

MODUL 1

CODELAB

PEMROGRAMAN MOBILE D

Dosen Pengampu:

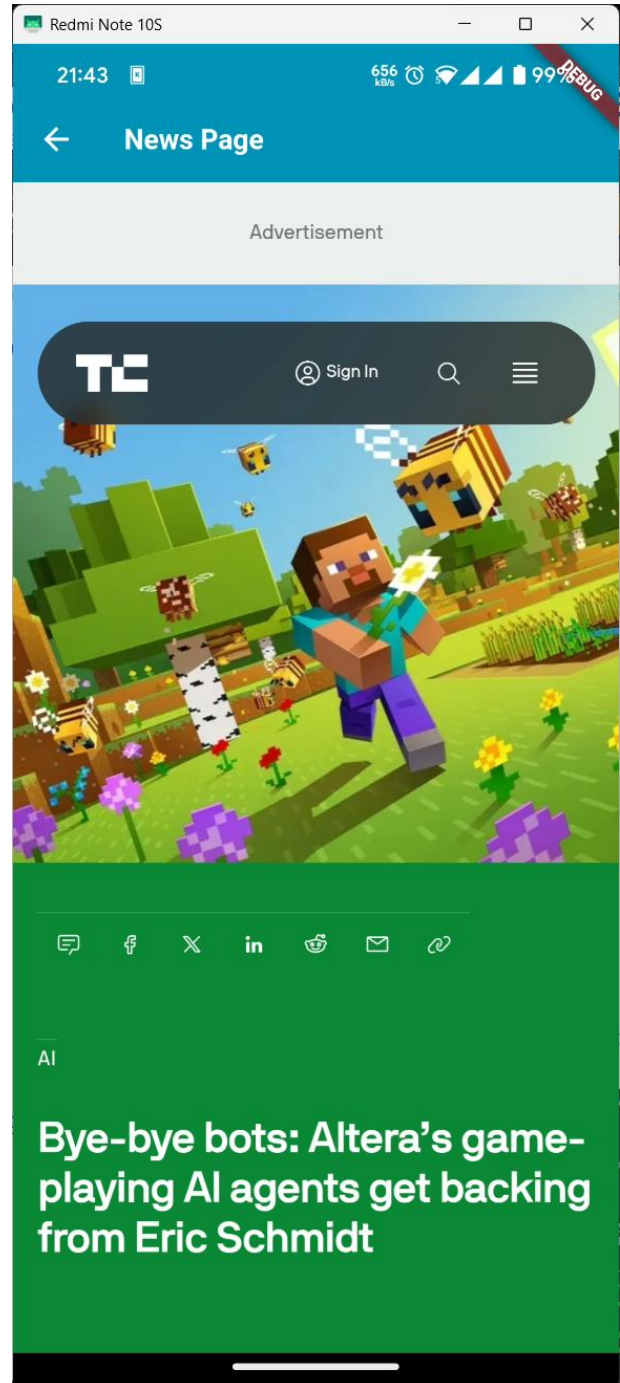
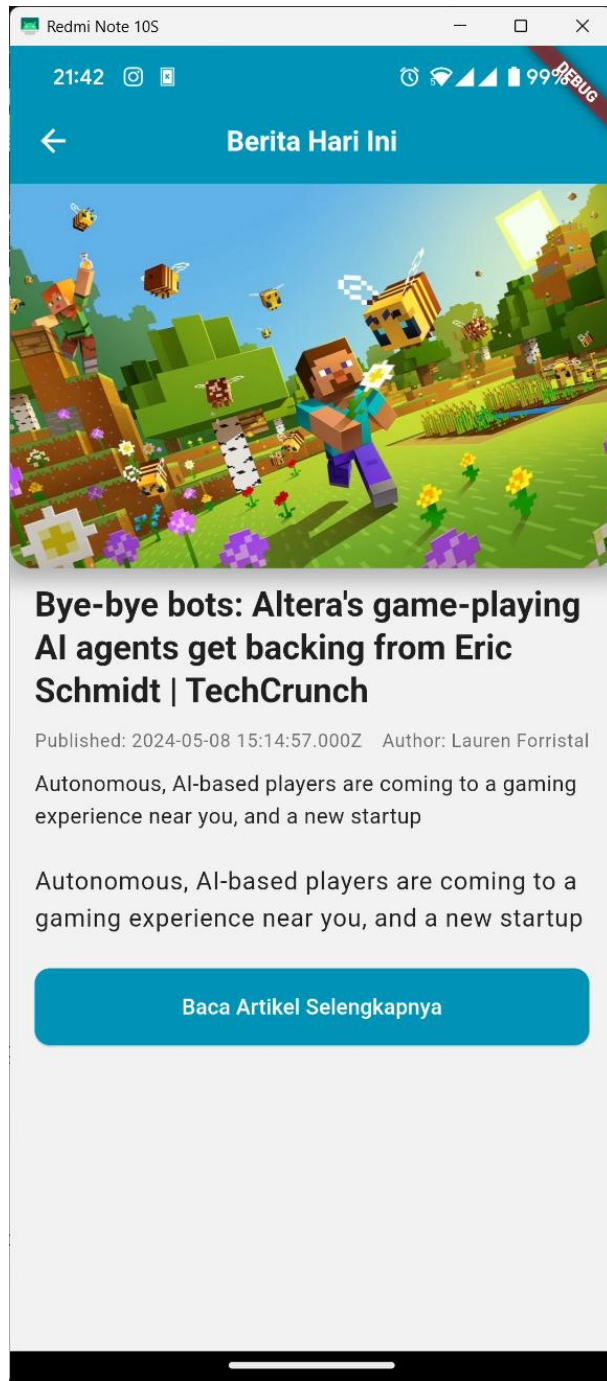
Bashor Fauzan Muthohirin, S.Kom., M.Kom

Source Code: https://github.com/hisyam99/PRAK-PEMROGRAMAN-MOBILE-2024/tree/main/TUGAS/modul_2/CODELAB

Muhammad Hisyam Kamil
202210370311060

Daftar Isi

MODUL 2: CODELAB	4
I. Pendahuluan.....	4
II. Instalasi Package dan WebView	4
III. Implementasi GetConnect dan WebView.....	6
IV. Tampilan Aplikasi setelah penerapan GetConnect, dan WebView.....	18



MODUL 2: CODELAB

Source Code: https://github.com/hisyam99/PRAK-PEMROGRAMAN-MOBILE-2024/tree/main/TUGAS/modul_2/CODELAB

CODE LAB

TUGAS 1 – INSTALASI PACKAGE DAN WEBVIEW

1. Lakukan **installasi package** (http, Dio, GetConnect) bebas sesuai keinginan kalian, selain itu juga install **WebView** dengan menambahkan dependencies di file **pubspec.yaml**

TUGAS 2 – LATIHAN IMPLEMENTASI HTTP, DIO, GETCONNECT DAN WEBVIEW

1. Lakukan latihan dari salah satu dari 3 package [http-package](#), [dio-package](#), [getconnect-package](#). API yang dipakai yaitu melalui <https://my-json-server.typicode.com/Fallid/codelab-api/db>.
2. Lakukan konversi pada data JSON menjadi sebuah model menggunakan aplikasi [QuickType](#).
3. Implementasikan penggunaan salah satu dari 3 package tersebut.
4. Buatlah **User Interface** untuk menampilkan data dari API yang sudah di ambil.
5. Implementasikan penggunaan [WebView](#) sederhana. Untuk URL website yang ingin ditampilkan bebas.
6. Untuk mempermudah pengerjaan Latihan CodeLab, kalian bisa melihat referensi langsung dari [online editor ini](#)

I. Pendahuluan

Pada modul 2 codelab ini, kita diminta untuk melakukan instalasi salah satu dari ketiga package yaitu http-package, dio-package, dan getconnect-package. Setelah itu melakukan instalasi WebView dan terakhir melakukan penerapan API.

II. Instalasi Package dan WebView

Karena saya menggunakan arsitektur pattern GetX sebelumnya, maka disini saya akan menggunakan getconnect-package. Salah satu alasannya agar cukup menggunakan dependencies dari get untuk melakukan penerapan API ini, jadi tidak perlu import dependencies yang lain lagi kecuali WebView dan Get yang sudah saya import sebelumnya.

1. Buka pubspec.yaml dan tambahkan dependencies seperti yang ada pada gambar dibawah ini:

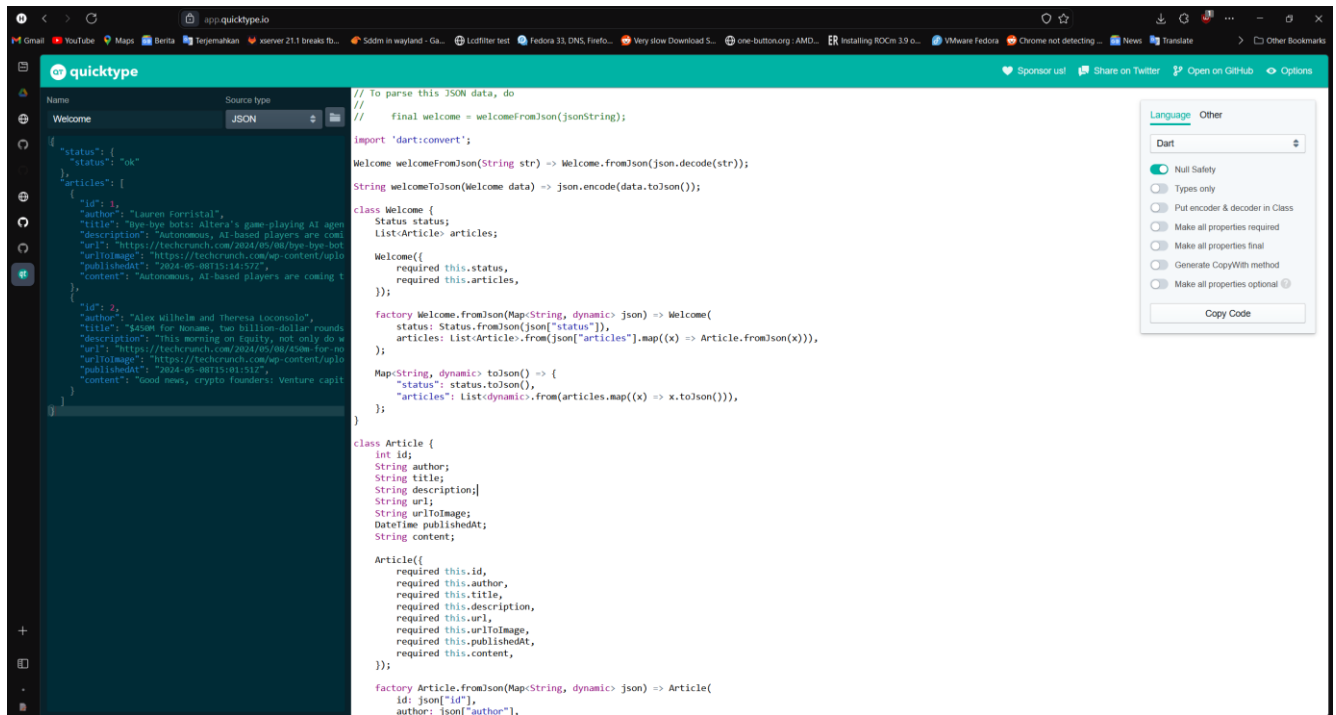
```
pubspec.yaml X
pubspec.yaml > {} dependencies > webview_flutter
1  name: tugas_1
2  version: 1.0.0+1
3  publish_to: none
4  description: A new Flutter project.
5  environment:
6    sdk: '>=3.2.6 <4.0.0'
7
8  dependencies:
9    cupertino_icons: ^1.0.2
10   get: 4.6.6
11   webview_flutter: ^4.8.0
12   flutter:
13     sdk: flutter
14
15  dev_dependencies:
16    flutter_lints: ^2.0.0
17    flutter_test:
18      sdk: flutter
19
20  flutter:
21    uses-material-design: true
22
PROBLEMS 1 TERMINAL OUTPUT DEBUG CONSOLE PORTS
PS C:\Users\hisyam99\Documents\TUGAS KULIAH\Semester 5\
```

2. Setelah ditambahkan, kita execute command “flutter pub get” untuk memastikan semua dependencies yang sudah kita tambahkan tadi akan terunduh semua

```
PS C:\Users\hisyam99\Documents\TUGAS KULIAH\Semester 5\Pemrograman Mobile\PRAKTIKUM\TUGAS\modul_2\CODELAB\codeLab_2> flutter pub get
Resolving dependencies...
Downloading packages...
  async 2.11.0 (2.12.0 available)
  characters 1.3.0 (1.3.1 available)
  collection 1.18.0 (1.19.0 available)
  flutter_lints 2.0.3 (5.0.0 available)
  leak_tracker 10.0.5 (10.0.7 available)
  leak_tracker_flutter_testing 3.0.5 (3.0.8 available)
  lints 2.1.1 (5.1.0 available)
  material_color_utilities 0.11.1 (0.12.0 available)
  meta 1.15.0 (1.16.0 available)
  stack_trace 1.11.1 (1.12.0 available)
  string_scanner 1.2.0 (1.3.0 available)
  test_api 0.7.2 (0.7.3 available)
  vm_service 14.2.5 (14.3.0 available)
  webview_flutter 4.9.0 (4.10.0 available)
  webview_flutter_android 3.16.9 (4.0.0 available)
Got dependencies!
15 packages have newer versions incompatible with dependency constraints.
Try `flutter pub outdated` for more information.
PS C:\Users\hisyam99\Documents\TUGAS KULIAH\Semester 5\Pemrograman Mobile\PRAKTIKUM\TUGAS\modul_2\CODELAB\codeLab_2>
```

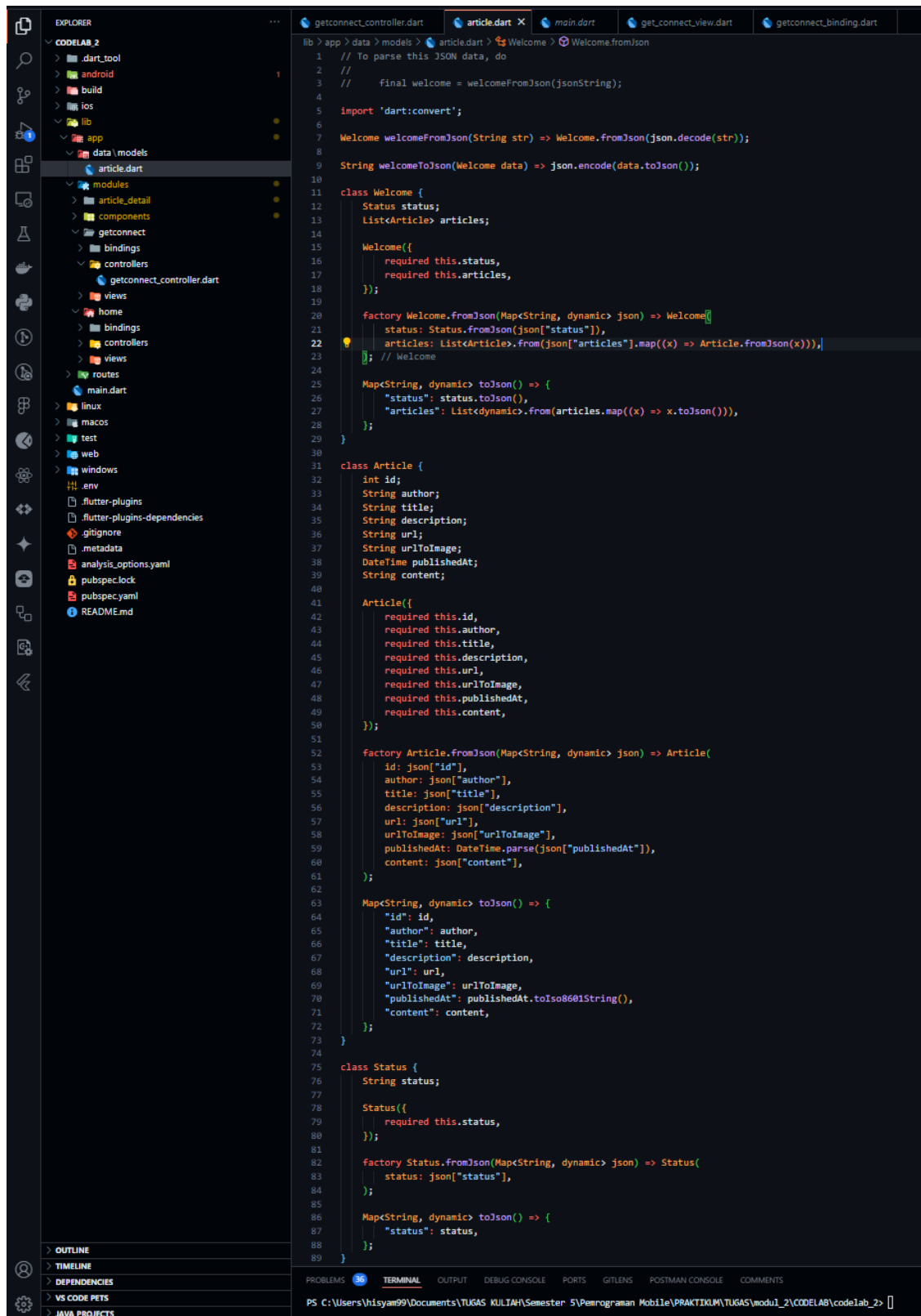
III. Implementasi GetConnect dan WebView

1. Pertama kita melakukan konversi pada data JSON menjadi sebuah model menggunakan website [QuickType](https://my-json-server.typicode.com/Fallid/codelab-api/db). Untuk data json yang kita gunakan diambil dari sini: <https://my-json-server.typicode.com/Fallid/codelab-api/db>



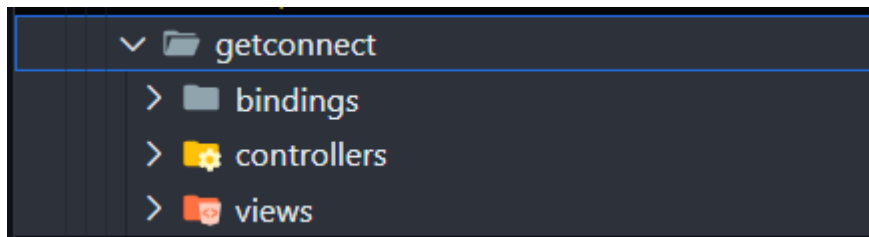
Dapat terlihat data yang sebelumnya dalam bentuk JSON, sekarang diconvert kedalam bentuk model menggunakan bahasa dart.

2. Buat file `"/lib/app/data/models/article.dart"` lalu salin semua kode model yang sudah kita dapatkan tadi kedalam file tersebut, seperti pada gambar dibawah ini:

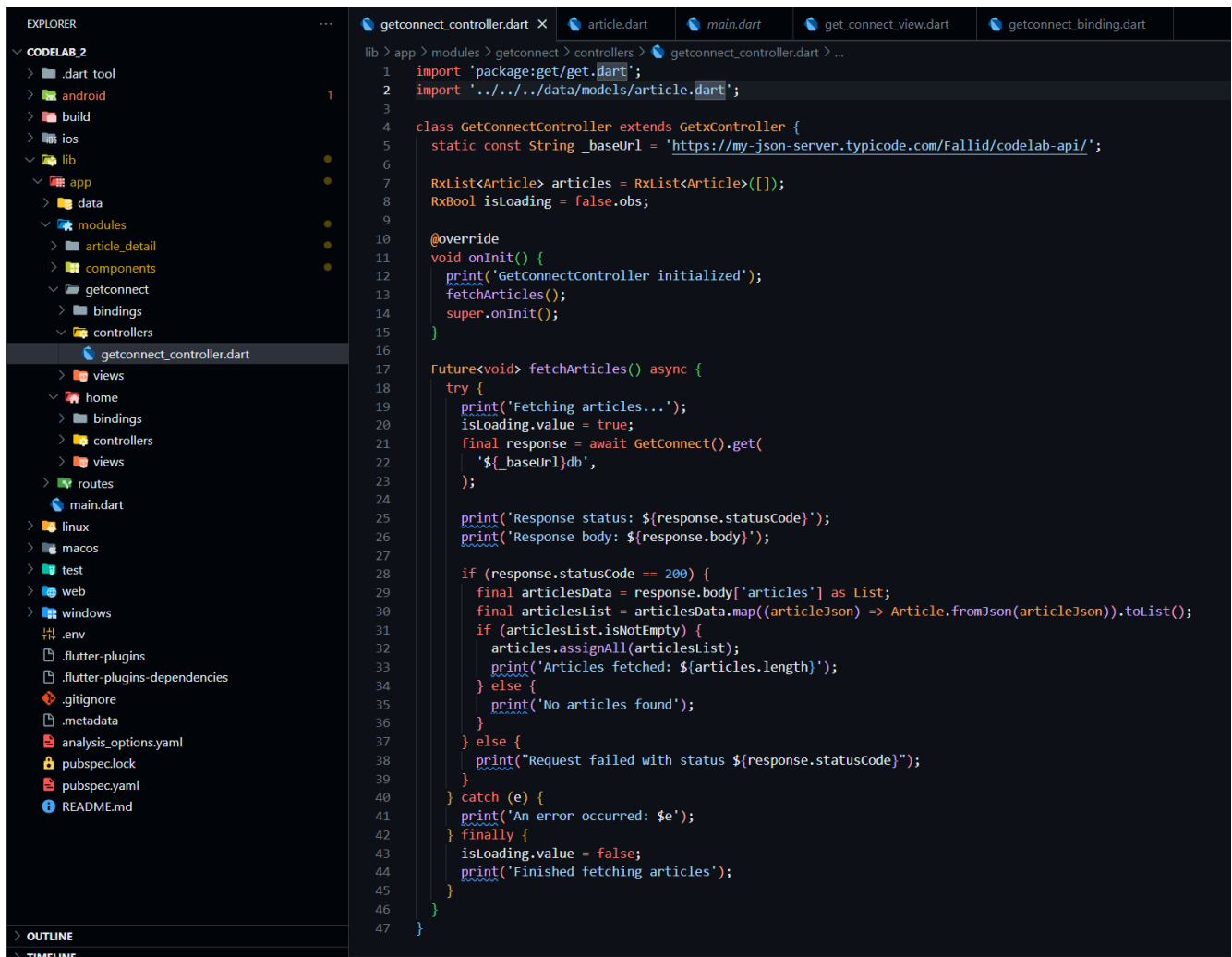


```
lib > app > data > models > article.dart > Welcome > Welcome.fromJson
1 // To parse this JSON data, do
2 //
3 //     final welcome = welcomeFromJson(jsonString);
4
5 import 'dart:convert';
6
7 Welcome welcomeFromJson(String str) => Welcome.fromJson(json.decode(str));
8
9 String welcomeToJson(Welcome data) => json.encode(data.toJson());
10
11 class Welcome {
12   Status status;
13   List<Article> articles;
14
15   Welcome({
16     required this.status,
17     required this.articles,
18   });
19
20   factory Welcome.fromJson(Map<String, dynamic> json) => Welcome(
21     status: Status.fromJson(json["status"]),
22     articles: List<Article>.from(json["articles"].map((x) => Article.fromJson(x))),
23   ); // Welcome
24
25   Map<String, dynamic> toJson() => {
26     "status": status.toJson(),
27     "articles": List<dynamic>.from(articles.map((x) => x.toJson())),
28   };
29
30   class Article {
31     int id;
32     String author;
33     String title;
34     String description;
35     String url;
36     String urlToImage;
37     DateTime publishedAt;
38     String content;
39
40     Article({
41       required this.id,
42       required this.author,
43       required this.title,
44       required this.description,
45       required this.url,
46       required this.urlToImage,
47       required this.publishedAt,
48       required this.content,
49     });
50
51     factory Article.fromJson(Map<String, dynamic> json) => Article(
52       id: json["id"],
53       author: json["author"],
54       title: json["title"],
55       description: json["description"],
56       url: json["url"],
57       urlToImage: json["urlToImage"],
58       publishedAt: DateTime.parse(json["publishedAt"]),
59       content: json["content"],
60     );
61
62     Map<String, dynamic> toJson() => {
63       "id": id,
64       "author": author,
65       "title": title,
66       "description": description,
67       "url": url,
68       "urlToImage": urlToImage,
69       "publishedAt": publishedAt.toIso8601String(),
70       "content": content,
71     };
72   }
73
74   class Status {
75     String status;
76
77     Status({
78       required this.status,
79     });
80
81     factory Status.fromJson(Map<String, dynamic> json) => Status(
82       status: json["status"],
83     );
84
85     Map<String, dynamic> toJson() => {
86       "status": status,
87     };
88   }
89 }
```


3. Buat folder pada path `"/lib/app/modules/"` dengan struktur seperti dibawah ini:



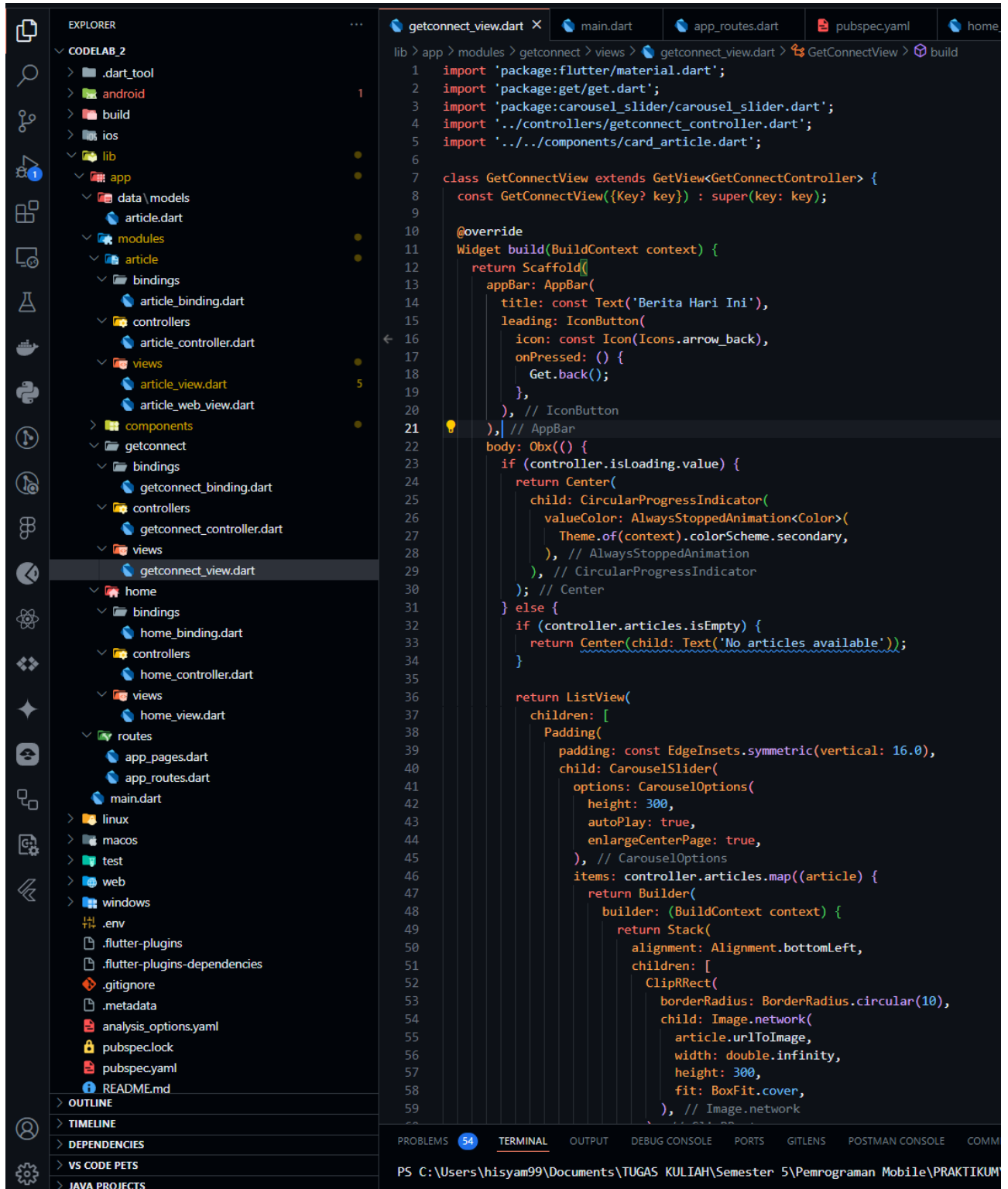
4. Buat `"getconnect_controller.dart"` dan taruh pada folder `"/lib/app/modules/getconnect/controllers/"`, untuk kodingannya seperti dibawah ini:



Perhatikan pada variable `_baseUrl`, pastikan value-nya kita ubah menjadi link api yang sudah kita siapkan sebelumnya, yaitu:

<https://my-json-server.typicode.com/Fallid/codelab-api/db>

5. Buat “getconnect_view.dart” dan taruh pada folder “/lib/app/modules/getconnect/views/”, untuk kodingannya seperti dibawah ini:



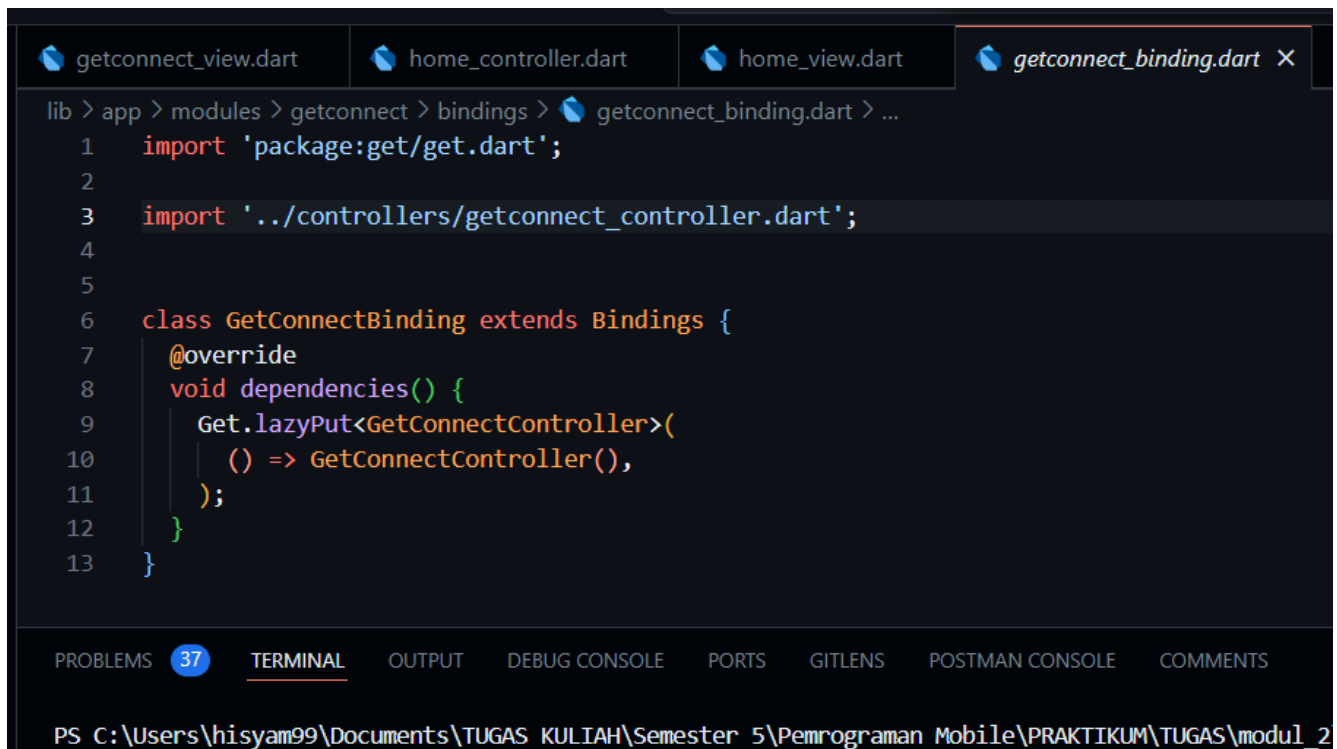
The screenshot shows the VS Code interface with the Explorer view on the left and the Editor view on the right. The Explorer view shows the project structure with the file `getconnect_view.dart` selected under the path `lib/app/modules/getconnect/views/`. The Editor view displays the code for `getconnect_view.dart`, which is a Flutter widget class extending `GetView<GetConnectController>`. The code includes imports for Flutter material, Get, carousel_slider, and local controllers and components. The `build` method returns a `Scaffold` with an `AppBar` and a `body` that uses `Obx` to check the loading state and display either a progress indicator, a message, or a carousel of articles.

```
lib > app > modules > getconnect > views > getconnect_view.dart > GetConnectView > build
1  import 'package:flutter/material.dart';
2  import 'package:get/get.dart';
3  import 'package:carousel_slider/carousel_slider.dart';
4  import '../controllers/getconnect_controller.dart';
5  import '../../components/card_article.dart';
6
7  class GetConnectView extends GetView<GetConnectController> {
8    const GetConnectView({Key? key}) : super(key: key);
9
10   @override
11   Widget build(BuildContext context) {
12     return Scaffold(
13       appBar: AppBar(
14         title: const Text('Berita Hari Ini'),
15         leading: IconButton(
16           icon: const Icon(Icons.arrow_back),
17           onPressed: () {
18             Get.back();
19           },
20         ), // IconButton
21     ), // AppBar
22     body: Obx(() {
23       if (controller.isLoading.value) {
24         return Center(
25           child: CircularProgressIndicator(
26             valueColor: AlwaysStoppedAnimation<Color>(
27               Theme.of(context).colorScheme.secondary,
28             ), // AlwaysStoppedAnimation
29           ), // CircularProgressIndicator
30         ); // Center
31       } else {
32         if (controller.articles.isEmpty) {
33           return Center(child: Text('No articles available'));
34         }
35
36         return ListView(
37           children: [
38             Padding(
39               padding: const EdgeInsets.symmetric(vertical: 16.0),
40               child: CarouselSlider(
41                 options: CarouselOptions(
42                   height: 300,
43                   autoplay: true,
44                   enlargeCenterPage: true,
45                 ), // CarouselOptions
46                 items: controller.articles.map((article) {
47                   return Builder(
48                     builder: (BuildContext context) {
49                       return Stack(
50                         alignment: Alignment.bottomLeft,
51                         children: [
52                           ClipRRect(
53                             borderRadius: BorderRadius.circular(10),
54                             child: Image.network(
55                               article.urlToImage,
56                               width: double.infinity,
57                               height: 300,
58                               fit: BoxFit.cover,
59                             ), // Image.network
60                         ],
61                       );
62                     },
63                   ),
64                 ),
65               ),
66             ),
67           ],
68         );
69       }
70     }));
71   }
```

*kode lengkapnya dapat dicek disini

https://github.com/hisyam99/PRAK-PEMROGRAMAN-MOBILE-2024/blob/main/TUGAS/modul_2/CODELAB/codelab_2/lib/app/modules/getconnect/views/getconnect_view.dart

6. Buat “getconnect_binding.dart” yang gunanya untuk menangani dependency injection dengan menginisialisasi controller atau layanan hanya ketika dibutuhkan pada rute tertentu. Taruh pada folder “/lib/app/modules/getconnect/bindings/”, untuk kodingannya seperti dibawah ini:

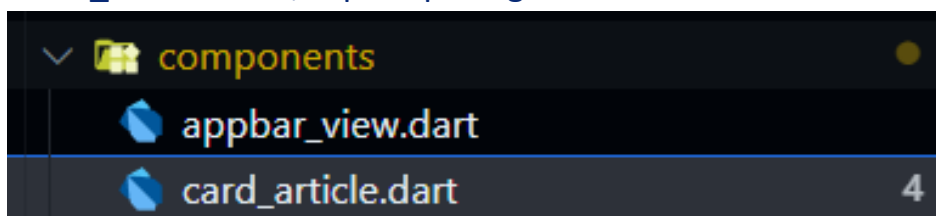


```
lib > app > modules > getconnect > bindings > getconnect_binding.dart > ...
1  import 'package:get/get.dart';
2
3  import '../controllers/getconnect_controller.dart';
4
5
6  class GetConnectBinding extends Bindings {
7    @override
8    void dependencies() {
9      Get.lazyPut<GetConnectController>(
10         () => GetConnectController(),
11      );
12    }
13  }
```

PROBLEMS 37 TERMINAL OUTPUT DEBUG CONSOLE PORTS GITLENS POSTMAN CONSOLE COMMENTS

PS C:\Users\hisyam99\Documents\TUGAS KULIAH\Semester 5\Pemrograman Mobile\PRAKTIKUM\TUGAS\modul_2

7. Buat folder “components” pada path “/lib/app/modules/”, lalu buat file “card_article.dart”, seperti pada gambar dibawah ini:



card_article.dart tersebut berguna sebagai komponen untuk menampilkan view kartu pada list article yang akan digunakan pada getconnect_view.dart nantinya.

8. Lalu masukkan kode berikut kedalam file "card_article.dart" tersebut:




```
lib > app > modules > components > card_article.dart > ...
1  import 'package:flutter/material.dart';
2  import 'package:get/get.dart';
3  import '../routes/app_pages.dart';
4  import '../data/models/article.dart';
5
6  class CardArticle extends StatelessWidget {
7    final Article article;
8    const CardArticle({Key? key, required this.article}) : super(key: key);
9
10   @override
11   Widget build(BuildContext context) {
12     return Card(
13       elevation: 4,
14       shape: RoundedRectangleBorder(
15         borderRadius: BorderRadius.circular(15),
16       ), // RoundedRectangleBorder
17       margin: const EdgeInsets.only(bottom: 20),
18       child: InkWell(
19         onTap: () {
20           Get.toNamed(Routes.ARTICLE_DETAILS, arguments: article);
21         },
22       borderRadius: BorderRadius.circular(15),
23       child: Column(
24         crossAxisAlignment: CrossAxisAlignment.start,
25         children: [
26           Hero(
27             tag: article.urlToImage ?? article.title,
28             child: ClipRRect(
29               borderRadius: const BorderRadius.only(
30                 topLeft: Radius.circular(15),
31                 topRight: Radius.circular(15),
32               ), // BorderRadius.only
33               child: article.urlToImage != null
34                 ? Image.network(
35                     article.urlToImage!,
36                     height: 180,
37                     width: double.infinity,
38                     fit: BoxFit.cover,
39                   ) // Image.network
40                 : Container(
41                     height: 180,
42                     width: double.infinity,
43                     color: Colors.grey[300],
44                     child: const Center(
45                       child: Text(
46                         'No Image Available',
47                         style: TextStyle(color: Colors.black45),
48                       ), // Text
49                     ), // Center
50                   ), // Container
51             ), // ClipRRect
52           ), // Hero
53           Padding(
54             padding: const EdgeInsets.all(16.0),
55             child: Column(
56               crossAxisAlignment: CrossAxisAlignment.start,
57               children: [
58                 Text(
59                   article.title,
```

*kode lengkapnya dapat dicek disini

https://github.com/hisyam99/PRAK-PEMROGRAMAN-MOBILE-2024/blob/main/TUGAS/modul_2/CODELAB/codelab_2/lib/app/modules/components/card_article.dart

9. Selanjutnya kita akan membuat “article_view.dart”, yang dimana akan menjadi view untuk menampilkan isi dari artikel yang sudah kita klik pada list di getconnect_view.dart, yang dimana saya sudah mengaturnya pada “card_article.dart” agar ketika card tersebut diklik, akan mengarah ke route ARTICLE_DETAILS, dengan kode berikut ini:

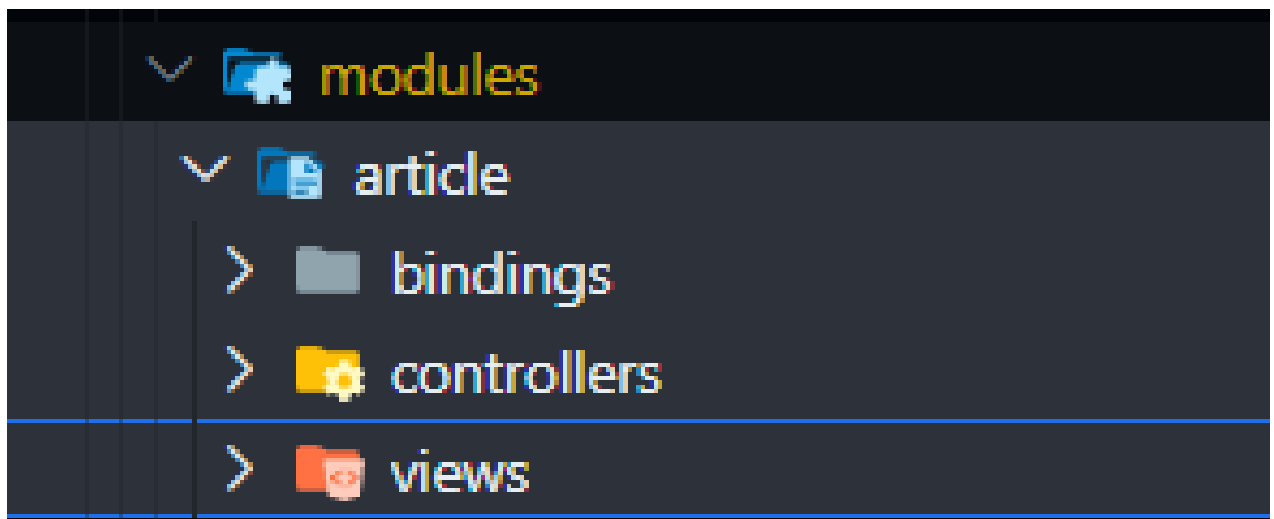


```
lib > app > modules > components > card_article.dart > CardArticle > build
6  class CardArticle extends StatelessWidget {
11  widget build(BuildContext context) {
19  ..... onTap: () {
20  ..... Get.toNamed(Routes.ARTICLE_DETAILS, arguments: article);
21  ..... },
22  ..... borderRadius: BorderRadius.circular(15),
23  ..... child: Column(
24  ..... crossAxisAlignment: CrossAxisAlignment.start,
```

PROBLEMS 58 TERMINAL OUTPUT DEBUG CONSOLE PORTS GITLENS POSTMAN CONSOLE

PS C:\Users\hisyam99\Documents\TUGAS KULIAH\Semester 5\Pemrograman Mobile\PRAKTIKUM\

Sebelumnya kita buat dulu folder bernama “article” pada path “/lib/app/modules/” dengan struktur seperti berikut ini:



10. Buat “article_view.dart” dan taruh pada folder “/lib/app/modules/article/views/”, untuk kodingannya seperti dibawah ini:

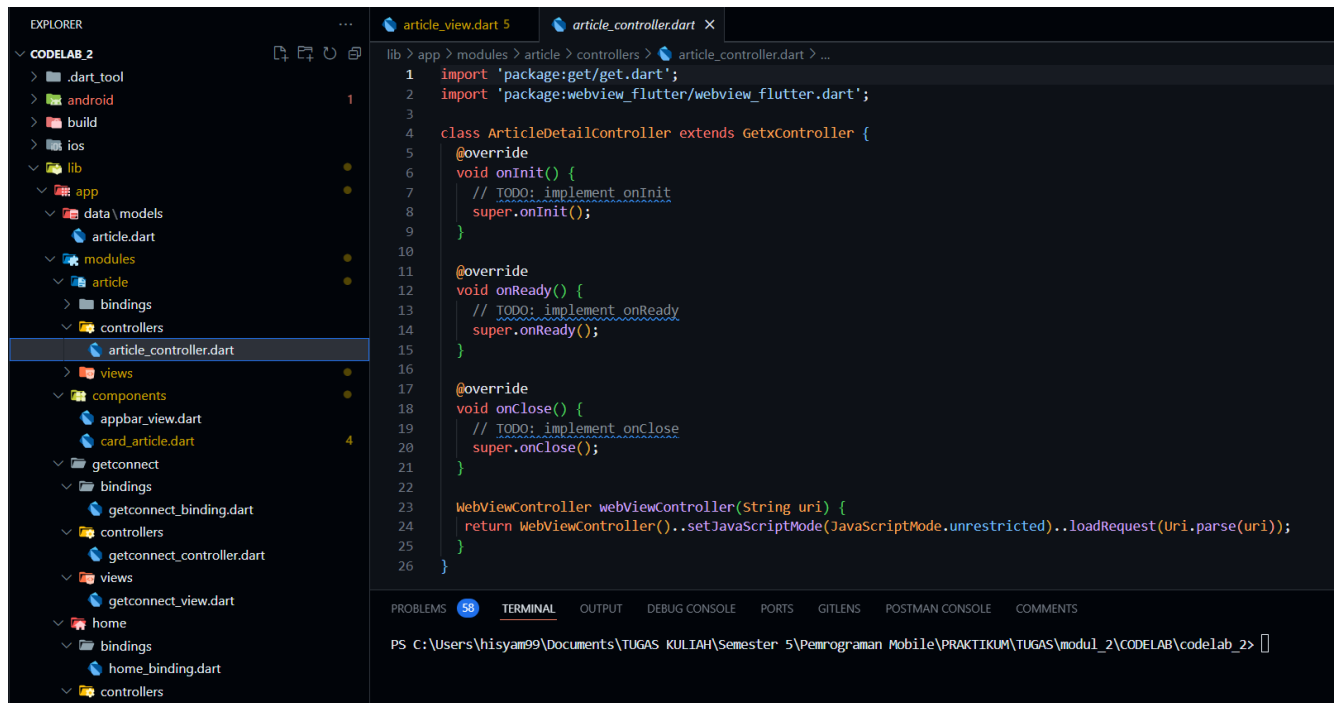


```
lib > app > modules > article > views > article_view.dart > ArticleDetailPage > build
1  import 'package:flutter/material.dart';
2  import 'package:get/get.dart';
3
4  import '../../data/models/article.dart';
5  import '../controllers/article_controller.dart';
6  import 'article_webview_view.dart';
7
8  class ArticleDetailPage extends GetView<ArticleDetailController> {
9    final Article article;
10    const ArticleDetailPage({Key? key, required this.article}) : super(key: key);
11
12    @override
13    Widget build(BuildContext context) {
14      return Scaffold(
15        appBar: AppBar(
16          title: const Text('Berita Hari Ini'),
17          centerTitle: true,
18          elevation: 0,
19        ), // AppBar
20        body: SingleChildScrollView(
21          child: Column(
22            crossAxisAlignment: CrossAxisAlignment.start,
23            children: [
24              Hero(
25                tag: article.urlToImage ?? article.title,
26                child: Container(
27                  height: 250,
28                  decoration: BoxDecoration(
29                    boxShadow: [
30                      BoxShadow(
31                        color: Colors.black26,
32                        blurRadius: 10,
33                        spreadRadius: 5,
34                        offset: Offset(0, 3), // Shadow position
35                      ), // BoxShadow
36                    ],
37                  ), // BoxDecoration
38                  child: ClipRect(
39                    borderRadius: const BorderRadius.only(
40                      bottomLeft: Radius.circular(15),
41                      bottomRight: Radius.circular(15),
42                    ), // BorderRadius.only
43                    child: article.urlToImage != null
44                      ? Image.network(
45                          article.urlToImage!,
46                          fit: BoxFit.cover,
47                          width: double.infinity,
48                        ) // Image.network
49                      : Container(
50                          color: Colors.grey[300],
51                          child: const Center(
52                            child: Text(
53                              'No Image Available',
54                              style: TextStyle(
55                                color: Colors.black45,
56                                fontSize: 16,
57                              ), // TextStyle
58                            ), // Text
59                          ), // Center
60                        ),
61                ),
62              ),
63            ],
64          ),
65        ),
66      );
67    }
68  }
```

*kode lengkapnya dapat dicek disini

https://github.com/hisyam99/PRAK-PEMROGRAMAN-MOBILE-2024/blob/main/TUGAS/modul_2/CODELAB/codelab_2/lib/app/modules/article/views/article_view.dart

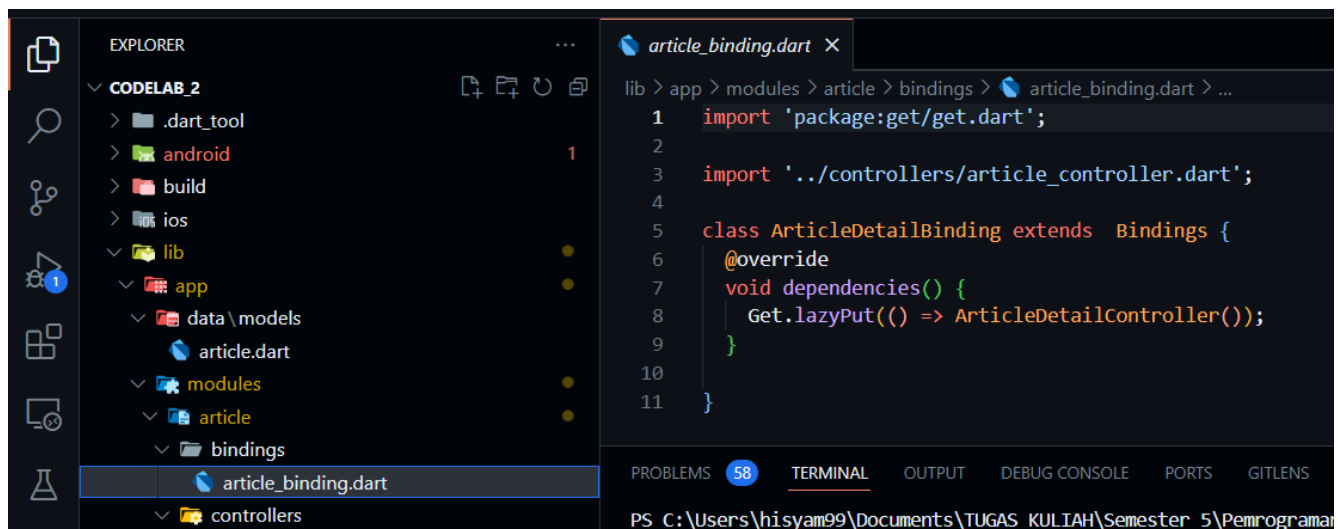
11. Buat “article_controller.dart” dan taruh pada folder “/lib/app/modules/article/controllers/”, untuk kodingannya seperti dibawah ini:



The screenshot shows the VS Code interface. On the left, the Explorer panel displays the project structure. The file `article_controller.dart` is selected under the path `lib > app > modules > article > controllers`. The main editor shows the code for `article_controller.dart`, which extends `GetxController` and implements `onInit`, `onReady`, and `onClose` methods. It also includes a `WebViewController` instance.

```
lib > app > modules > article > controllers > article_controller.dart > ...
1  import 'package:get/get.dart';
2  import 'package:webview_flutter/webview_flutter.dart';
3
4  class ArticleDetailController extends GetxController {
5    @override
6    void onInit() {
7      // TODO: implement onInit
8      super.onInit();
9    }
10
11    @override
12    void onReady() {
13      // TODO: implement onReady
14      super.onReady();
15    }
16
17    @override
18    void onClose() {
19      // TODO: implement onClose
20      super.onClose();
21    }
22
23    WebViewController webViewController(String uri) {
24      return WebViewController().setJavaScriptMode(JavaScriptMode.unrestricted)..loadRequest(Uri.parse(uri));
25    }
26  }
```

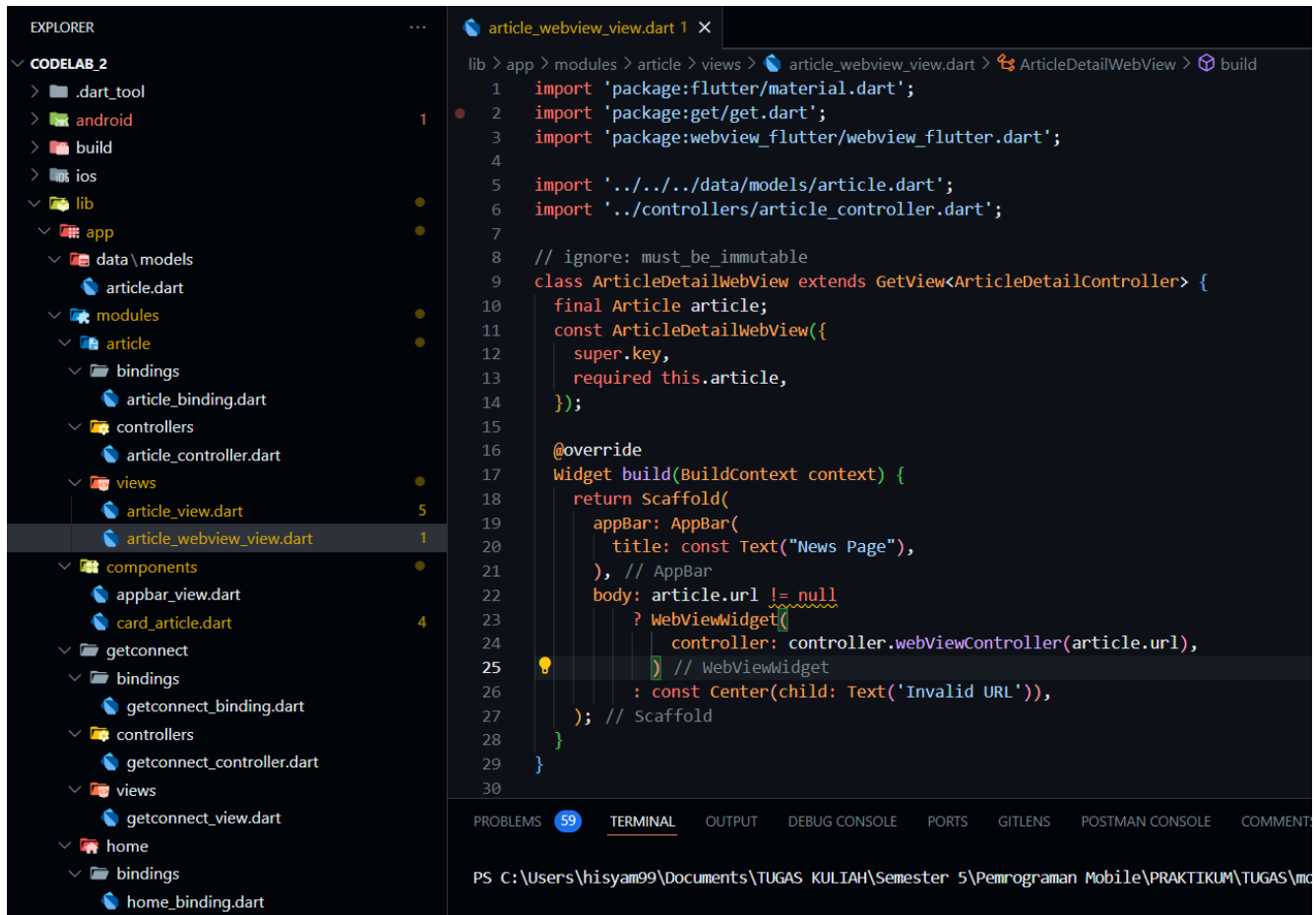
12. Buat “article_binding.dart” dan taruh pada folder “/lib/app/modules/article/bindings/”, untuk kodingannya seperti dibawah ini:



The screenshot shows the VS Code interface. On the left, the Explorer panel displays the project structure. The file `article_binding.dart` is selected under the path `lib > app > modules > article > bindings`. The main editor shows the code for `article_binding.dart`, which extends `Bindings` and implements the `dependencies` method to return an `ArticleDetailController`.

```
lib > app > modules > article > bindings > article_binding.dart > ...
1  import 'package:get/get.dart';
2
3  import '../controllers/article_controller.dart';
4
5  class ArticleDetailBinding extends Bindings {
6    @override
7    void dependencies() {
8      Get.lazyPut(() => ArticleDetailController());
9    }
10
11  }
```

13. Langkah selanjutnya saya akan membuat file "article_webview_view.dart", yang dimana ini adalah penerapan WebView saya yang berguna agar masuk ke halaman web Ketika saya mengklik tombol "Lihat Artikel Selengkapnya". Taruh file tersebut pada folder "/lib/app/modules/article/views/". Kodenya sebagai berikut:



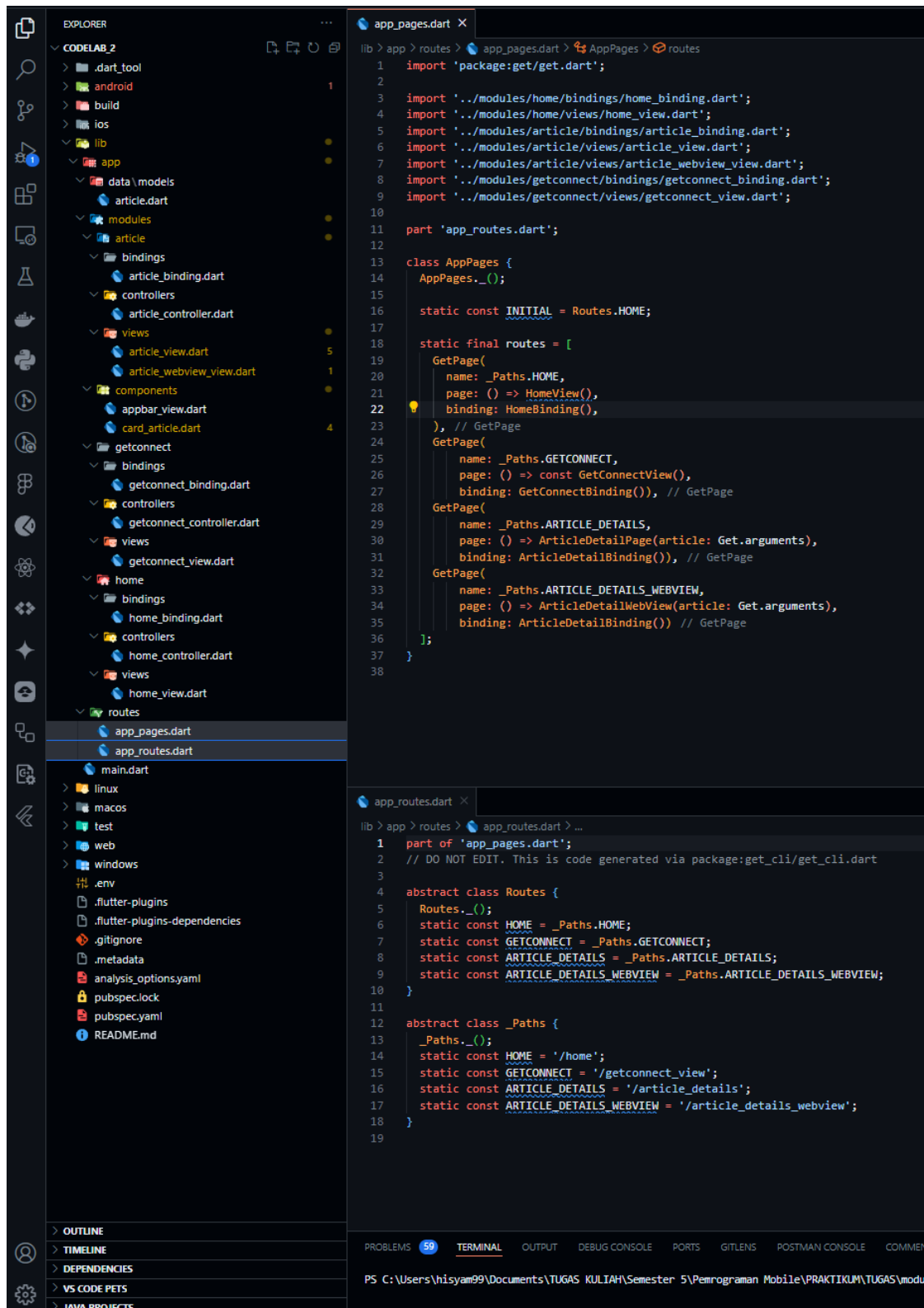
The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer shows the project structure, with the file `article_webview_view.dart` selected in the `views` folder. The code editor shows the implementation of the `ArticleDetailWebView` class, which extends `GetView<ArticleDetailController>`. The class has a `final Article article` property and a `build` method that returns a `Scaffold` widget. The `Scaffold` has an `AppBar` with the title "News Page" and a `body` that is a `WebViewWidget` if the `article.url` is not null, or a `Center` widget with the text "Invalid URL" otherwise.

```
lib > app > modules > article > views > article_webview_view.dart > ArticleDetailWebView > build
1  import 'package:flutter/material.dart';
2  import 'package:get/get.dart';
3  import 'package:webview_flutter/webview_flutter.dart';
4
5  import '../data/models/article.dart';
6  import '../controllers/article_controller.dart';
7
8  // ignore: must_be_immutable
9  class ArticleDetailWebView extends GetView<ArticleDetailController> {
10     final Article article;
11     const ArticleDetailWebView({
12       super.key,
13       required this.article,
14     });
15
16     @override
17     Widget build(BuildContext context) {
18       return Scaffold(
19         appBar: AppBar(
20           title: const Text("News Page"),
21         ), // AppBar
22         body: article.url != null
23           ? WebViewWidget(
24             controller: controller.webViewController(article.url),
25           ) // WebViewWidget
26           : const Center(child: Text('Invalid URL')),
27       ); // Scaffold
28     }
29   }
30
```

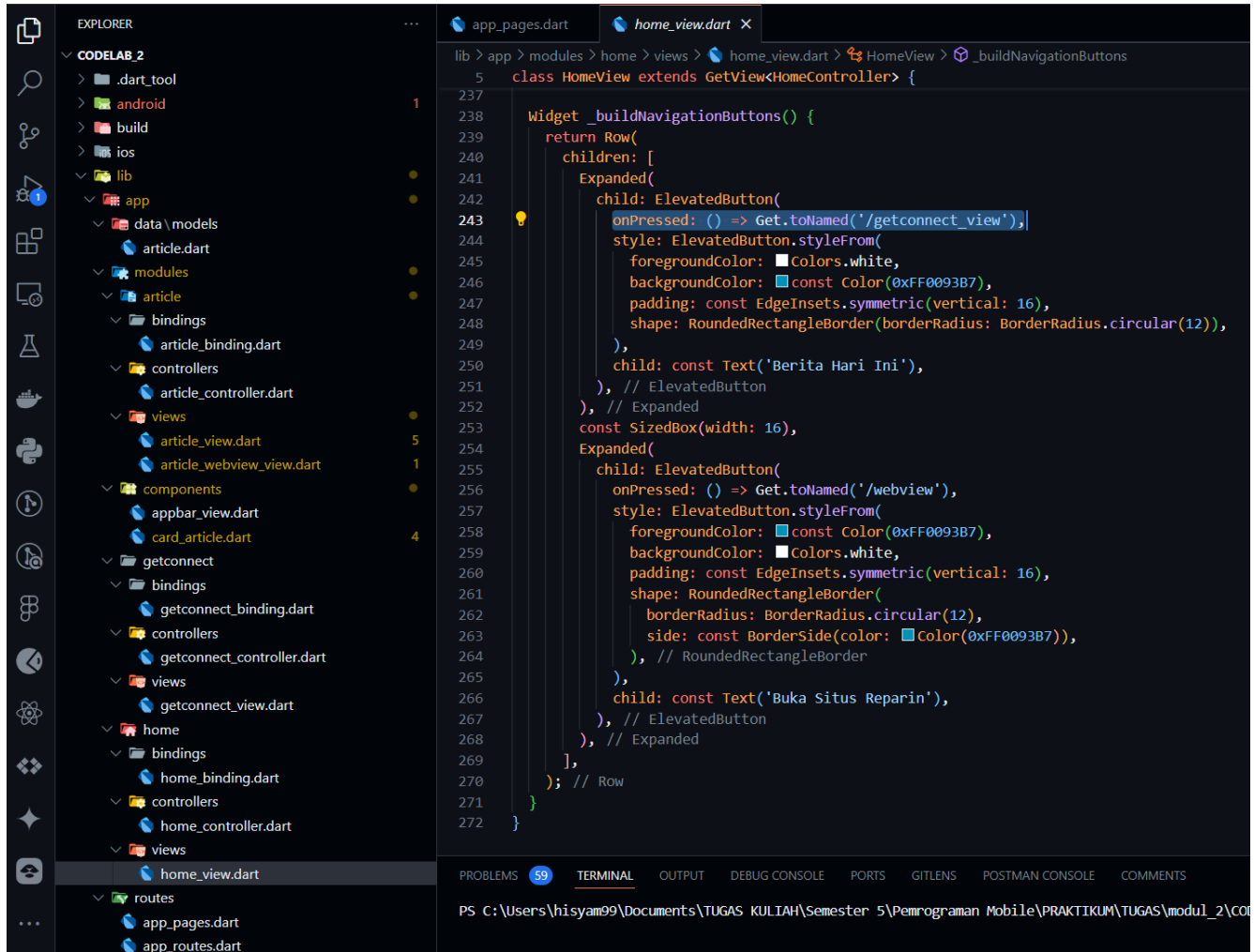
PROBLEMS 59 TERMINAL OUTPUT DEBUG CONSOLE PORTS GITLENS POSTMAN CONSOLE COMMENTS

PS C:\Users\hisyam99\Documents\TUGAS KULIAH\Semester 5\Pemrograman Mobile\PRAKTIKUM\TUGAS\mo

14. Lakukan modifikasi pada "app_pages.dart" dan "app_routes.dart" dengan menambahkan page untuk list artikel, detail artikel, dan artikel webview. Seperti pada gambar berikut ini:



15. Modifikasi “home_view.dart” agar memiliki tombol yang dapat mengarah ke route getconnect_view, yaitu dengan kode sebagai berikut:



The screenshot shows an IDE with the Explorer panel on the left displaying the project structure. The main editor shows the `home_view.dart` file. The code defines a `HomeView` class extending `GetView<HomeController>`. The `Widget _buildNavigationButtons()` method returns a `Row` of two `ElevatedButton` widgets. The first button is labeled "Berita Hari Ini" and has an `onPressed` handler that calls `Get.toNamed('/getconnect_view')`. The second button is labeled "Buka Situs Reparin" and has an `onPressed` handler that calls `Get.toNamed('/webview')`. The buttons are styled with a white foreground color, a blue background color, and a rounded rectangular border.

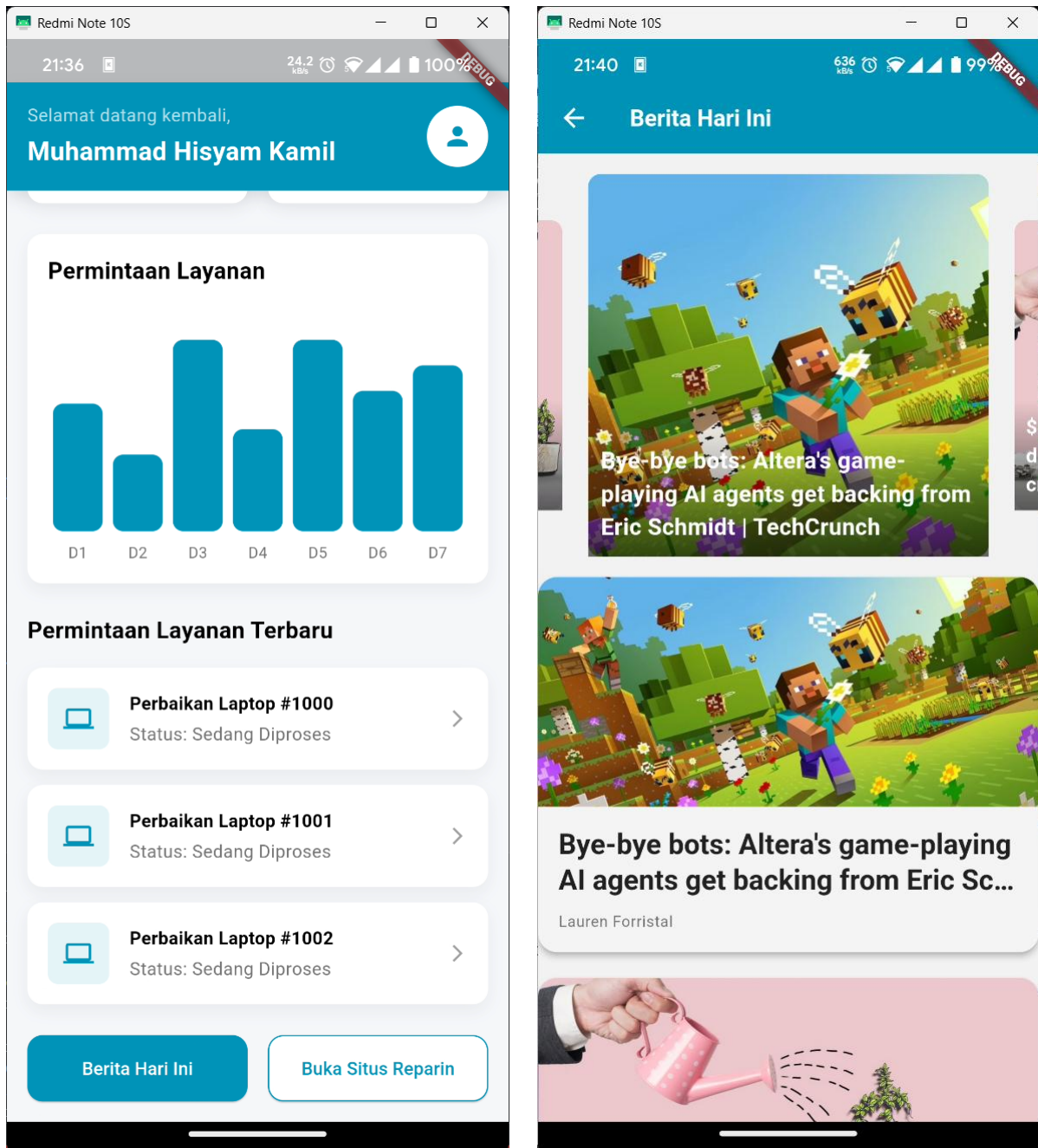
```
lib > app > modules > home > views > home_view.dart > HomeView > _buildNavigationButtons
5 class HomeView extends GetView<HomeController> {
237
238   Widget _buildNavigationButtons() {
239     return Row(
240       children: [
241         Expanded(
242           child: ElevatedButton(
243             onPressed: () => Get.toNamed('/getconnect_view'),
244             style: ElevatedButton.styleFrom(
245               foregroundColor: Colors.white,
246               backgroundColor: const Color(0xFF0093B7),
247               padding: const EdgeInsets.symmetric(vertical: 16),
248               shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(12)),
249             ),
250             child: const Text('Berita Hari Ini'),
251           ), // ElevatedButton
252         ), // Expanded
253         const SizedBox(width: 16),
254         Expanded(
255           child: ElevatedButton(
256             onPressed: () => Get.toNamed('/webview'),
257             style: ElevatedButton.styleFrom(
258               foregroundColor: const Color(0xFF0093B7),
259               backgroundColor: Colors.white,
260               padding: const EdgeInsets.symmetric(vertical: 16),
261               shape: RoundedRectangleBorder(
262                 borderRadius: BorderRadius.circular(12),
263                 side: const BorderSide(color: Color(0xFF0093B7)),
264               ), // RoundedRectangleBorder
265             ),
266             child: const Text('Buka Situs Reparin'),
267           ), // ElevatedButton
268         ), // Expanded
269       ],
270     ); // Row
271   }
272 }
```

Disini saya tidak akan menjelaskannya secara detail terkait cara membuat desain home_viewnya, intinya kita memanggil `ElevatedButton`, lalu `onPressed` kita atur agar memanggil `() => Get.toNamed('/getconnect_view')`”.

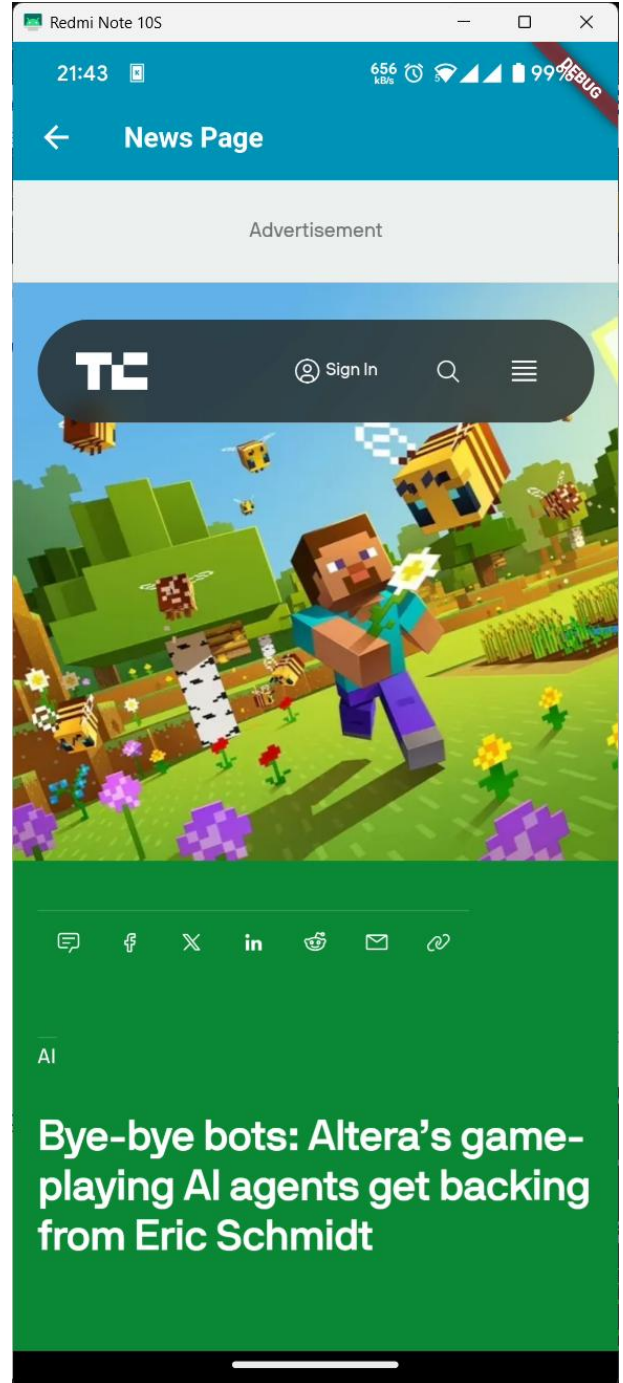
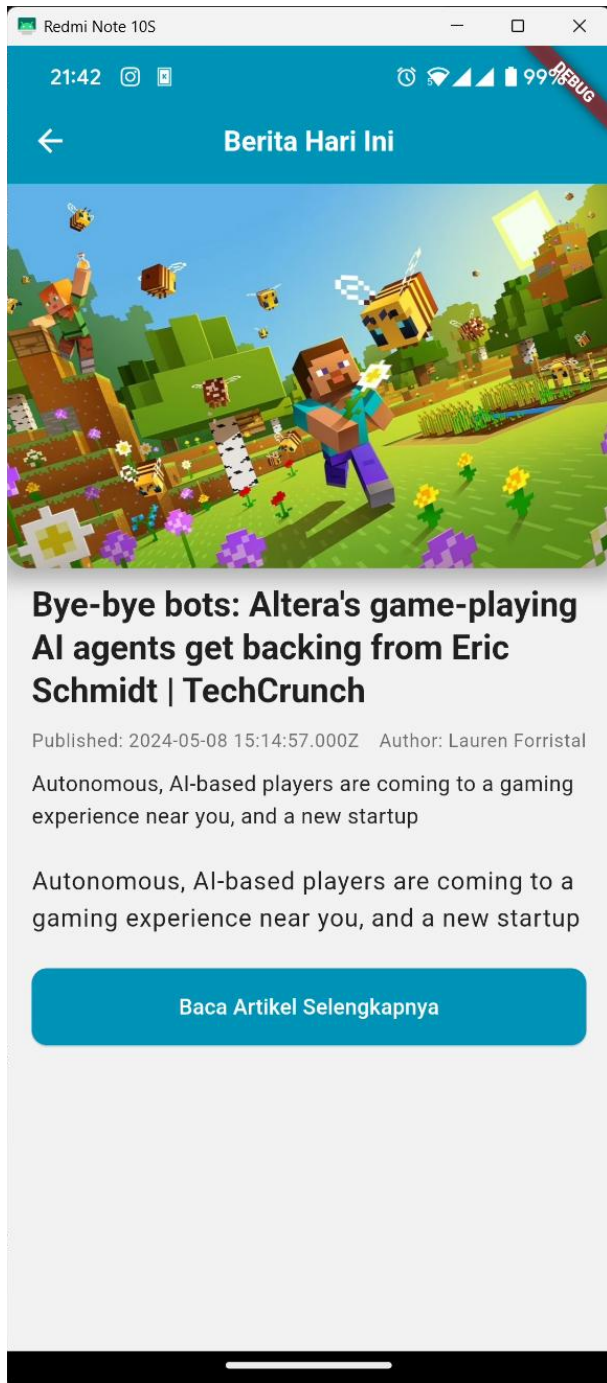
*kode lengkapnya dapat dicek disini

https://github.com/hisyam99/PRAK-PEMROGRAMAN-MOBILE-2024/blob/main/TUGAS/modul_2/CODELAB/codelab_2/lib/app/modules/home/views/home_view.dart

IV. Tampilan Aplikasi setelah penerapan GetConnect, dan WebView



Tampilan HomeView > klik Berita Hari Ini > Tampilan getconnect_view.dart dan card_article.dart



Tampilan article_view.dart > klik Baca Artikel Selengkapnya > Tampilan article_webview_view.dart