
  State<LoginPage> createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  final TextEditingController _emailController =
    TextEditingController();
  final TextEditingController _passwordController =
    TextEditingController();

  @override
  Widget build(BuildContext context) {
    var size = MediaQuery.of(context).size.width;
    return Scaffold(
      body: Center(
        child: Padding(
          padding: const
EdgeInsets.symmetric(horizontal: 20.0),
          child: Column(
            mainAxisAlignment:
MainAxisAlignment.center,
            children: [
              Text(
                'Login Page',
                style: GoogleFonts.poppins(
                  fontSize: size * 0.080,
                ),
              ),
              const SizedBox(
                height: 30,
              ),
              InputWidget(
                hintText: 'Email',
                obscureText: false,
                controller: _emailController,
              ),
              const SizedBox(
                height: 20,
              ),
            ],
          ),
        ),
      ),
    );
  }
}

```

```

    ),
    InputWidget(
      hintText: 'Password',
      obscureText: true,
      controller: _passwordController,
    ),
    const SizedBox(
      height: 30,
    ),
    ElevatedButton(
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.black,
        elevation: 0,
        padding: const
EdgeInsets.symmetric(horizontal: 50, vertical: 15),
      ),
      onPressed: () {
        // Add your login logic here
      },
      child: Text(
        'Login',
        style: GoogleFonts.poppins(
          fontSize: size * 0.040,
        ),
      ),
    ),
  ),
),
],
),
),
),
);
}
}

class InputWidget extends StatelessWidget {
  final String hintText;
  final bool obscureText;
  final TextEditingController controller;

  const InputWidget({
    required this.hintText,
    required this.obscureText,
    required this.controller,
  });

  @override
  Widget build(BuildContext context) {
    return TextField(
      controller: controller,
      obscureText: obscureText,
      decoration: InputDecoration(
        hintText: hintText,
        border: OutlineInputBorder(),
      ),
    );
  }
}

```

```

    }
  }
}

```

## 7. File add\_screen.dart

```

import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:get/get.dart';
import 'register_page.dart';
import 'widgets/input_widget.dart';

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

  @override
  State<LoginPage> createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  final TextEditingController _emailController =
    TextEditingController();
  final TextEditingController _passwordController =
    TextEditingController();

  @override
  Widget build(BuildContext context) {
    var size = MediaQuery.of(context).size.width;
    return Scaffold(
      body: Center(
        child: Padding(
          padding: const
EdgeInsets.symmetric(horizontal: 20.0),
          child: Column(
            mainAxisAlignment:
MainAxisAlignment.center,
            children: [
              Text(
                'Login Page',
                style: GoogleFonts.poppins(
                  fontSize: size * 0.080,
                ),
              ),
              const SizedBox(
                height: 30,
              ),
              InputWidget(
                hintText: 'Email',
                obscureText: false,
                controller: _emailController,
              ),
              const SizedBox(
                height: 20,
              ),
              InputWidget(

```





```

    @override
    State<RegisterPage> createState() =>
    _RegisterPageState();
  }

  class _RegisterPageState extends State<RegisterPage> {
    final TextEditingController _emailController =
    TextEditingController();
    final TextEditingController _passwordController =
    TextEditingController();
    final TextEditingController _nameController =
    TextEditingController();
    final TextEditingController _usernameController =
    TextEditingController();

    @override
    Widget build(BuildContext context) {
      var size = MediaQuery.of(context).size.width;
      return Scaffold(
        body: Center(
          child: Padding(
            padding: const
            EdgeInsets.symmetric(horizontal: 20.0),
            child: Column(
              mainAxisAlignment:
              MainAxisAlignment.center,
              children: [
                Text(
                  'Register Page',
                  style: GoogleFonts.poppins(
                    fontSize: size * 0.080,
                  ),
                ),
                const SizedBox(
                  height: 30,
                ),
                InputWidget(
                  hintText: 'Name',
                  obscureText: false,
                  controller: _nameController,
                ),
                const SizedBox(
                  height: 20,
                ),
                InputWidget(
                  hintText: 'Username',
                  obscureText: false,
                  controller: _usernameController,
                ),
                const SizedBox(
                  height: 30,
                ),
                InputWidget(
                  hintText: 'Email',
                  obscureText: false,

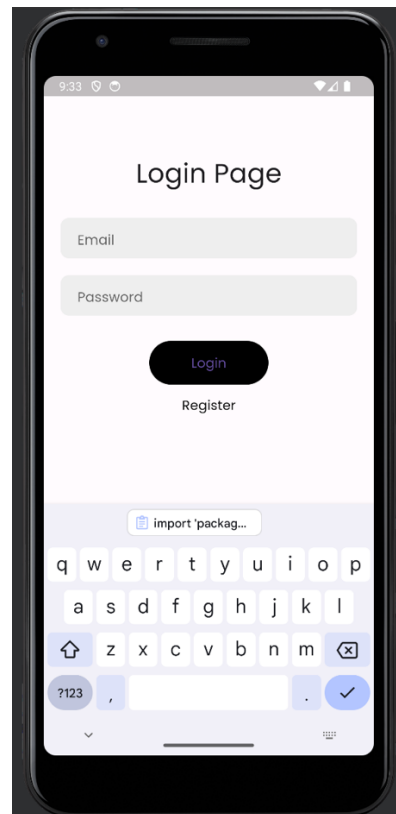
```

```

        controller: _emailController,
      ),
      const SizedBox(
        height: 20,
      ),
      InputWidget(
        hintText: 'Password',
        obscureText: true,
        controller: _passwordController,
      ),
      const SizedBox(
        height: 30,
      ),
      ElevatedButton(
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.black,
          elevation: 0,
          padding: const
EdgeInsets.symmetric(horizontal: 50, vertical: 15),
        ),
        onPressed: () {
          // Add your register logic here
        },
        child: Text(
          'Register',
          style: GoogleFonts.poppins(
            fontSize: size * 0.040,
          ),
        ),
      ),
      TextButton(
        onPressed: () {
          Get.to(() => const LoginPage());
        },
        child: Text(
          'Login',
          style: GoogleFonts.poppins(
            fontSize: size * 0.040,
            color: Colors.black,
          ),
        ),
      ),
      const SizedBox(
        height: 20,
      ),
    ],
  ),
),
),
),
);
}
}

```

Kode di atas akan membuat tampilan seperti berikut



c. **Push Code to GitHub** [https://github.com/adelianurlinap/Prak\\_PBM.git](https://github.com/adelianurlinap/Prak_PBM.git)

1. Clone Repository yang telah dibuat :

```
git clone https://github.com/adelianurlinap/Prak_PBM.git
```

2. Add dan commit

```
git add .  
git commit -m "Tugas9"
```

3. Push code

```
git push
```

## B. Kesimpulan

Pada tugas 9 ini, telah dilakukan praktek untuk membuat frontend register dan login

**C. Referensi**

modul prak10