TUGAS 01

APLIKASI FLUTTER

Praktikum Pemrograman Berbasis Web - A

Nama :

Adelia Nurlina Putri 4521210059

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS TEKNIK

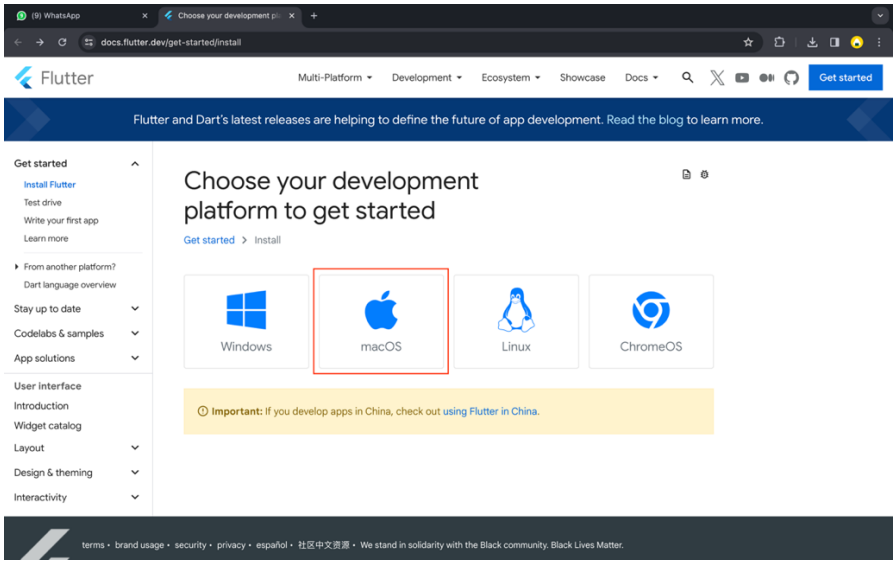
UNIVERSITAS PANCASILA

JAKARTA

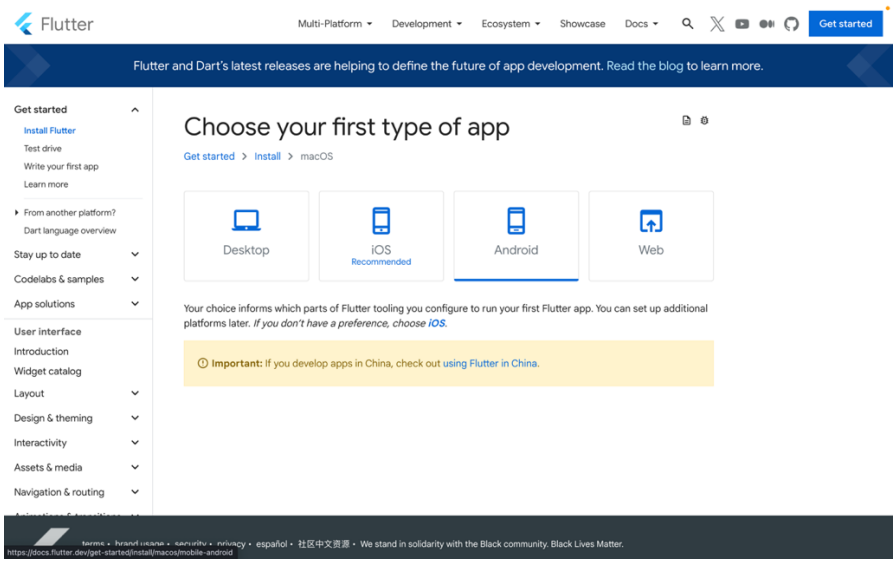
2024

A. Instalasi Flutter SDK

1. Akses web https://docs.flutter.dev/get-started/install
2. Pilih OS yang digunakan untuk penggunaan Flutter : macOS



3. Pilih tipe App : Android



4. Cek default shell (Flutter support MacOS Catalina default shell menggunakan zsh)

```
dscl . -read ~/ UserShell
```

```
anps:~ Adel$ dscl . -read ~/ UserShell
UserShell: /bin/zsh
anps:~ Adel$
```

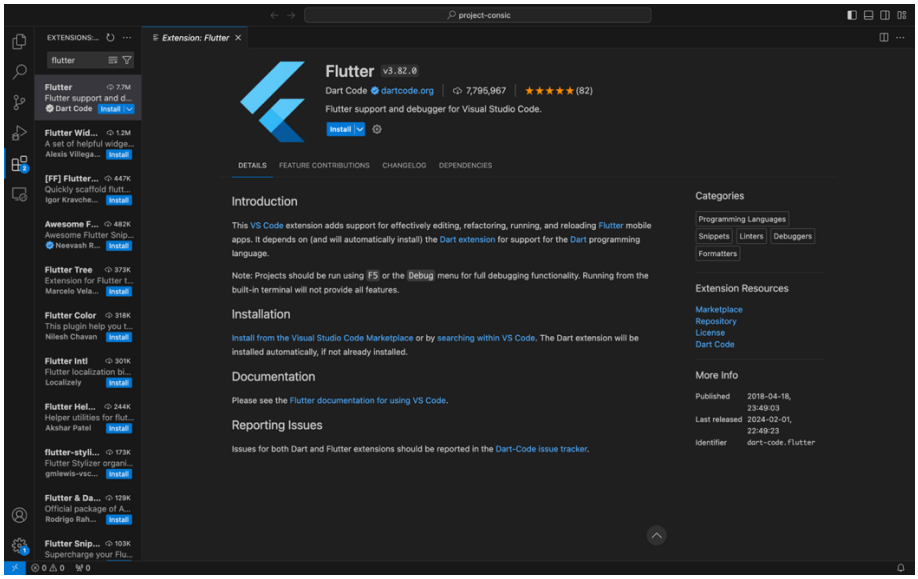
5. Beberapa komponen Flutter memerlukan proses Rosetta 2 di Mac Apple Silicon. Untuk menjalankan semua komponen Flutter di Apple Sillicon, maka perlu instal Rosetta 2.

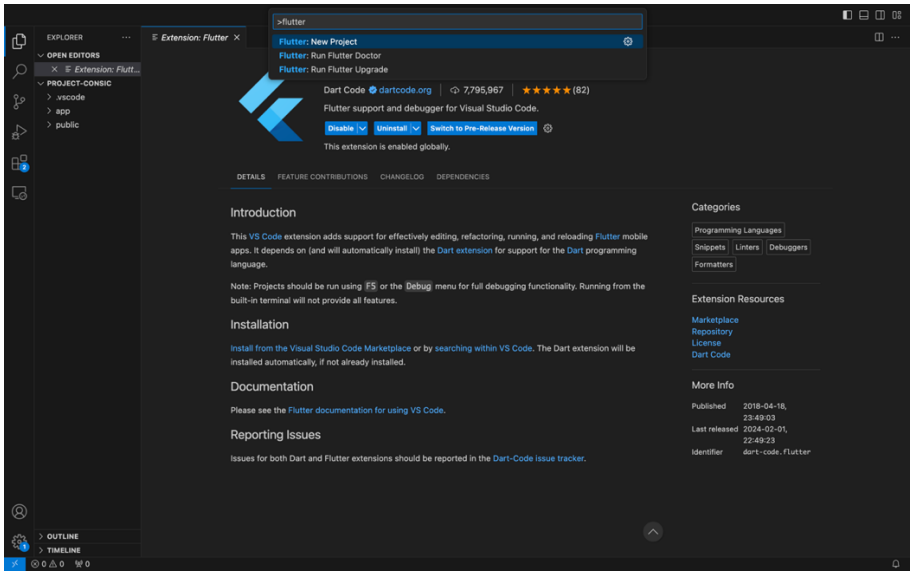
```
sudo softwareupdate --install-rosetta --agree-to-license
```

```
anps:~ Adel$ sudo softwareupdate --install-rosetta --agree-to-license
By using the agreetolicense option, you are agreeing that you have run this tool with the license only option and have read and agreed to the terms.
If you do not agree, press CTRL-C and cancel this process immediately.

2024-02-26 11:28:10.039 softwareupdate[2058:28100] Package Authoring Error: 012-92132: Package reference com.apple.pkg.RosettaUpdateAuto is missing installKBytes attribute
Install of Rosetta 2 finished successfully
anps:~ Adel$
```

6. Download & Install Visual Studio Code (Installed)
7. Install Flutter Extension

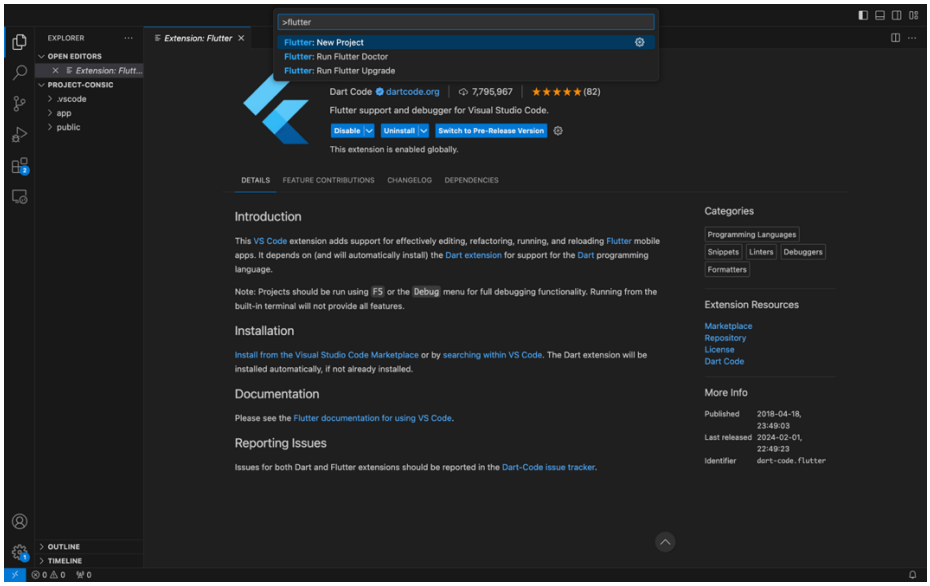
```
Buka VSCode -> Marketplace -> Ketik Flutter -> Install
```



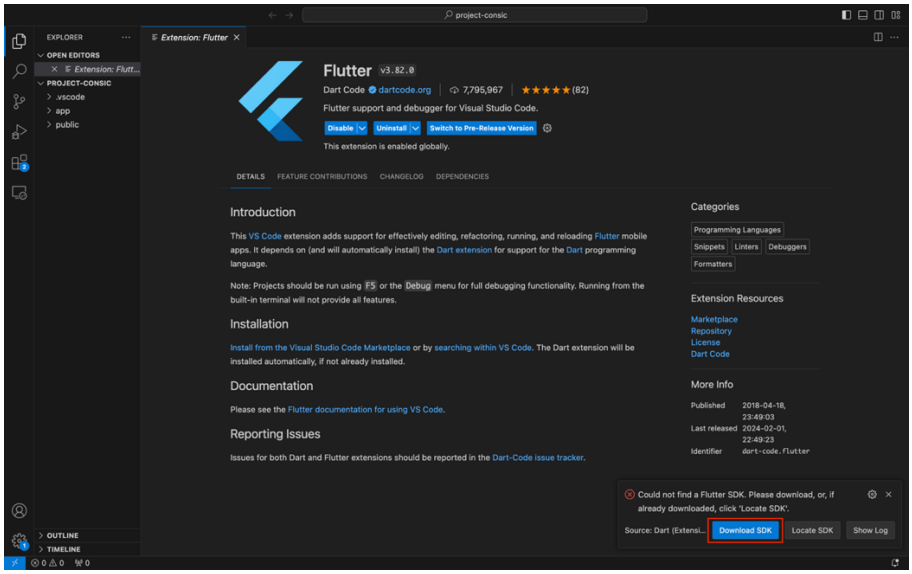


8. Install Flutter SDK

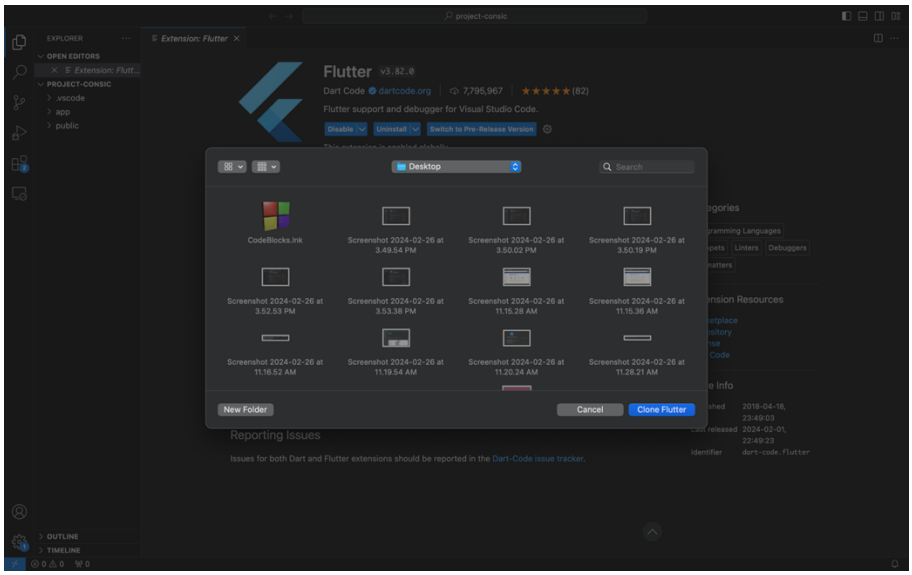
Buka VSCode -> Command+Shift+P -> Ketik Flutter -> Pilih Flutter: New Project



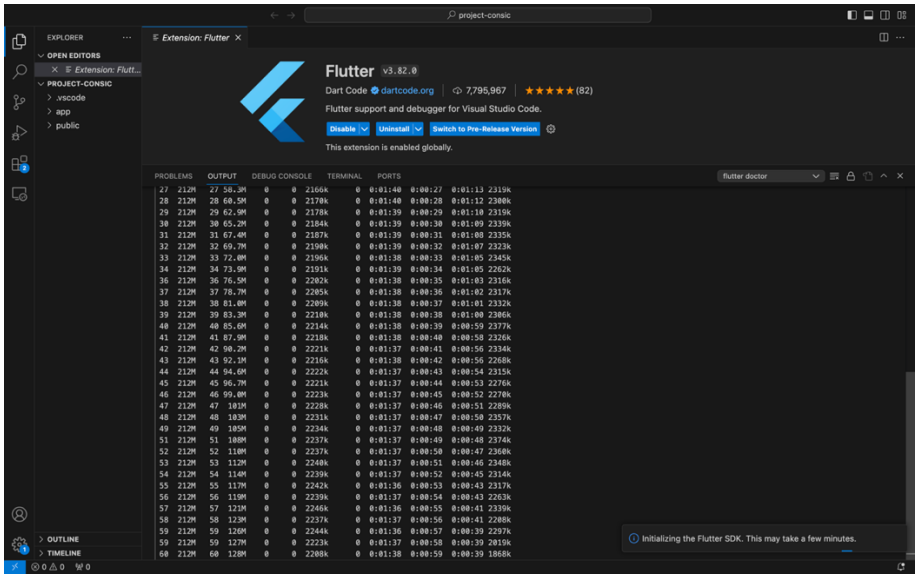
9. Pilih Download SDK



10. Pilih tempat untuk menyimpan file SDK



11. Downloading dan Initializing proses



12. Mengaktifkan Flutter pada seluruh terminal

```
sudo nano ~/.zshrc

---

export PATH="$PATH:/Users/Adel/Desktop/flutter/bin"

---

echo $ PATH
```

B. Validasi Komponen Flutter

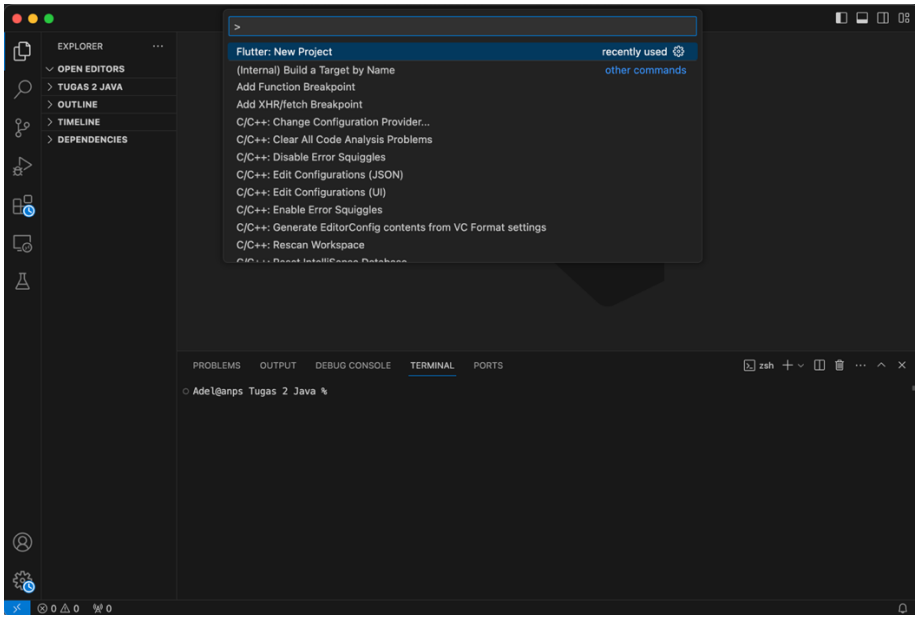
```
$ flutter doctor
```

```
anps:~ Adel$ flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.19.1, on macOS 13.0 22A380 darwin-arm64, locale en-US)
[✓] Android toolchain - develop for Android devices (Android SDK version 34.0.0)
[✓] Xcode - develop for iOS and macOS (Xcode 14.3.1)
[✓] Chrome - develop for the web
[✓] Android Studio (version 2023.1)
[✓] VS Code (version 1.86.2)
[✓] Connected device (2 available)
[✓] Network resources

• No issues found!
anps:~ Adel$
```

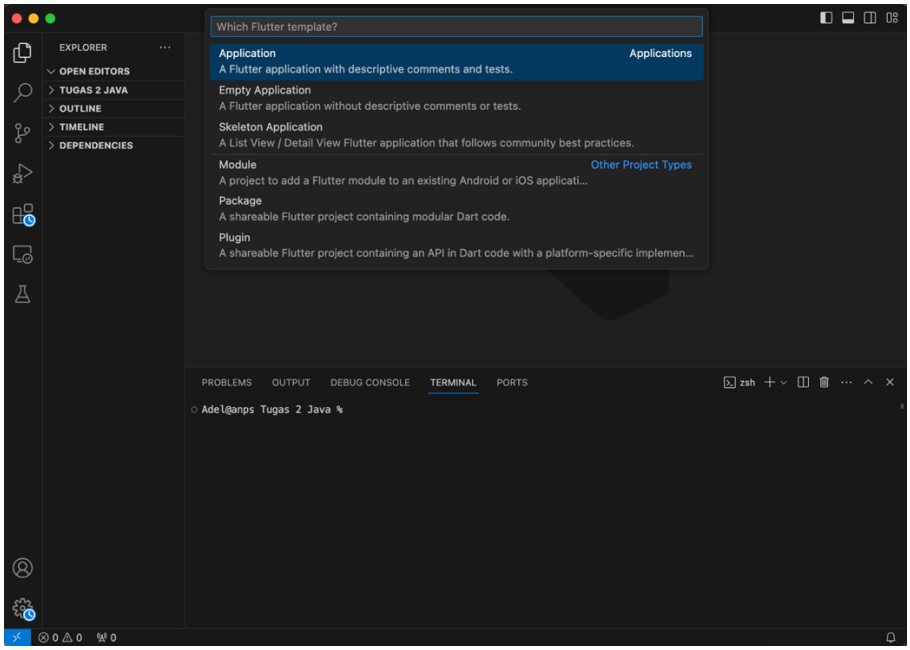
C. Membuat Aplikasi Flutter

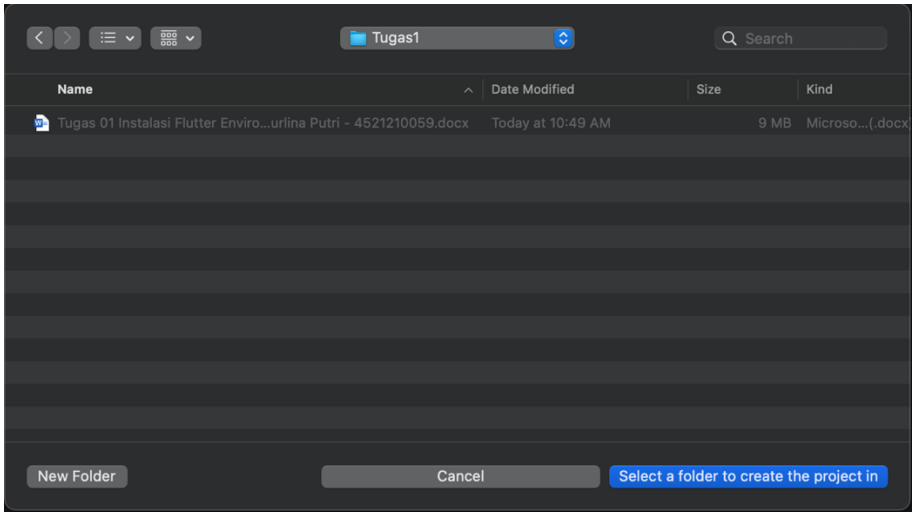
1. Membuat proyek Flutter

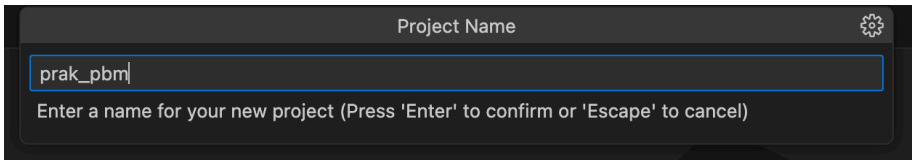
Buka Visual Studio Code dan buka palet perintah (Shift+Cmd+P). Ketik "flutter new". Pilih perintah Flutter: New Project.



Berikutnya, pilih Application lalu folder tempat project akan dibuat. Beri nama prak_pbm



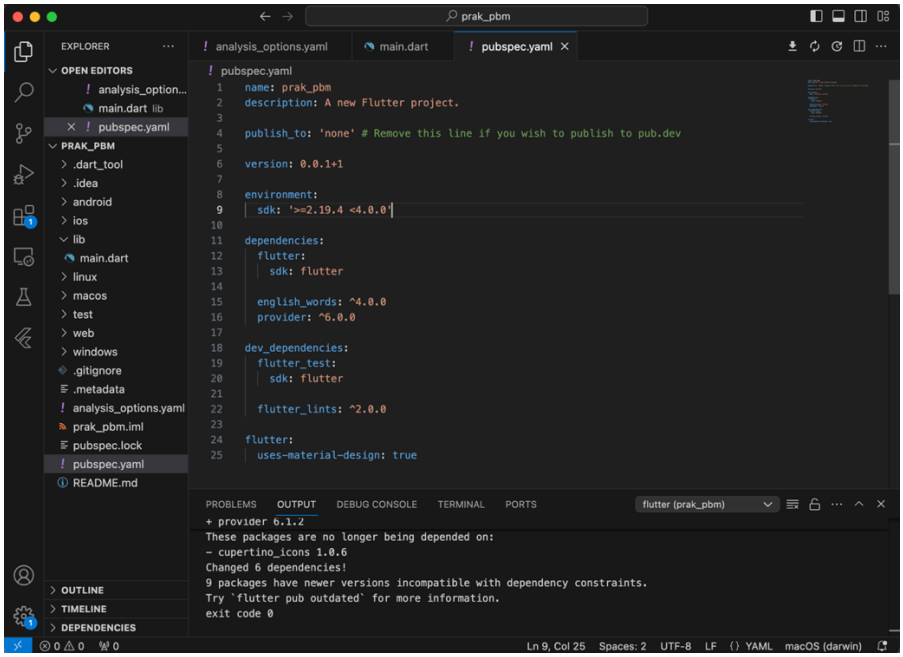




Buka file pubspec.yaml. copy paste :

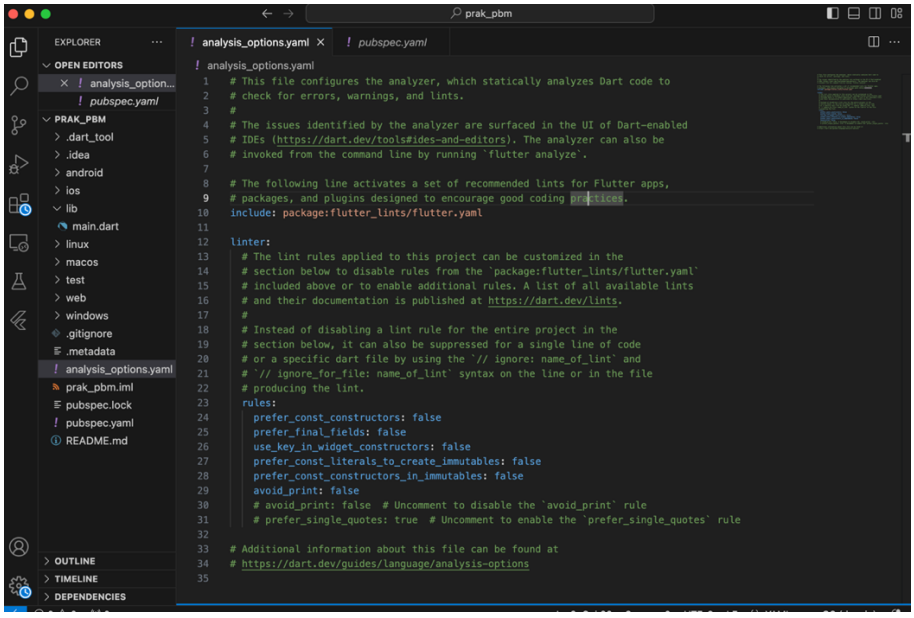
```
name: prak_pbm
description: A new Flutter project.

publish_to: 'none' # Remove this line if you wish to publish to pub.dev
```

```
version: 0.0.1+1

environment:
  sdk: '>=2.19.4 <4.0.0'

dependencies:
  flutter:
    sdk: flutter

  english_words: ^4.0.0
  provider: ^6.0.0

dev_dependencies:
  flutter_test:
    sdk: flutter

  flutter_lints: ^2.0.0

flutter:
  uses-material-design: true
```



Buka file analysis_options.yaml. copy paste :

```
include: package:flutter_lints/flutter.yaml

linter:
  rules:
    prefer_const_constructors: false
    prefer_final_fields: false
    use_key_in_widget_constructors: false
    prefer_const_literals_to_create_immutables: false
    prefer_const_constructors_in_immutables: false
    avoid_print: false
```

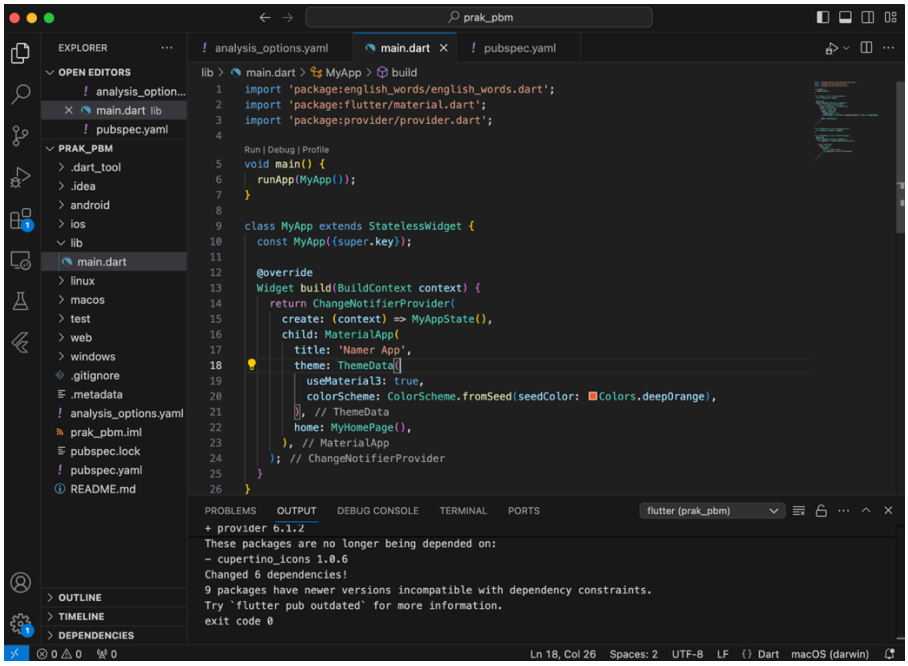


Terakhir buka main.dart pada direktori lib/. copy paste :

```
import 'package:english_words/english_words.dart';
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';

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

class MyApp extends StatelessWidget {
```

```
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return ChangeNotifierProvider(
      create: (context) => MyAppState(),
      child: MaterialApp(
        title: 'Namer App',
        theme: ThemeData(
          useMaterial3: true,
          colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepOrange),
        ),
        home: MyHomePage(),
      ),
    );
  }
}

class MyAppState extends ChangeNotifier {
  var current = WordPair.random();
}

class MyHomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    var appState = context.watch<MyAppState>();

    return Scaffold(
      body: Column(
        children: [
          Text('A random idea:'),
          Text(appState.current.asLowerCase),
        ],
      ),
    );
```

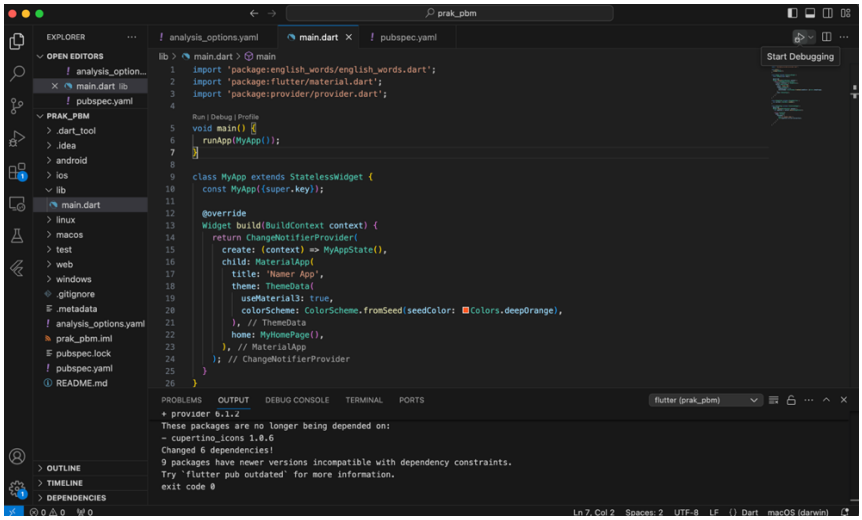
```
  }
}
```

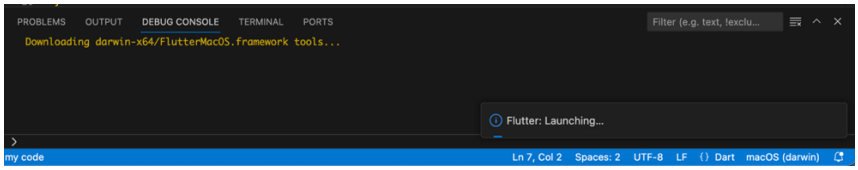


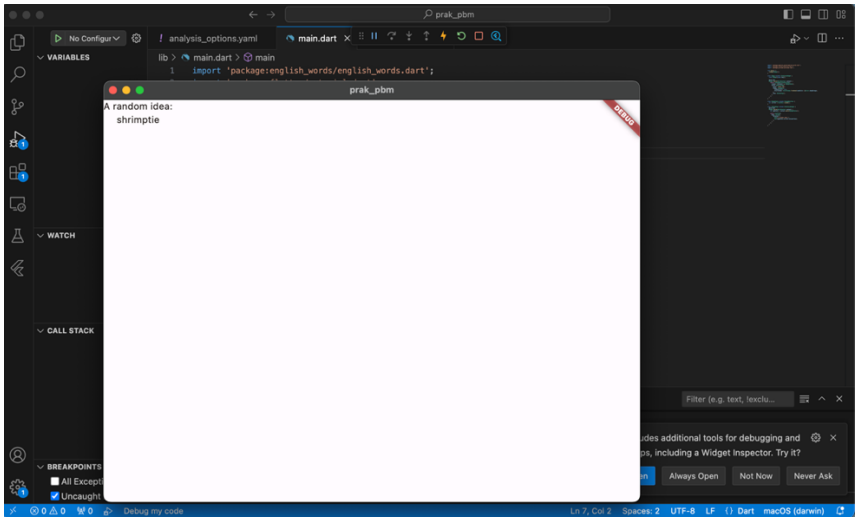
2. Menambahkan Tombol

a. Meluncurkan Aplikasi

Pertama, buka lib/main.dart dan pastikan Anda memilih perangkat target. Di bagian pojok kanan bawah VS Code. Dan Klik tombol Play diatas kanan untuk Start Debugging.







b. Hot Reload Pertama

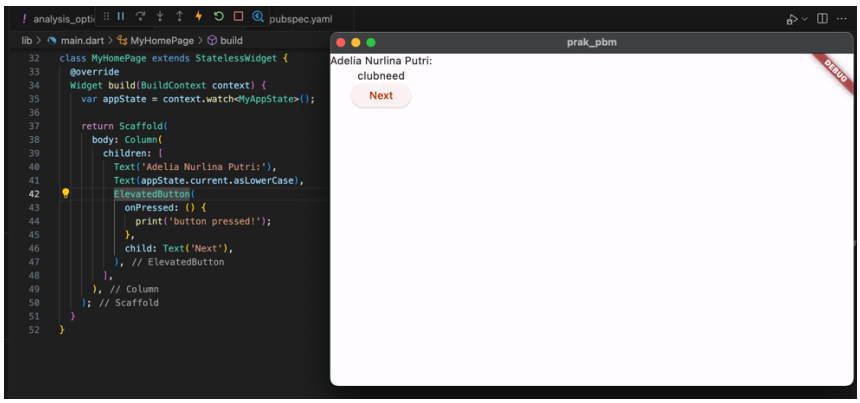
Di bagian bawah lib/main.dart, tambahkan sesuatu pada string di objek Text pertama, dan simpan file tersebut.

```
// ...

  return Scaffold(
    body: Column(
      children: [
        Text('Adelia Nurlina Putri:'), // ← Example change.
        Text(appState.current.asLowerCase),
      ],
    ),
  );

// ...
```

c. Menambahkan Tombol

Berikutnya, tambahkan tombol di bagian bawah Column, tepat di bawah instance Text kedua.

```
// ...

  return Scaffold(
    body: Column(
      children: [
        Text('A random AWESOME idea:'),
        Text(appState.current.asLowerCase),

        // ↓ Add this.
        ElevatedButton(
          onPressed: () {
            print('button pressed!');
          },
          child: Text('Next'),
        ),

      ],
    ),
  );

// ...
```



3. Memperindah Tampilan Aplikasi

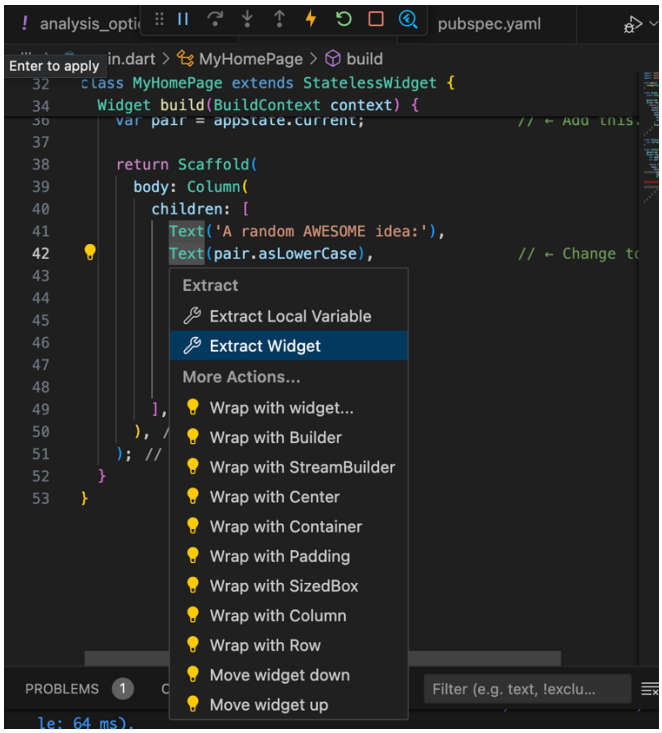
a. Mengekstrak widget

Tulis ulang widget MyHomePage sebagai berikut:

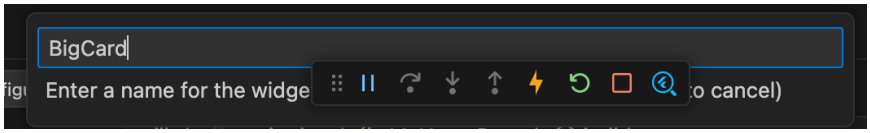
```
// ...

class MyHomePage extends StatelessWidget {
```

```
@override
Widget build(BuildContext context) {
  var appState = context.watch<MyAppState>();
  var pair = appState.current;              // ← Add this.

  return Scaffold(
    body: Column(
      children: [
        Text('A random AWESOME idea:'),
        Text(pair.asLowerCase),             // ← Change to this.
        ElevatedButton(
          onPressed: () {
            appState.getNext();
          },
          child: Text('Next'),
        ),
      ],
    ),
  );
 }
}

// ...
```

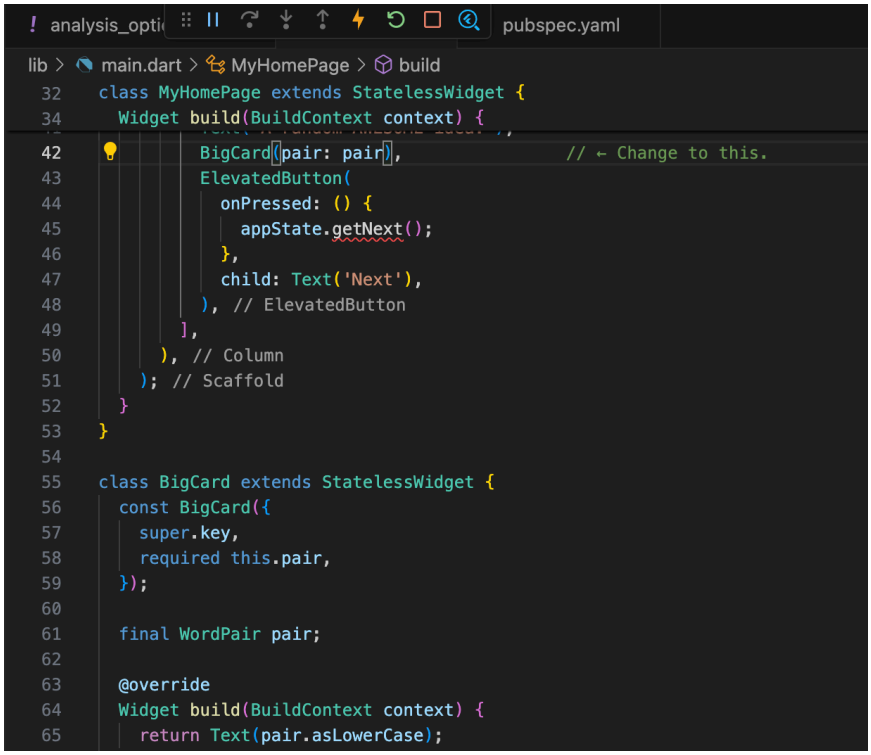
Panggil menu Refactor. Pada VS Code pindahkan kursor Anda ke potongan kode yang ingin di faktorkan ulang (dalam hal ini, Text), lalu tekan Cmd+.



Pada menu Refactor, pilih Extract Widget. Tetapkan nama, seperti BigCard, lalu klik Enter

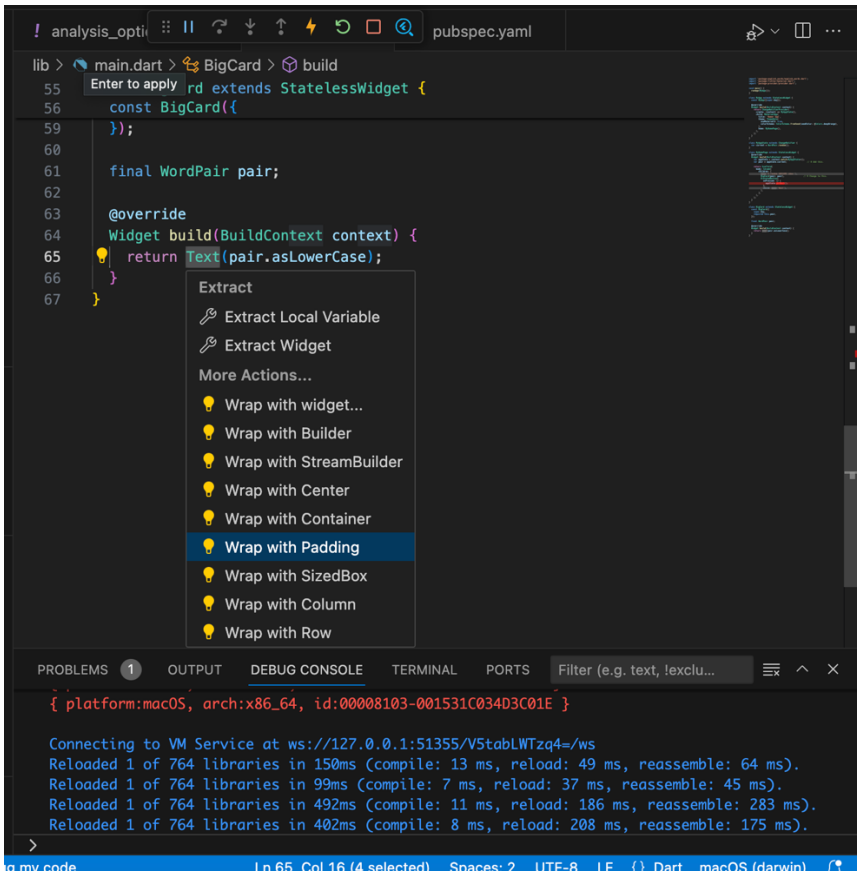


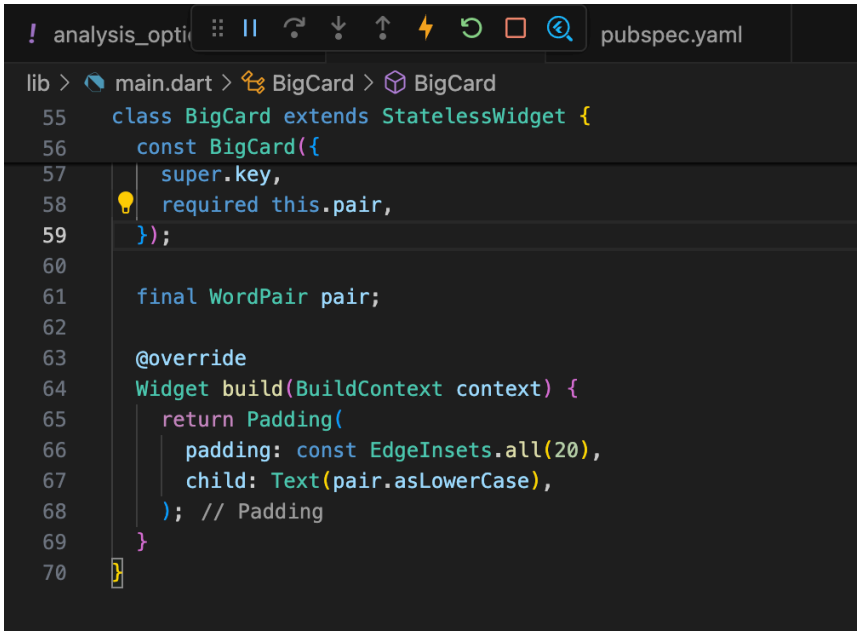
Tindakan ini secara otomatis membuat class baru, BigCard, di akhir file saat ini. Class tersebut akan terlihat seperti berikut:



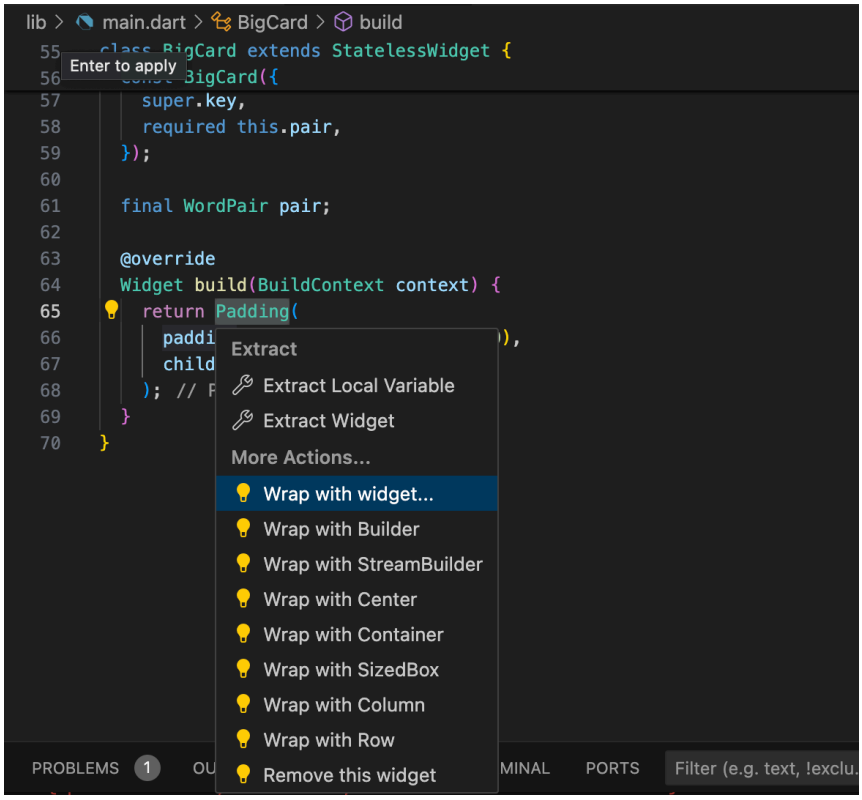
b. Menambahkan Kartu

Temukan class BigCard dan metode build() yang berada di dalamnya. Sama seperti sebelumnya, panggil menu Refactor pada widget Text. pilih Wrap with Padding.





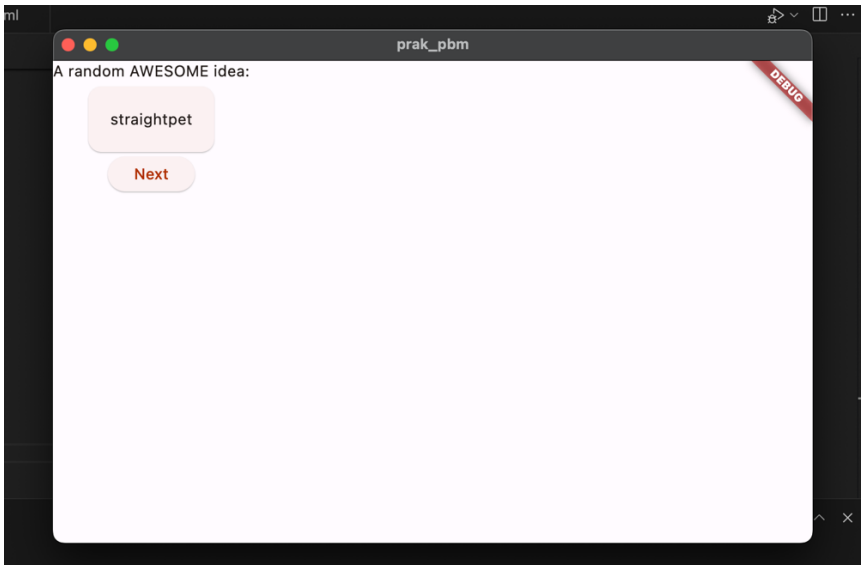
Tempatkan kursor pada widget Padding, buka menu Refactor, lalu pilih Wrap with widget....



Ubah Padding menjadi Card

```
63    @override
64    Widget build(BuildContext context) {
65      return Card(
66        child: Padding(
67          padding: const EdgeInsets.all(20),
68          child: Text(pair.asLowerCase),
69        ), // Padding
70      ); // Card
71    }
72  }
```

Hasil :



c. Tema dan Gaya

Buat perubahan berikut untuk metode build() BigCard

```
// ...

 @override
 Widget build(BuildContext context) {
  final theme = Theme.of(context);      // ← Add this.

  return Card(
    color: theme.colorScheme.primary,    // ← And also this.
    child: Padding(
      padding: const EdgeInsets.all(20),
      child: Text(pair.asLowerCase),
    ),
  );
 }
```

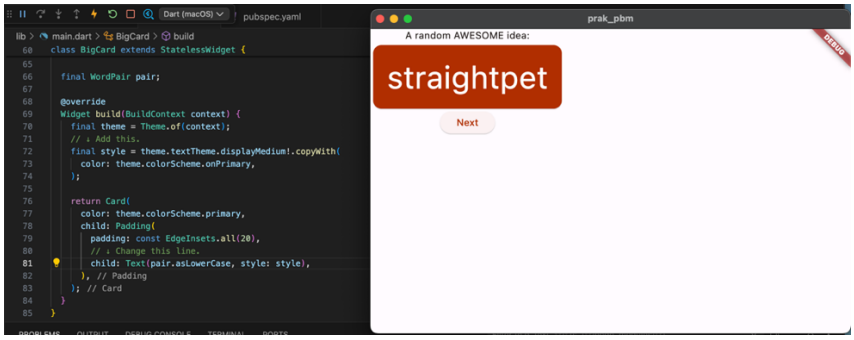
```
// ...
```

Hasil :

d. TextTheme

Perbaiki tampilan card dengan kode berikut :

```
// ...

 @override
 Widget build(BuildContext context) {
  final theme = Theme.of(context);
  // ↓ Add this.
  final style = theme.textTheme.displayMedium!.copyWith(
   color: theme.colorScheme.onPrimary,
  );

  return Card(
   color: theme.colorScheme.primary,
   child: Padding(
    padding: const EdgeInsets.all(20),
    // ↓ Change this line.
    child: Text(pair.asLowerCase, style: style),
   ),
  );
 }

// ...
```



e. Meningkatkan Aksesibilitas

Gunakan properti semanticsLabel Text untuk mengganti konten visual widget teks dengan konten semantik yang lebih sesuai untuk pembaca layar:

```
// ...

 @override
 Widget build(BuildContext context) {
  final theme = Theme.of(context);
  final style = theme.textTheme.displayMedium!.copyWith(
   color: theme.colorScheme.onPrimary,
  );

  return Card(
   color: theme.colorScheme.primary,
   child: Padding(
    padding: const EdgeInsets.all(20),

    // ↓ Make the following change.
    child: Text(
     pair.asLowerCase,
     style: style,
     semanticsLabel: "${pair.first} ${pair.second}",
    ),
   ),
  );
 }

// ...
```

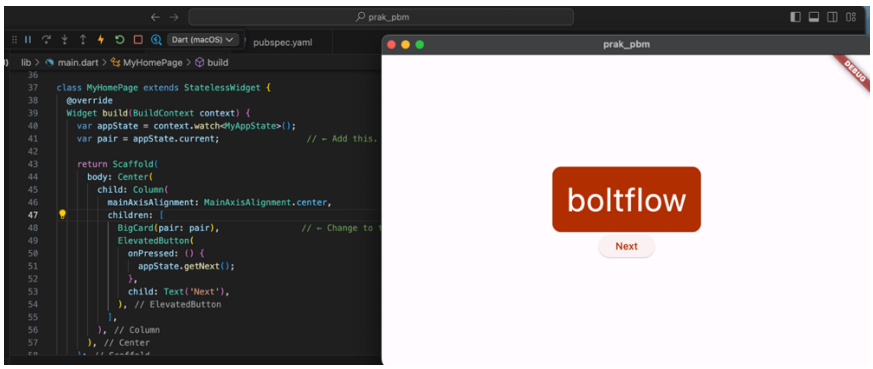
f. Menempatkan UI di tengah

Buka metode build() MyHomePage, dan buat perubahan berikut:

```
// ...

class MyHomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    var appState = context.watch<MyAppState>();
    var pair = appState.current;

    return Scaffold(
      body: Column(
        mainAxisAlignment: MainAxisAlignment.center,  // ← Add this.
        children: [
          Text('A random AWESOME idea:'),
          BigCard(pair: pair),
          ElevatedButton(
            onPressed: () {
              appState.getNext();
            },
            child: Text('Next'),
          ),
        ],
      ),
    );
  }
}

// ...
```

Letakkan kursor Anda di Column, buka menu Refactor (dengan Ctrl+. atau Cmd+.), lalu pilih Wrap with Center.

Hasil :



4. Menambahkan Fungsi

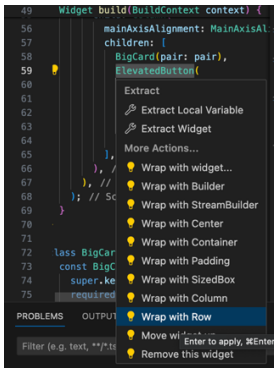
a. Menambahkan Logika Bisnis

Scroll ke MyAppState dan tambahkan kode berikut:

```
// ...

class MyAppState extends ChangeNotifier {
 var current = WordPair.random();

 void getNext() {
  current = WordPair.random();
  notifyListeners();
 }

 // ↓ Add the code below.
 var favorites = <WordPair>[];

 void toggleFavorite() {
  if (favorites.contains(current)) {
   favorites.remove(current);
  } else {
   favorites.add(current);
  }
  notifyListeners();
 }
}

// ...
```

b. Menambahkan Tombol

Meletakkan tombol ‘Like' di sebelah kiri tombol ‘Next' memerlukan Row. Pertama, gabungkan tombol yang ada pada Row. Buka metode build() MyHomePage, letakkan kursor pada ElevatedButton, buka menu Refactor dengan Ctrl+. atau Cmd+., lalu pilih Wrap with Row.



Buat perubahan berikut:

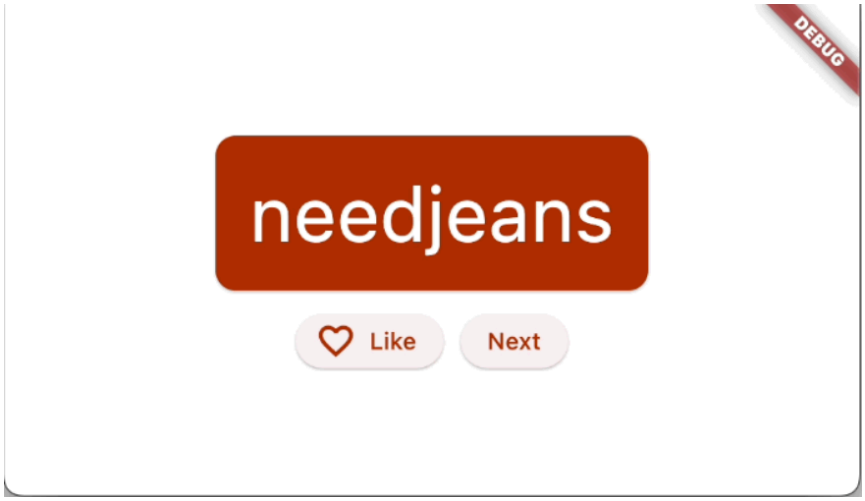
```
// ...

class MyHomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    var appState = context.watch<MyAppState>();
    var pair = appState.current;

    return Scaffold(
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            BigCard(pair: pair),
            SizedBox(height: 10),
            Row(
              mainAxisSize: MainAxisSize.min,   // ← Add this.
              children: [
                ElevatedButton(
                  onPressed: () {
                    appState.getNext();
                  },
                  child: Text('Next'),
                ),
              ],
            ),
          ],
        ),
      ),
    );
  }
}

// ...
```

Menambahkan tombol untuk myHomePage menggunakan konstruktor ElevatedButton.icon() untuk membuat tombol dengan ikon. Di bagian atas metode build, pilih ikon yang sesuai tergantung

pada apakah pasangan kata saat ini sudah berada di favorit atau tidak.

```
// ...

class MyHomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    var appState = context.watch<MyAppState>();
    var pair = appState.current;

    // ↓ Add this.
    IconData icon;
    if (appState.favorites.contains(pair)) {
      icon = Icons.favorite;
    } else {
      icon = Icons.favorite_border;
    }

    return Scaffold(
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            BigCard(pair: pair),
            SizedBox(height: 10),
            Row(
              mainAxisSize: MainAxisSize.min,
              children: [

                // ↓ And this.
                ElevatedButton.icon(
                  onPressed: () {
                    appState.toggleFavorite();
                  },
                  icon: Icon(icon),
                  label: Text('Like'),
                ),
                SizedBox(width: 10),

                ElevatedButton(
                  onPressed: () {
                    appState.getNext();
                  },
                  child: Text('Next'),
                ),
              ],
            ),
          ],
        ),
      ),
    );
  }
```

```
}

// ...
```



5. Menambahkan Kolom Samping

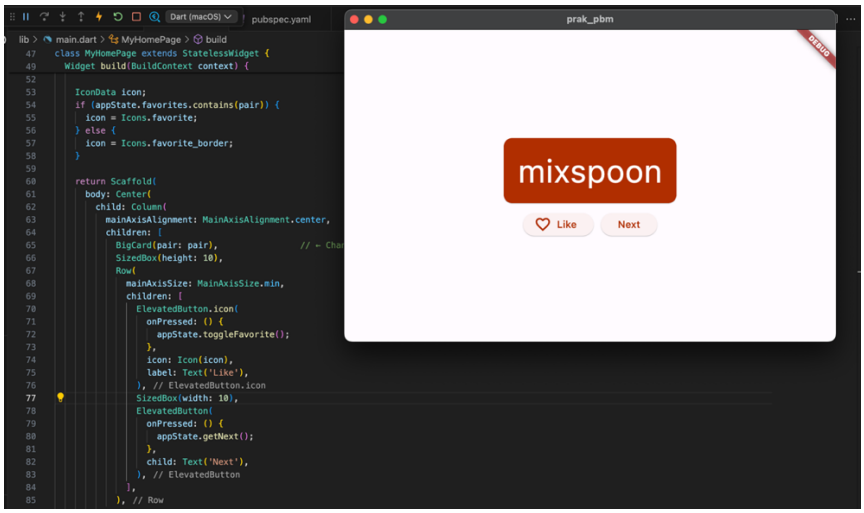
a. Pisahkan MyHomePage menjadi 2 widget terpisah

Pilih keseluruhan MyHomePage, hapus, dan gantikan dengan kode berikut:

```
// ...

class MyHomePage extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
  return Scaffold(
   body: Row(
    children: [
     SafeArea(
      child: NavigationRail(
       extended: false,
       destinations: [
        NavigationRailDestination(
         icon: Icon(Icons.home),
         label: Text('Home'),
        ),
        NavigationRailDestination(
         icon: Icon(Icons.favorite),
         label: Text('Favorites'),
        ),
       ],
```

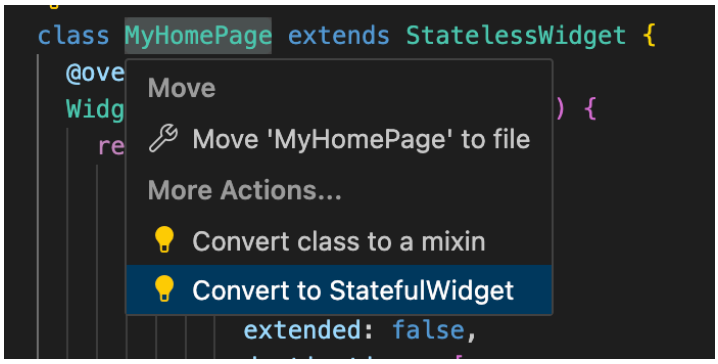
```
            selectedIndex: 0,
            onDestinationSelected: (value) {
              print('selected: $value');
            },
          ),
        ),
        Expanded(
          child: Container(
            color: Theme.of(context).colorScheme.primaryContainer,
            child: GeneratorPage(),
          ),
        ),
      ],
    ),
  );
 }
}

class GeneratorPage extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
  var appState = context.watch<MyAppState>();
  var pair = appState.current;

  IconData icon;
  if (appState.favorites.contains(pair)) {
   icon = Icons.favorite;
  } else {
   icon = Icons.favorite_border;
  }

  return Center(
   child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
     BigCard(pair: pair),
     SizedBox(height: 10),
     Row(
      mainAxisSize: MainAxisSize.min,
      children: [
       ElevatedButton.icon(
        onPressed: () {
         appState.toggleFavorite();
        },
        icon: Icon(icon),
        label: Text('Like'),
       ),
       SizedBox(width: 10),
       ElevatedButton(
        onPressed: () {
         appState.getNext();
        },
        child: Text('Next'),
       ),
```

```
          ],
        ),
      ],
    ),
  );
 }
}

// ...
```



b. Widget stateless versus stateful

Pertama, konversi MyHomePage menjadi widget stateful. Tempatkan kursor Anda di baris pertama MyHomePage (baris yang diawali dengan class MyHomePage...), lalu buka menu Refactor menggunakan Ctrl+. atau Cmd+.. Kemudian, pilih Convert to StatefulWidget.



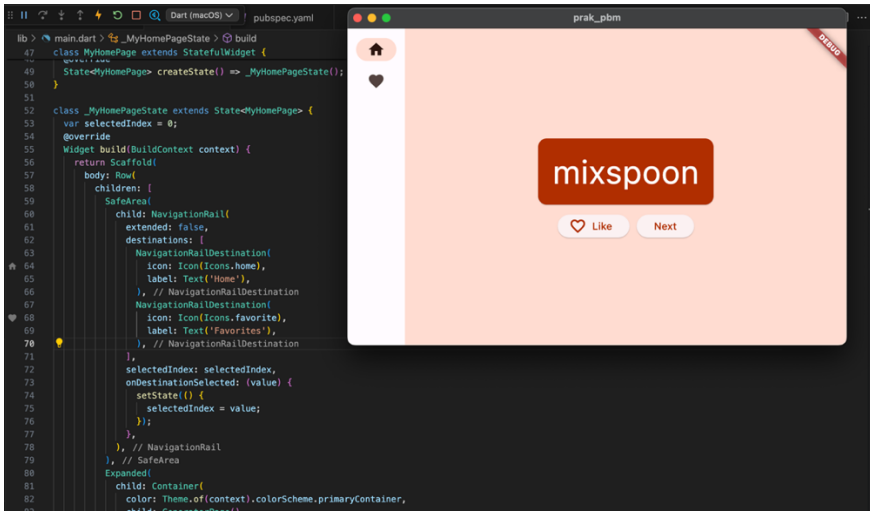
```dart
class MyHomePage extends StatefulWidget {
  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Row(
        children: [
          SafeArea(
```

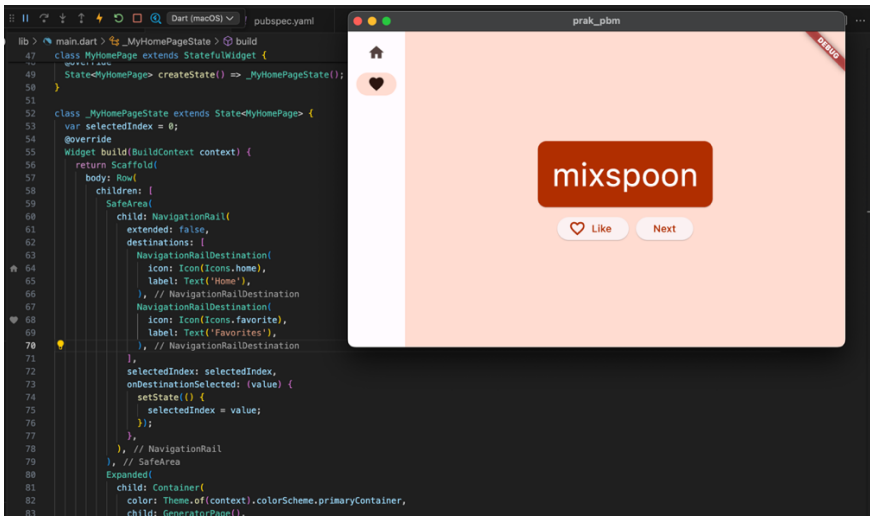
c. setState

Widget stateful baru hanya perlu melacak satu variabel: selectedIndex. Buat 3 perubahan berikut untuk _MyHomePageState :

```dart
// ...

class _MyHomePageState extends State<MyHomePage> {

 var selectedIndex = 0;    // ← Add this property.

 @override
 Widget build(BuildContext context) {
  return Scaffold(
    body: Row(
     children: [
      SafeArea(
       child: NavigationRail(
        extended: false,
        destinations: [
         NavigationRailDestination(
          icon: Icon(Icons.home),
          label: Text('Home'),
         ),
         NavigationRailDestination(
          icon: Icon(Icons.favorite),
          label: Text('Favorites'),
         ),
        ],
        selectedIndex: selectedIndex,    // ← Change to this.
        onDestinationSelected: (value) {

         // ↓ Replace print with this.
         setState(() {
          selectedIndex = value;
         });
```

```
          },
        ),
      ),
      Expanded(
        child: Container(
          color: Theme.of(context).colorScheme.primaryContainer,
          child: GeneratorPage(),
        ),
      ),
    ],
  ),
 );
 }
}

// ...
```



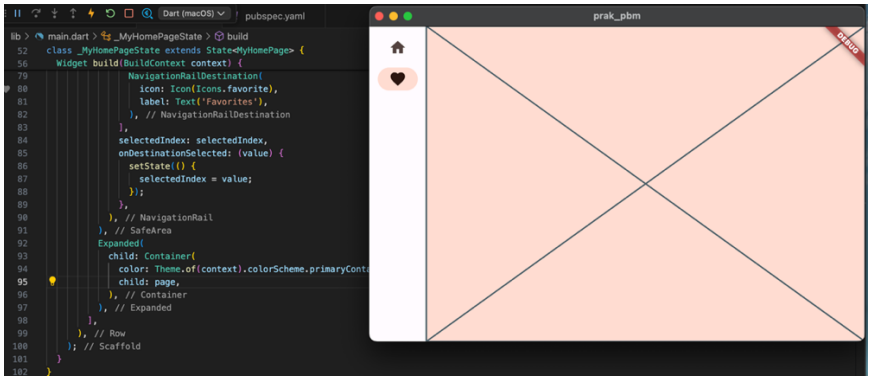


d. Menggunakan selectedIndex

Tempatkan kode berikut di bagian atas metode build _MyHomePageState, tepat sebelum return Scaffold:

```
// ...

Widget page;
switch (selectedIndex) {
  case 0:
    page = GeneratorPage();
    break;
  case 1:
    page = Placeholder();
    break;
  default:
    throw UnimplementedError('no widget for $selectedIndex');
}

// ...
```

Aplikasi sekarang beralih di antara GeneratorPage kita dan placeholder yang akan segera menjadi halaman Favorites.

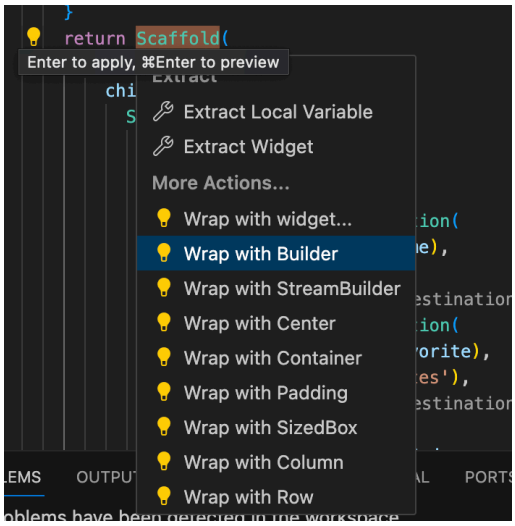


e. Tingkat respons

Gunakan menu Refactor Flutter di VS Code untuk membuat perubahan yang diperlukan

- Dalam metode build _MyHomePageState, letakkan kursor Anda pada Scaffold.
- Buka menu Refactor dengan Cmd+. (Mac).
- Pilih Wrap with Builder dan tekan Enter.

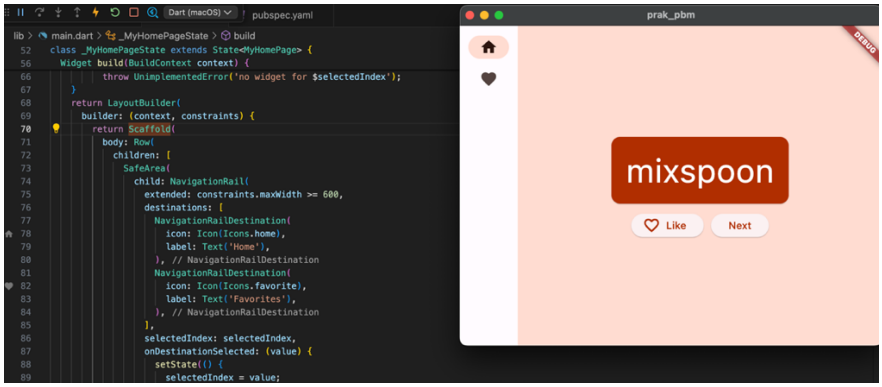
- Modifikasi nama Builder yang baru ditambahkan menjadi LayoutBuilder.
- Modifikasi daftar parameter callback dari (context) menjadi (context, constraints).

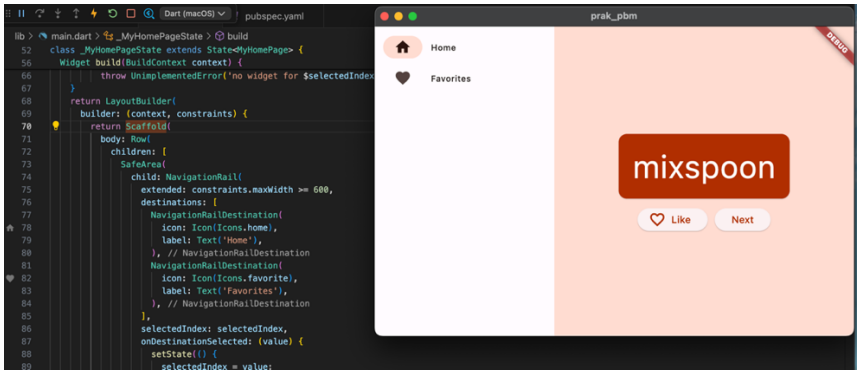


Buat perubahan baris tunggal berikut untuk metode build _MyHomePageState:

```
// ...

class _MyHomePageState extends State<MyHomePage> {
 var selectedIndex = 0;

 @override
 Widget build(BuildContext context) {
  Widget page;
  switch (selectedIndex) {
   case 0:
    page = GeneratorPage();
    break;
   case 1:
    page = Placeholder();
    break;
   default:
    throw UnimplementedError('no widget for $selectedIndex');
  }

  return LayoutBuilder(builder: (context, constraints) {
   return Scaffold(
    body: Row(
     children: [
      SafeArea(
       child: NavigationRail(
        extended: constraints.maxWidth >= 600,  // ← Here.
```

```
            destinations: [
              NavigationRailDestination(
                icon: Icon(Icons.home),
                label: Text('Home'),
              ),
              NavigationRailDestination(
                icon: Icon(Icons.favorite),
                label: Text('Favorites'),
              ),
            ],
            selectedIndex: selectedIndex,
            onDestinationSelected: (value) {
              setState(() {
                selectedIndex = value;
              });
            },
          ),
        ),
        Expanded(
          child: Container(
            color: Theme.of(context).colorScheme.primaryContainer,
            child: page,
          ),
        ),
      ],
    ),
  );
 });
 }
}

// ...
```





6. Menambahkan Halaman Baru

Class FavoritesPage baru :

```
// ...

class FavoritesPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    var appState = context.watch<MyAppState>();

    if (appState.favorites.isEmpty) {
      return Center(
        child: Text('No favorites yet.'),
      );
    }

    return ListView(
      children: [
        Padding(
          padding: const EdgeInsets.all(20),
          child: Text('You have '
              '${appState.favorites.length} favorites:'),
        ),
        for (var pair in appState.favorites)
          ListTile(
            leading: Icon(Icons.favorite),
            title: Text(pair.asLowerCase),
          ),
      ],
    );
  }
}
```

Mengganti widget Placeholder dengan FavoritesPage :

```
Widget build(BuildContext context) {

  Widget page;
```

```
            switch (selectedIndex) {
              case 0:
                page = GeneratorPage();
              break;
              case 1:
                page = FavoritesPage();
              break;
              default:
                throw UnimplementedError('no widget for
        $selectedIndex');
  }
```

Hasil :

D. Push Code to GitHub https://github.com/adelianurlinap/Prak_PBM.git

a. Clone Repository yang telah dibuat :

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

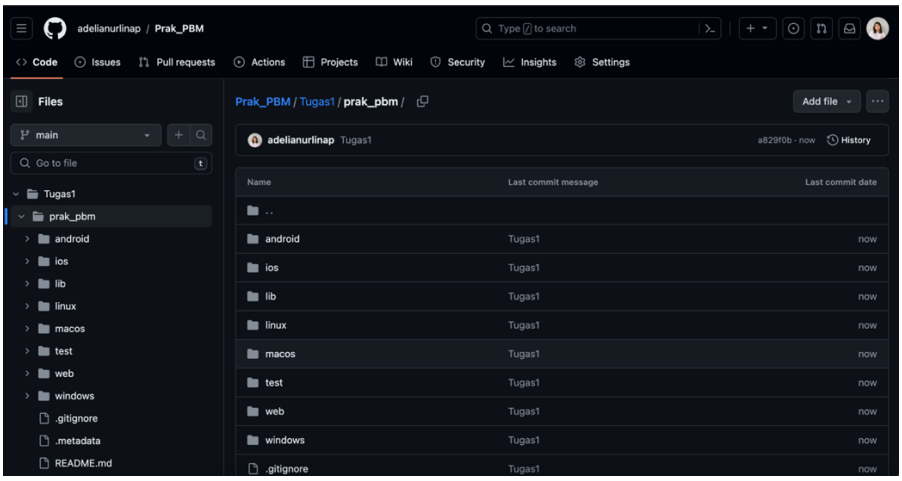
b. Add dan commit

```
git add .
git commit -m "Tugas1"
```

c. Push code

```
git push
```

```
create mode 100644 Tugas1/prak_pbm/windows/flutter/CMakeLists.txt
create mode 100644 Tugas1/prak_pbm/windows/flutter/generated_plugin_registrant.cc
create mode 100644 Tugas1/prak_pbm/windows/flutter/generated_plugin_registrant.h
create mode 100644 Tugas1/prak_pbm/windows/flutter/generated_plugins.cmake
create mode 100644 Tugas1/prak_pbm/windows/runner/CMakeLists.txt
create mode 100644 Tugas1/prak_pbm/windows/runner/Runner.rc
create mode 100644 Tugas1/prak_pbm/windows/runner/flutter_window.cpp
create mode 100644 Tugas1/prak_pbm/windows/runner/flutter_window.h
create mode 100644 Tugas1/prak_pbm/windows/runner/main.cpp
create mode 100644 Tugas1/prak_pbm/windows/runner/resource.h
create mode 100644 Tugas1/prak_pbm/windows/runner/resources/app_icon.ico
create mode 100644 Tugas1/prak_pbm/windows/runner/runner.exe.manifest
create mode 100644 Tugas1/prak_pbm/windows/runner/utils.cpp
create mode 100644 Tugas1/prak_pbm/windows/runner/utils.h
create mode 100644 Tugas1/prak_pbm/windows/runner/win32_window.cpp
create mode 100644 Tugas1/prak_pbm/windows/runner/win32_window.h
Adel@anps Prak_PBM % git push
Enumerating objects: 182, done.
Counting objects: 100% (182/182), done.
Delta compression using up to 8 threads
Compressing objects: 100% (150/150), done.
Writing objects: 100% (181/181), 2.52 MiB | 5.67 MiB/s, done.
Total 181 (delta 20), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (20/20), done.
To https://github.com/adelianurlinap/Prak_PBM.git
   de8919d..a829f0b  main -> main
Adel@anps Prak_PBM %
```



E. Kesimpulan

Pada tugas 1 ini, telah dilakukan praktek untuk membuat aplikasi flutter dasar. Aplikasi berbentuk 2 page dengan card random string yang dapat berubah-ubah di like sebagai favorite. Terdapat 2 page yaitu page Home dan Favorite. Diatur juga mengenai responsif dari window page. Terakhir yaitu melakukan push code ke GitHub.

F. Referensi

- https://docs.flutter.dev/get-started/install/macos/mobile-android?tab=virtual
- https://marketplace.visualstudio.com/items?itemName=Dart-Code.flutter
- https://docs.flutter.dev/get-started/install/macos/mobile-android?tab=virtual
- https://docs.flutter.dev/get-started/install/macos/mobile-ios
- https://www.codewithhussain.com/flutter-cmdline-tools-component-is-missing
- https://codelabs.developers.google.com/codelabs/flutter-codelab-first?hl=id#0