**TUGAS 02**

**WIDGET FLUTTER**



Praktikum Pemrograman Berbasis Web - A

Nama :

Adelia Nurlina Putri 4521210059

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

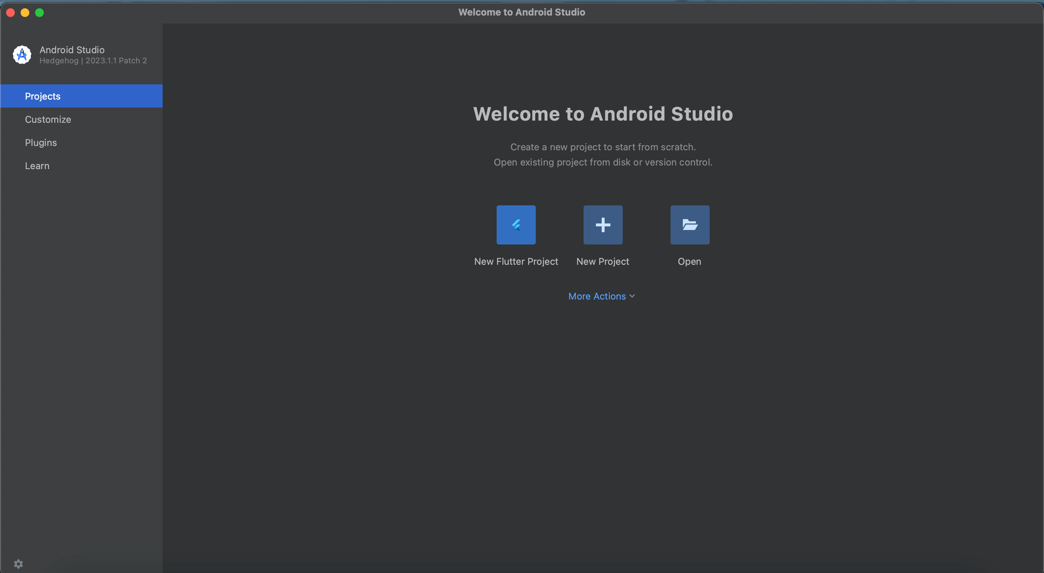
**FAKULTAS TEKNIK**

**UNIVERSITAS PANCASILA**

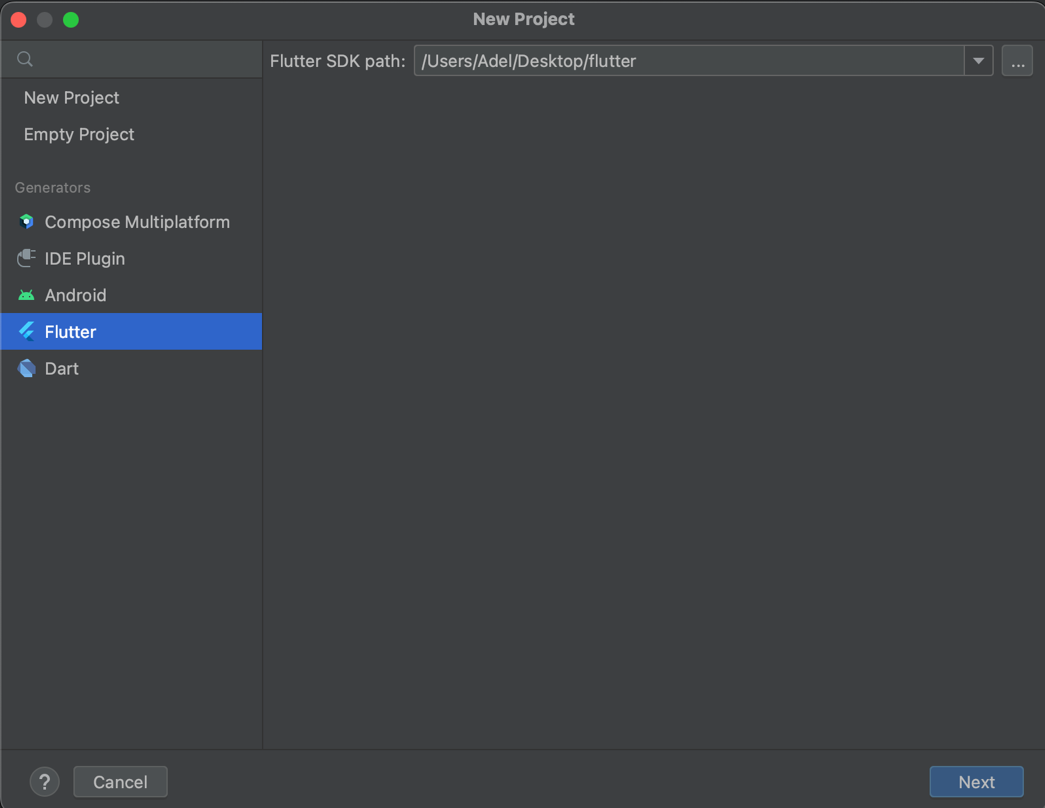
**JAKARTA**

**2024**

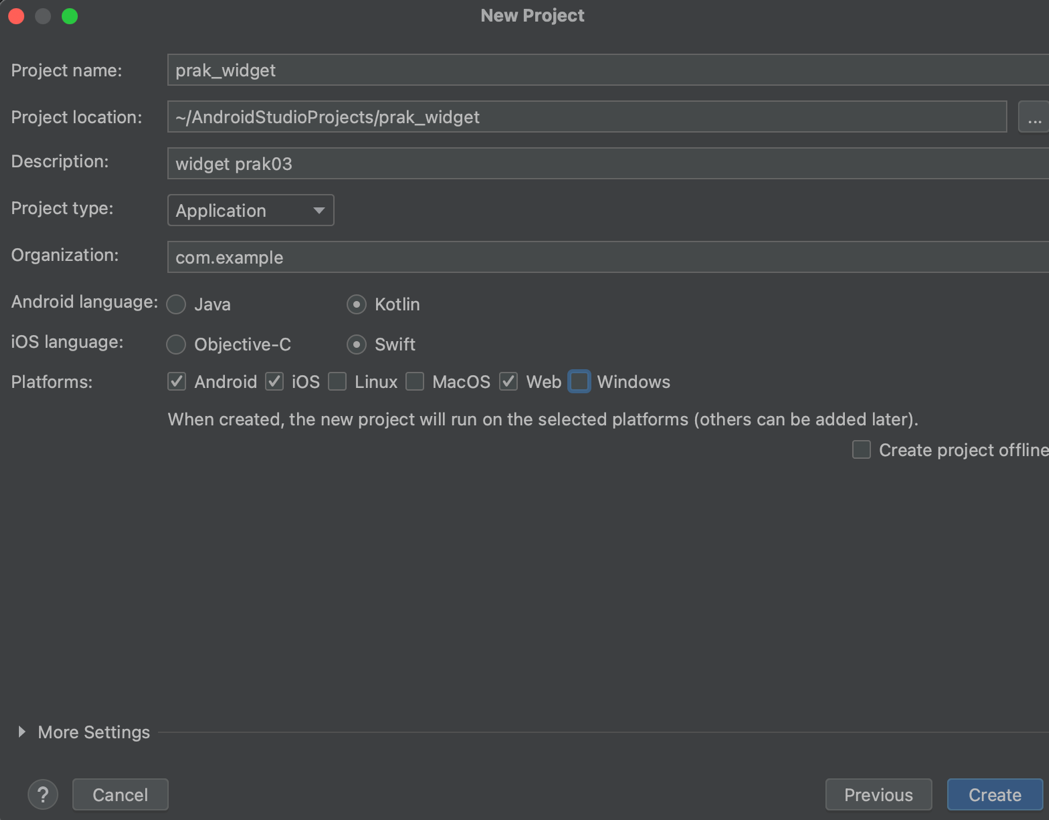
1. **Build a Flutter layout**
2. Akses web <https://docs.flutter.dev/ui/layout/tutorial>
3. Buka Android Studio



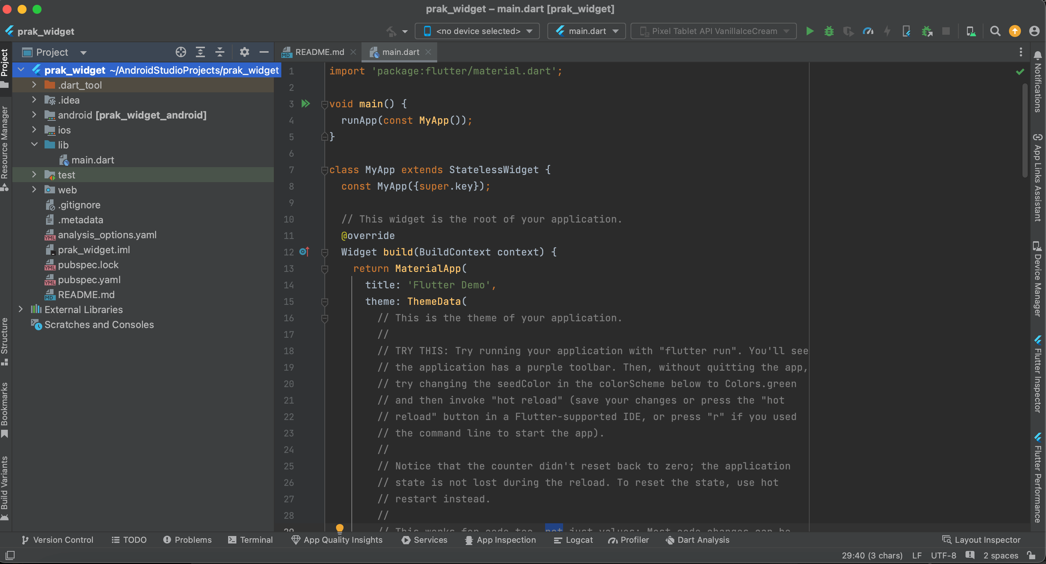
1. Pilih New Flutter Project -> Flutter -> Next



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



1. Tampilan Awal



1. Create the app base code

|  |
| --- |
| Import 'package:flutter/material.dart';  void main() => runApp(const MyApp());  class MyApp extends StatelessWidget {  const MyApp({super.key});  @override  Widget build(BuildContext context) {  const String appTitle = 'Flutter layout demo';  return MaterialApp(  title: appTitle,  home: Scaffold(  appBar: AppBar(  title: const Text(appTitle),  ),  body: const Center(  child: Text('Hello World'),  ),  ),  );  }  } |

1. Add the Title Section
2. Add the TitleSection Widget

|  |
| --- |
| class TitleSection extends StatelessWidget {  const TitleSection({  super.key,  required this.name,  required this.location,  });  final String name;  final String location;  @override  Widget build(BuildContext context) {  return Padding(  padding: const EdgeInsets.all(32),  child: Row(  children: [  Expanded(  /\*1\*/  child: Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  /\*2\*/  Padding(  padding: const EdgeInsets.only(bottom: 8),  child: Text(  name,  style: const TextStyle(  fontWeight: FontWeight.bold,  ),  ),  ),  Text(  location,  style: TextStyle(  color: Colors.grey[500],  ),  ),  ],  ),  ),  /\*3\*/  Icon(  Icons.star,  color: Colors.red[500],  ),  const Text('41'),  ],  ),  );  }  } |

1. Change the app body to a scrolling view

|  |
| --- |
| body: const SingleChildScrollView(  child: Column(  children: [ |

1. Update the app to display the section

|  |
| --- |
| TitleSection(  name: 'Oeschinen Lake Campground',  location: 'Kandersteg, Switzerland',  ), |

1. Add the Button Section
2. Add the ButtonSection Widget

|  |
| --- |
| class ButtonSection extends StatelessWidget {  const ButtonSection({super.key});  @override  Widget build(BuildContext context) {  final Color color = Theme.of(context).primaryColor;  // ···  }  } |

1. Create a widget to make buttons

|  |
| --- |
| class ButtonSection extends StatelessWidget {  const ButtonSection({super.key});  // ···  }  class ButtonWithText extends StatelessWidget {  const ButtonWithText({  super.key,  required this.color,  required this.icon,  required this.label,  });  final Color color;  final IconData icon;  final String label;  @override  Widget build(BuildContext context) {  return Column(  mainAxisSize: MainAxisSize.min,  mainAxisAlignment: MainAxisAlignment.center,  children: [  Icon(icon, color: color),  Padding(  padding: const EdgeInsets.only(top: 8),  child: Text(  label,  style: TextStyle(  fontSize: 12,  fontWeight: FontWeight.w400,  color: color,  ),  ),  ),  ],  );  } |

1. Position the buttons with a Row widget

|  |
| --- |
| class ButtonSection extends StatelessWidget {  const ButtonSection({super.key});  @override  Widget build(BuildContext context) {  final Color color = Theme.of(context).primaryColor;  return SizedBox(  child: Row(  mainAxisAlignment: MainAxisAlignment.spaceEvenly,  children: [  ButtonWithText(  color: color,  icon: Icons.call,  label: 'CALL',  ),  ButtonWithText(  color: color,  icon: Icons.near\_me,  label: 'ROUTE',  ),  ButtonWithText(  color: color,  icon: Icons.share,  label: 'SHARE',  ),  ],  ),  );  }  }  class ButtonWithText extends StatelessWidget {  const ButtonWithText({  super.key,  required this.color,  required this.icon,  required this.label,  });  final Color color;  final IconData icon;  final String label;  @override  Widget build(BuildContext context) {  return Column(  // ···  );  }  } |

1. Update the app to display the button section

|  |
| --- |
| TitleSection(  name: 'Oeschinen Lake Campground',  location: 'Kandersteg, Switzerland',  ),  ButtonSection(), |

1. Add the Text Section
2. Add the TextSection widget

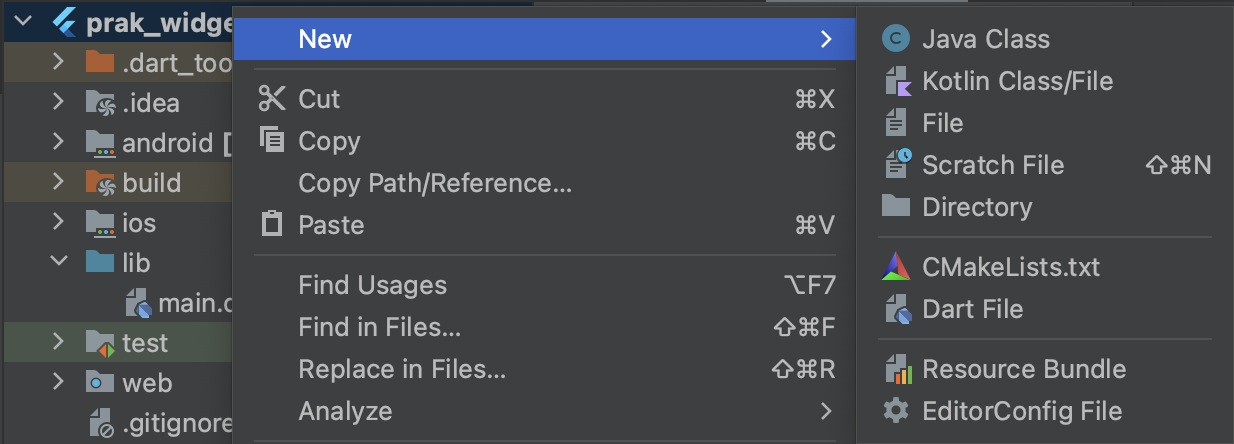
|  |
| --- |
| class TextSection extends StatelessWidget {  const TextSection({  super.key,  required this.description,  });  final String description;  @override  Widget build(BuildContext context) {  return Padding(  padding: const EdgeInsets.all(32),  child: Text(  description,  softWrap: true,  ),  );  }  } |

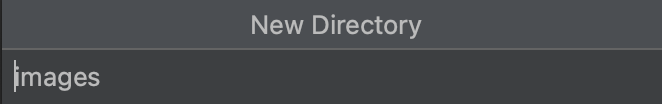
1. Update the app to display the text section

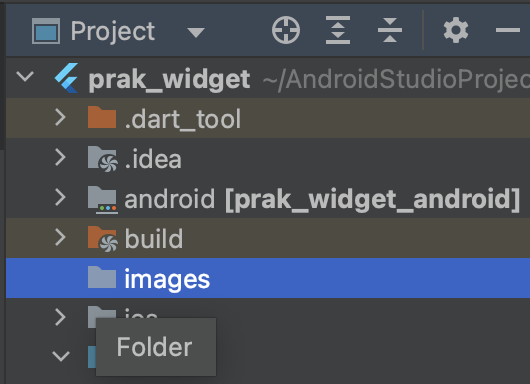
|  |
| --- |
| TextSection(  description:  'Lake Oeschinen lies at the foot of the Blüemlisalp in the '  'Bernese Alps. Situated 1,578 meters above sea level, it '  'is one of the larger Alpine Lakes. A gondola ride from '  'Kandersteg, followed by a half-hour walk through pastures '  'and pine forest, leads you to the lake, which warms to 20 '  'degrees Celsius in the summer. Activities enjoyed here '  'include rowing, and riding the summer toboggan run.',  ), |

1. Add the image section
2. Download image
3. Create Directory in project folder

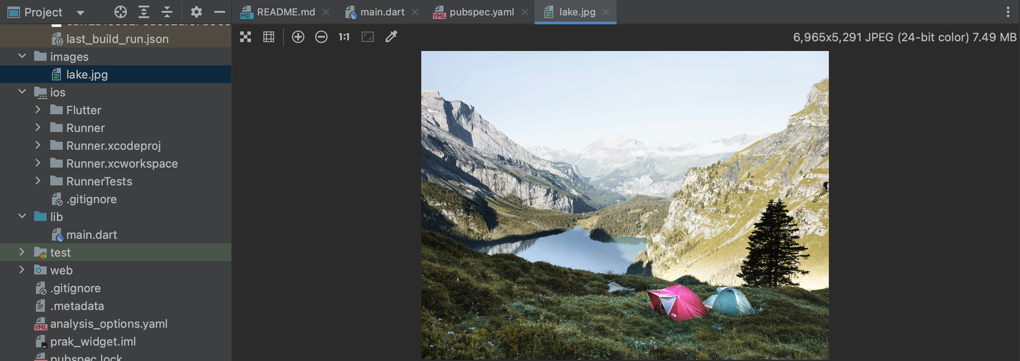
|  |
| --- |
| Klik kanan project -> New -> Directory |

****

****

****

1. Copy image to folder

****

1. Create the ImageSction widget

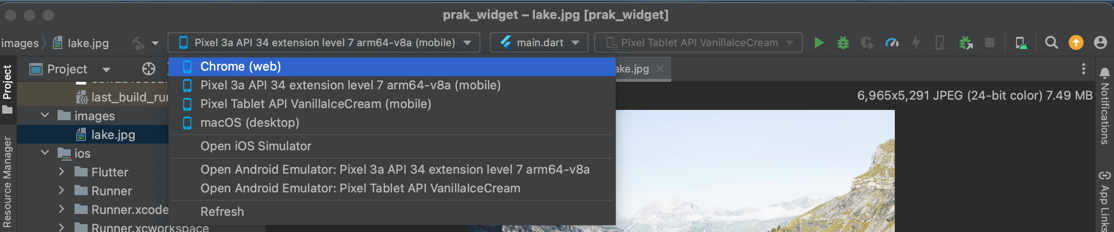
|  |
| --- |
| class ImageSection extends StatelessWidget {  const ImageSection({super.key, required this.image});  final String image;  @override  Widget build(BuildContext context) {  return Image.asset(  image,  width: 600,  height: 240,  fit: BoxFit.cover,  );  }  } |

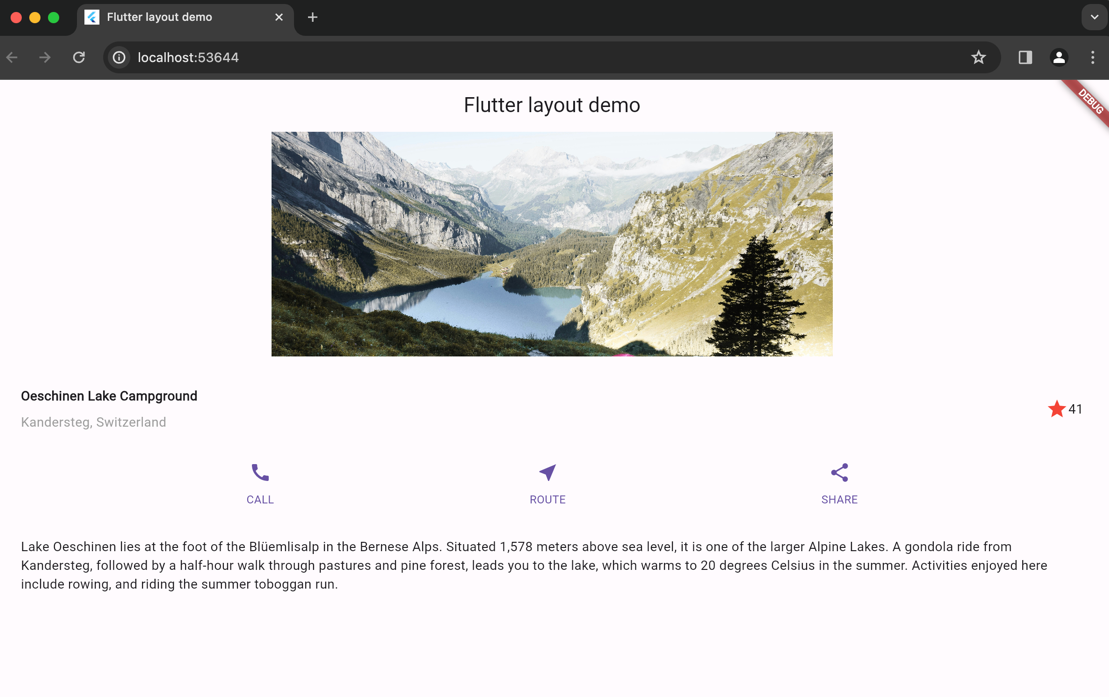
1. Update the app to display the image section

|  |
| --- |
| ImageSection(  image: 'images/lake.jpg',  ), |

1. Run App

|  |
| --- |
| Pilih simulator -> Klik Icon Run |



****

1. Push Code to GitHub <https://github.com/adelianurlinap/Prak_PBM.git>
2. Clone Repository yang telah dibuat :

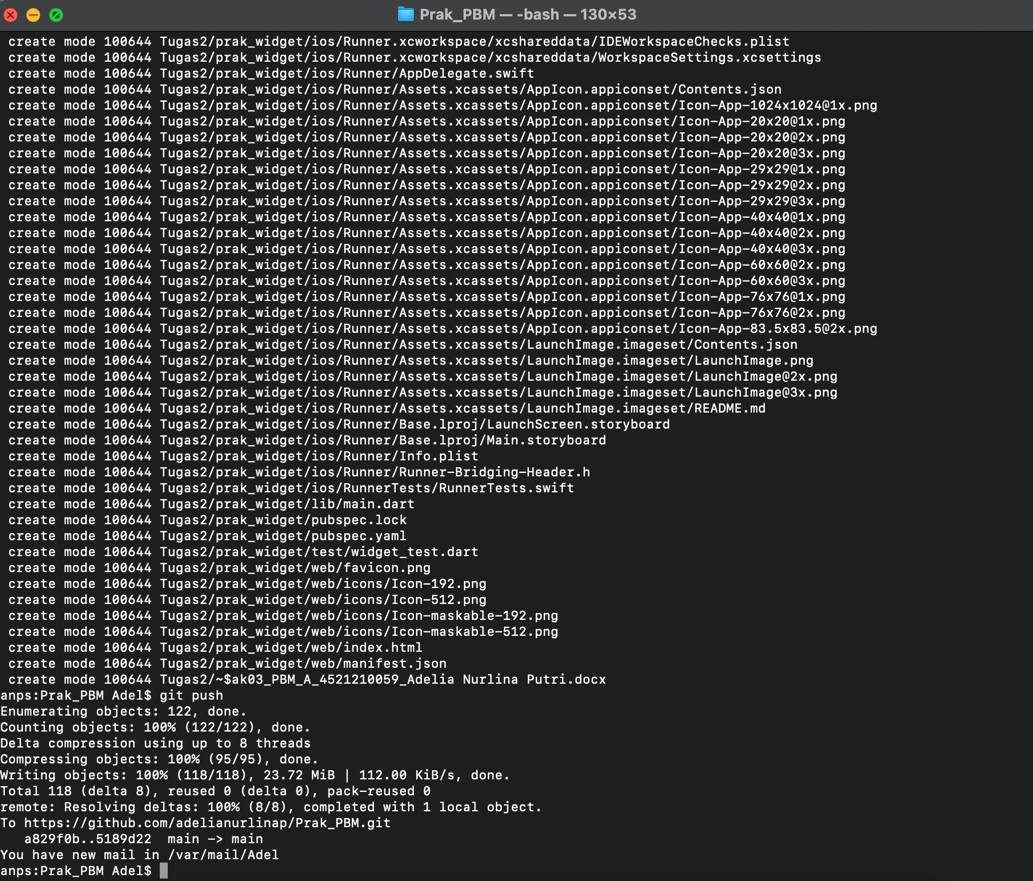
|  |
| --- |
| git clone https://github.com/adelianurlinap/Prak\_PBM.git |

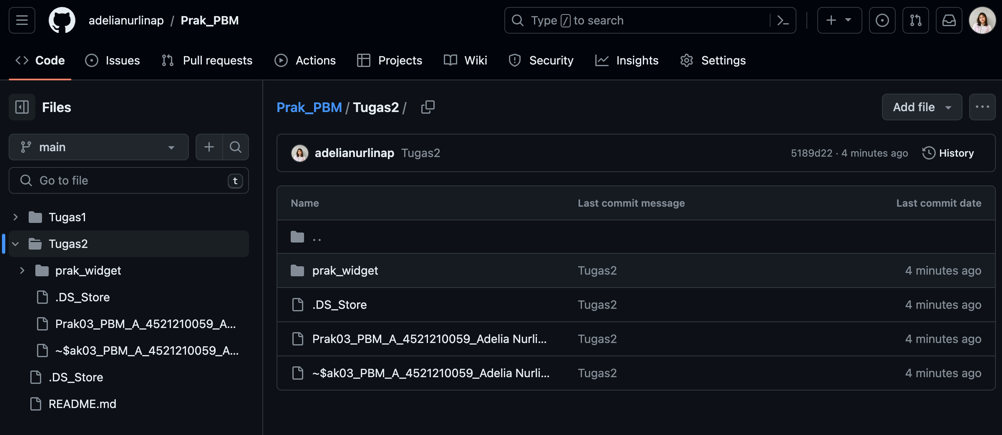
1. Add dan commit

|  |
| --- |
| git add .  git commit -m "Tugas2" |

1. Push code

|  |
| --- |
| git push |





1. Kesimpulan

Pada tugas 2 ini, telah dilakukan praktek untuk membuat widget flutter dasar. Aplikasi berbentuk 1 page dengan beberapa button, text, dan 1 gambar. Terakhir yaitu melakukan push code ke GitHub.

1. Referensi

* <https://docs.flutter.dev/ui/layout/tutorial>