

# PRAKTIKUM 5

---

## FLUTTER HTTP

---

### A. TUJUAN PEMBELAJARAN

1. Mahasiswa dapat membuat aplikasi flutter yang terhubung dengan server

### B. DASAR TEORI

Dasar teori.

### C. TUGAS PENDAHULUAN

1. Siapkan project Tourism Place yang telah dikerjakan pada praktikum sebelumnya.
2. Check aplikasi Tourism Place dan pastikan semua fitur yang ada pada praktikum sebelumnya telah dikerjakan
3. Jalankan aplikasi Tourism Place dan cobalah semua fitur yang ada. Pastikan semua fitur dapat berjalan dengan baik.

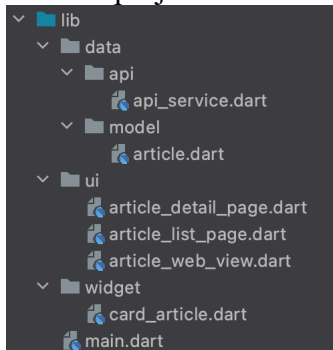
## D. PERCOBAAN

### Percobaan 1 : State Management menggunakan setState()

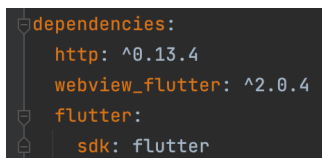
Pada percobaan ini kita akan membuat aplikasi portal berita yang datanya berasal dari internet atau API. Untuk API yang akan digunakan adalah API publik yang bisa diakses pada laman: <https://newsapi.org/>.

Langkah Percobaan:

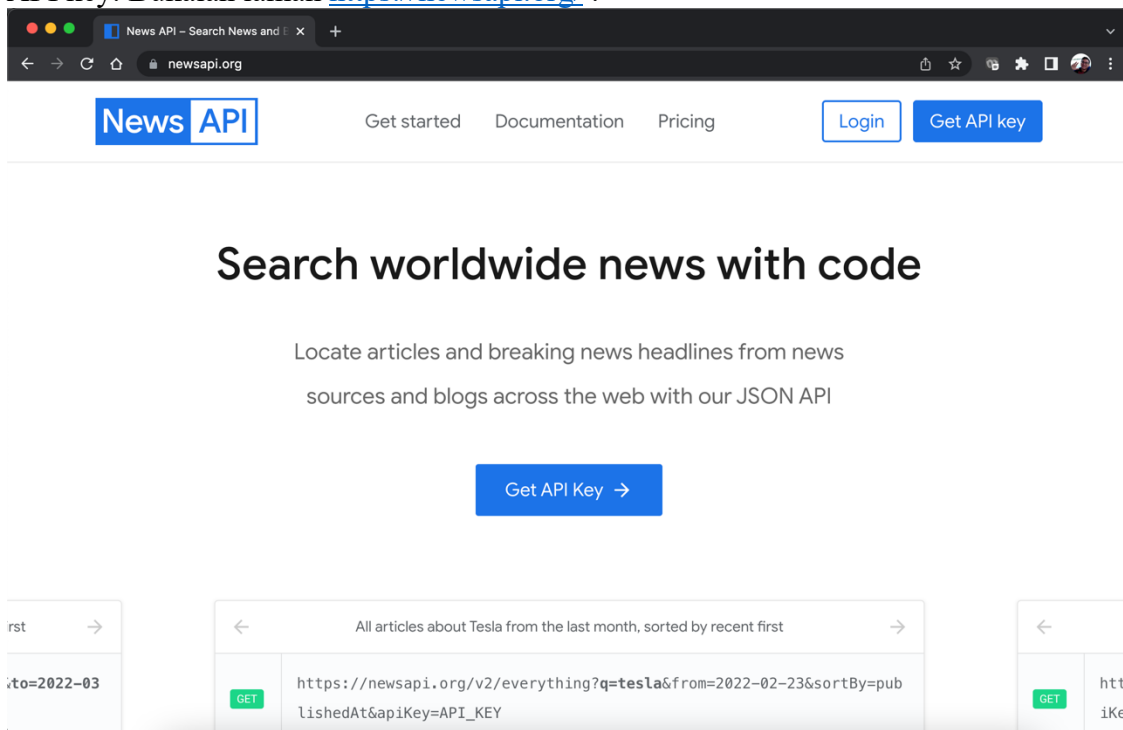
1. Buatlah project baru dan siapkan beberapa direktori dan berkas dart seperti berikut:



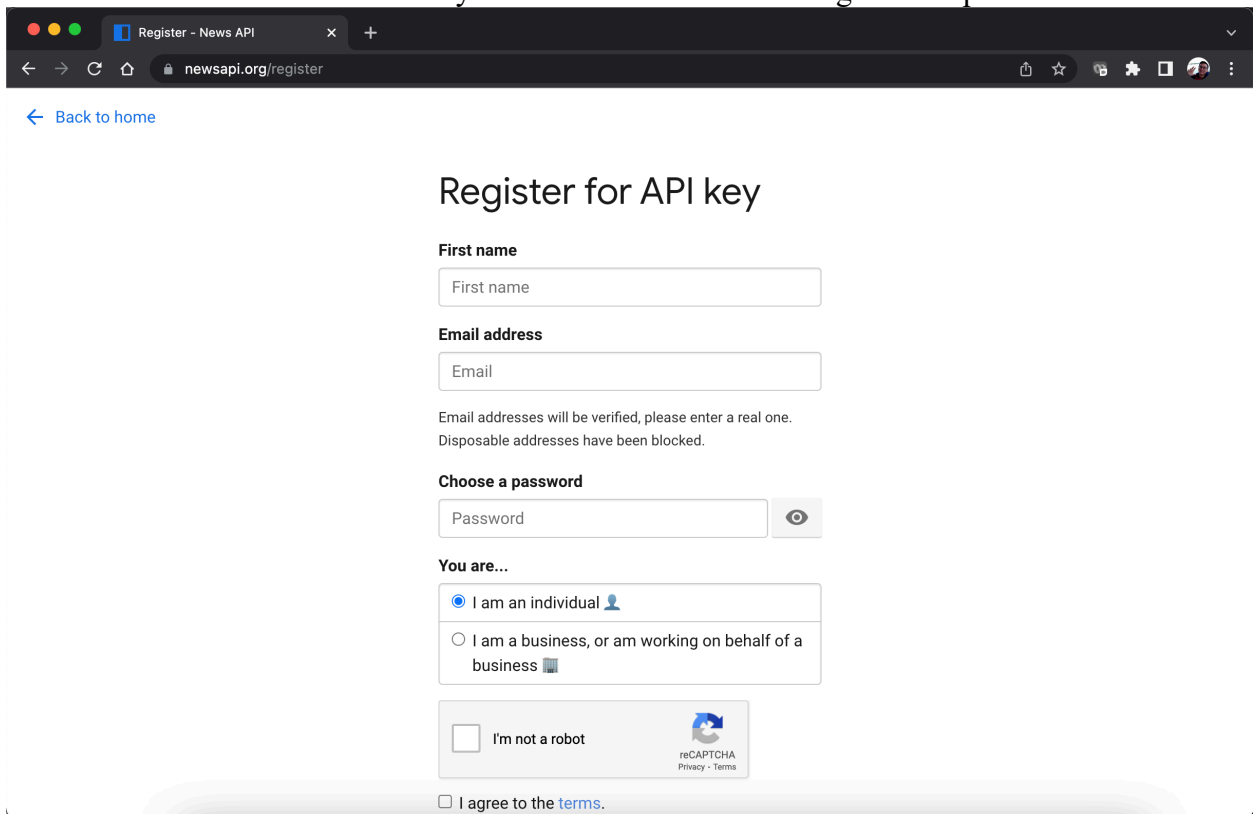
2. Kemudian tambahkan dependency http pada `pubspec.yaml`



3. Sebelum memulai membuat kode program, kita perlu membuka API publik yang akan kita gunakan. Untuk menggunakan API kita perlu mendaftar terlebih dahulu untuk mendapatkan API key. Bukalah laman <https://newsapi.org/> :



4. Kemudian klik tombol Get API Key untuk membuka halaman registrasi seperti berikut:



The screenshot shows a web browser window with the address bar displaying "newsapi.org/register". The page has a dark header with a "Back to home" link. The main content area is titled "Register for API key". It contains several form fields: "First name", "Email address", and "Choose a password". Below the email field, there is a note: "Email addresses will be verified, please enter a real one. Disposable addresses have been blocked." Under the password field, there is a "You are..." section with two radio buttons: "I am an individual" (selected) and "I am a business, or am working on behalf of a business". At the bottom, there is a checkbox for "I'm not a robot" with a reCAPTCHA logo, and a checkbox for "I agree to the terms".

Register for API key

First name

Email address

Email addresses will be verified, please enter a real one.  
Disposable addresses have been blocked.

Choose a password

You are...

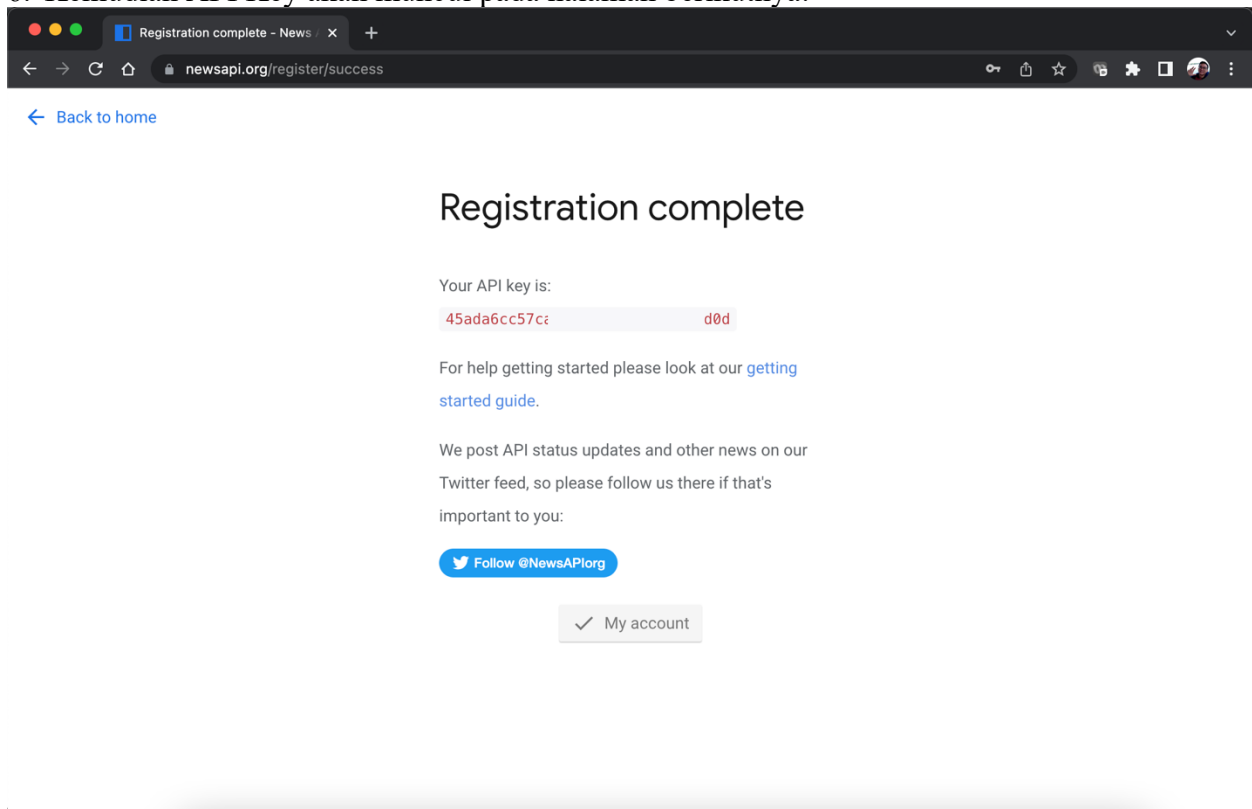
☒ I am an individual

☐ I am a business, or am working on behalf of a business

☐ I'm not a robot

☐ I agree to the [terms](#).

5. Silahkan isi kolom first name, email address, choose password, kemudian pilih I am an individual, verifikasi captcha, klik tombol submit.
6. Kemudian API Key akan muncul pada halaman berikutnya:



The screenshot shows a web browser window with the address bar displaying "newsapi.org/register/success". The page has a dark header with a "Back to home" link. The main content area is titled "Registration complete". It displays the API key: "45ada6cc57c2d0d". Below the API key, there is a link to the "getting started guide". At the bottom, there is a button to "Follow @NewsAPIorg" and a button to "My account".

Registration complete

Your API key is:

45ada6cc57c2d0d

For help getting started please look at our [getting started guide](#).

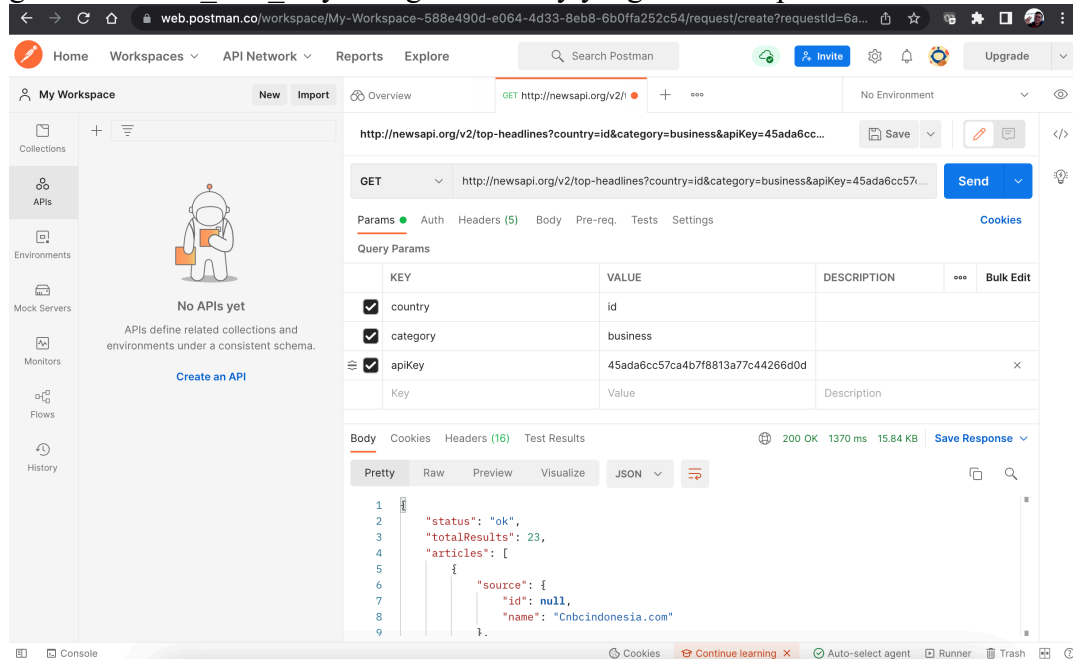
We post API status updates and other news on our Twitter feed, so please follow us there if that's important to you:

[Follow @NewsAPIorg](#)

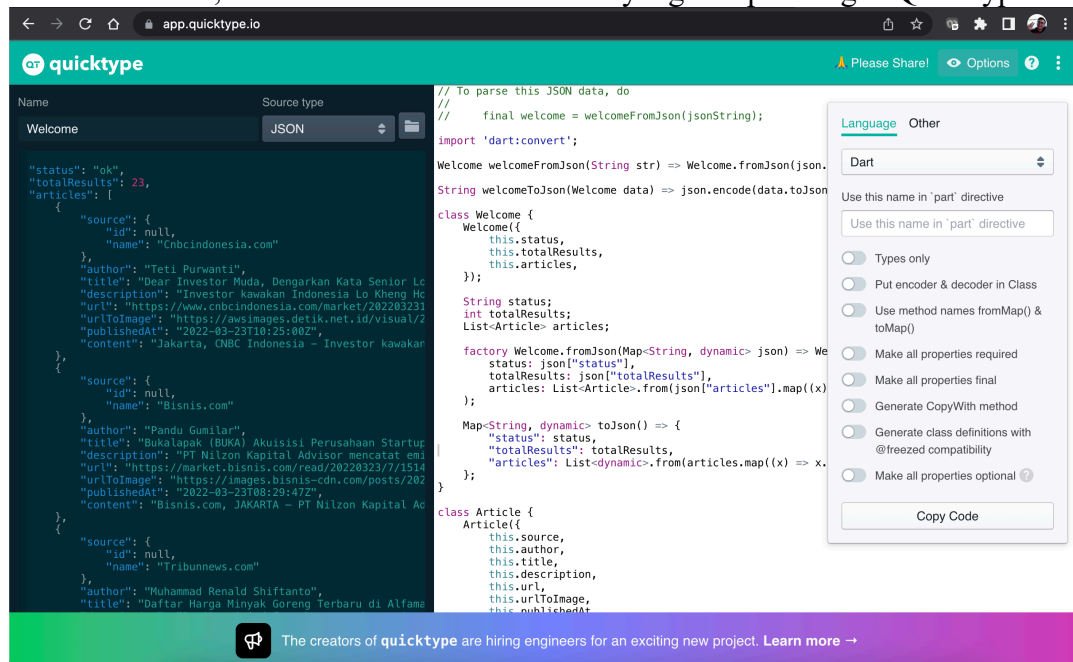
✓ My account

7. Sekarang kita coba API Key yang telah didapat. Bukalah postman dan masukkan url berikut untuk mendapatkan json dari API.

`http://newsapi.org/v2/top-headlines?country=id&category=business&apiKey=Your_API_Key`  
gantilah "Your\_API\_Key" dengan API Key yang telah didapatkan.



8. Setelah berhasil, kemudian kita konversi JSON yang didapat dengan Quicktype.io



9. Kemudian copy dan paste kelas model tersebut ke dalam file article.dart  
Catatan: karena kelas model tersebut hasil dari konversi bukan coding manual, akan terdapat beberapa variabel yang tidak cocok dengan tipe datanya. Kemudian hasil dari konversi juga

terkadang membuat beberapa fungsi yang sebenarnya tidak akan kita pakai atau implementasikan seperti `articlesResultFromJson()`, `articlesResultToJson()`, `Source`, dan yang berakhiran `.toJson()`. Anda dapat menghapusnya jika tidak menggunakan beberapa fungsi tersebut agar kode Anda menjadi lebih ringkas dan bersih

Berikut adalah hasilnya:

```
1  class ArticlesResult {
2      ArticlesResult({
3          required this.status,
4          required this.totalResults,
5          required this.articles,
6      });
7
8      String status;
9      int totalResults;
10     List<Article> articles;
11
12     factory ArticlesResult.fromJson(Map<String, dynamic> json) => ArticlesResult(
13         status: json["status"],
14         totalResults: json["totalResults"],
15         articles: List<Article>.from(
16             json["articles"].map((x) => Article.fromJson(x))), // List.from
17     ); // ArticlesResult
18 }
19
20 class Article {
21     Article({
22         required this.author,
23         required this.title,
24         required this.description,
25         required this.url,
26         required this.urlToImage,
27         required this.publishedAt,
28         required this.content,
29     });
30
31     String? author;
32     String title;
33     String? description;
34     String url;
35     String? urlToImage;
36     DateTime? publishedAt;
37     String? content;
38
39     factory Article.fromJson(Map<String, dynamic> json) => Article(
40         author: json["author"],
41         title: json["title"],
42         description: json["description"],
43         url: json["url"],
44         urlToImage: json["urlToImage"],
45         publishedAt: DateTime.parse(json["publishedAt"]),
46         content: json["content"],
47     );
48 }
```

10. Jika anda perhatikan data yang dikembalikan dari API, akan ada beberapa data yang tidak lengkap dan bernilai null. Apabila data ini tidak lengkap, kita bisa memfilternya saja agar tidak ditampilkan dalam aplikasi. Pada factory method, tambahkan fungsi where untuk menyaring property tertentu yang tidak boleh null

```
12 factory ArticlesResult.fromJson(Map<String, dynamic> json) => ArticlesResult(  
13     status: json["status"],  
14     totalResults: json["totalResults"],  
15     articles: List<Article>.from((json["articles"] as List)  
16         .map((x) => Article.fromJson(x))  
17         .where((article) =>  
18             article.author != null &&  
19             article.urlToImage != null &&  
20             article.publishedAt != null &&  
21             article.content != null)), // List.from  
22 ); // ArticlesResult
```

11. Setelah kita membuat model, langkah selanjutnya kita akan membuat kelas baru bernama `ApiService`. Di dalam kelas tersebut kita akan membuat fungsi Future yang bersifat async untuk menghubungkan antara aplikasi Flutter (klien) dengan data yang bersumber dari API (server). Berikut adalah kelas yang akan dibuat pada berkas `api_service.dart`. Jangan lupa mengganti value dari `_apiKey` sesuai dengan API Key yang didapatkan dari <https://newsapi.org>

```
1 import 'dart:convert';  
2 import 'package:http/http.dart' as http;  
3  
4 import '../model/article.dart';  
5  
6 class ApiService {  
7     static final String _baseUrl = 'https://newsapi.org/v2/';  
8     static final String _apiKey = 'YOUR_API_KEY';  
9     static final String _category = 'business';  
10    static final String _country = 'id';  
11  
12    Future<ArticlesResult> topHeadlines() async {  
13        final response = await http.get(Uri.parse(_baseUrl +  
14            "top-headlines?country=$_country&category=$_category&apiKey=$_apiKey"));  
15        if (response.statusCode == 200) {  
16            return ArticlesResult.fromJson(json.decode(response.body));  
17        } else {  
18            throw Exception('Failed to load top headlines');  
19        }  
20    }  
21 }
```

12. Kemudian pada berkas `card_article.dart` buatlah kelas dengan nama `CardArticle` dengan meng-extends kelas `StatelessWidget`. Berikut adalah kodenya:

```
9 class CardArticle extends StatelessWidget {
10   final Article article;
11
12   const CardArticle({required this.article});
13
14   @override
15   Widget build(BuildContext context) {
16     return Material(
17       color: Colors.white,
18       child: ListTile(
19         contentPadding:
20           const EdgeInsets.symmetric(horizontal: 16.0, vertical: 8.0),
21         leading: Hero(
22           tag: article.urlToImage!,
23           child: Image.network(
24             article.urlToImage!,
25             width: 100,
26           ), // Image.network
27         ), // Hero
28         title: Text(
29           article.title,
30         ), // Text
31         subtitle: Text(article.author!),
32         onTap: () => Navigator.pushNamed(
33           context,
34           ArticleDetailPage.routeName,
35           arguments: article,
36         ),
37       ), // ListTile
38     ); // Material
39   }
40 }
```

Jika kita lihat pada kode di atas, kita menambahkan bang operator (!) pada variabel `article.urlToImage`. Ini karena kita mendefinisikan properti `urlToImage` sebagai nullable pada kelas `Article`. Namun, kita telah memfilter url image yang null ketika proses parsing, sehingga pada proses ini kita bisa yakin bahwa data yang ditampilkan tidak akan null.

13. Setelah membuat widget `CardArticle` yang nantinya akan digunakan sebagai item dari list portal berita, selanjutnya kita akan melakukan beberapa perubahan pada berkas `article_list_page.dart`. Pertama, ubahlah kelas `ArticleListPage` menjadi turunan dari kelas `StatefulWidget`. Berikut adalah kode untuk berkas tersebut:

```
7 class ArticleListPage extends StatefulWidget {
8   @override
9   _ArticleListPageState createState() => _ArticleListPageState();
10 }
11
12 class _ArticleListPageState extends State<ArticleListPage> {
13   late Future<ArticlesResult> _article;
14
15   @override
16   void initState() {
17     super.initState();
18     _article = ApiService().topHeadlines();
19   }
20
21   Widget _buildList(BuildContext context) {
22     return FutureBuilder(
23       future: _article,
24       builder: (context, AsyncSnapshot<ArticlesResult> snapshot) {
25         var state = snapshot.connectionState;
26         if (state != ConnectionState.done) {
27           return Center(child: CircularProgressIndicator());
28         } else {
29           if (snapshot.hasData) {
30             return ListView.builder(
31               shrinkWrap: true,
32               itemCount: snapshot.data?.articles.length,
33               itemBuilder: (context, index) {
34                 var article = snapshot.data?.articles[index];
35                 return CardArticle(article: article!);
36               },
37             ); // ListView.builder
38           } else if (snapshot.hasError) {
39             return Center(child: Text(snapshot.error.toString()));
40           } else {
41             return Text('');
42           }
43         }
44       },
45     ); // FutureBuilder
46   }
47
48   @override
49   Widget build(BuildContext context) {
50     return Scaffold(
51       appBar: AppBar(
52         title: Text('News App'),
53       ), // AppBar
54       body: _buildList(context),
55     ); // Scaffold
56   }
57 }
```



14. Selanjutnya kita akan melakukan perubahan pada halaman detail pada berkas `article_detail_page.dart`. Berikut adalah kodenya:

```
6 class ArticleDetailPage extends StatelessWidget {
7   static const routeName = '/article_detail';
8
9   final Article article;
10
11   const ArticleDetailPage(this.article);
12
13   @override
14   Widget build(BuildContext context) {
15     return Scaffold(
16       appBar: AppBar(
17         title: Text('News App'),
18       ), // AppBar
19       body: SingleChildScrollView(
20         child: Column(
21           children: [
22             Hero(
23               tag: article.urlToImage!,
24               child: Image.network(article.urlToImage!),
25             ), // Hero
26             Padding(
27               padding: EdgeInsets.all(10),
28               child: Column(
29                 crossAxisAlignment: CrossAxisAlignment.start,
30                 children: [
31                   Text(
32                     article.description,
33                     style: Theme.of(context).textTheme.bodyText2,
34                     style: Theme.of(context).textTheme.bodyText2,
35                   ), // Text
36                   Divider(color: Colors.grey),
37                   Text(
38                     article.title,
39                     style: Theme.of(context).textTheme.headline6,
40                   ), // Text
41                   Divider(color: Colors.grey),
42                   Text(
43                     'Date: ${article.publishedAt}',
44                     style: Theme.of(context).textTheme.caption,
45                   ), // Text
46                   SizedBox(height: 10),
47                   Text(
48                     'Author: ${article.author}',
49                     style: Theme.of(context).textTheme.caption,
50                   ), // Text
51                   Divider(color: Colors.grey),
52                   Text(
53                     article.content ?? "",
54                     style: Theme.of(context).textTheme.bodyText1,
55                   ), // Text
56                   SizedBox(height: 10),
57                   ElevatedButton(
58                     child: Text('Read more'),
59                     onPressed: () {
60                       Navigator.pushNamed(context, ArticleWebView.routeName,
61                         arguments: article.url);
62                     },
63                   ), // ElevatedButton
64                 ],
65               ), // Column
66             ), // Padding
67           ],
68         ), // Column
69       ), // SingleChildScrollView
70     ); // Scaffold
71   }
```

15. Buatlah class `ArticleWebView` pada file `article web view.dart`

```
1  import 'package:flutter/material.dart';
2  import 'package:webview_flutter/webview_flutter.dart';
3
4  class ArticleWebView extends StatelessWidget {
5      static const routeName = '/article_web';
6
7      final String url;
8
9      const ArticleWebView({required this.url});
10
11     @override
12     Widget build(BuildContext context) {
13         return Scaffold(
14             appBar: AppBar(
15                 title: Text('News App'),
16             ), // AppBar
17             body: WebView(
18                 initialUrl: url,
19             ), // WebView
20         ); // Scaffold
21     }
22 }
```

16. Terakhir ubahlah `main.dart` sebagai berikut:

```
7  >> void main() {
8      runApp(const MyApp());
9  }
10
11  class MyApp extends StatelessWidget {
12      const MyApp({Key? key}) : super(key: key);
13
14      @override
15      Widget build(BuildContext context) {
16          return MaterialApp(
17              title: 'Flutter Demo',
18              theme: ThemeData(
19                  primarySwatch: Colors.blue,
20              ), // ThemeData
21              initialRoute: '/',
22              routes: {
23                  '/': (context) => ArticleListPage(),
24                  '/article_web': (context) =>
25                      ArticleWebView(
26                          ModalRoute.of(context)?.settings.arguments as String
27                      ), // ArticleWebView
28                  '/article_detail': (context) =>
29                      ArticleDetailPage(
30                          ModalRoute.of(context)?.settings.arguments as Article
31                      ), // ArticleDetailPage
32              },
33          ); // MaterialApp
34      }
35  }
```

17. Jalankan aplikasi

### **E. TUGAS**

1. Buatlah sebuah backend server yang memiliki API untuk proses CRUD data tempat wisata. Kemudian ubahlah aplikasi list tempat wisata sehingga penyimpanan data berada pada backend server yang telah dibuat.

### **F. LAPORAN RESMI**

Kumpulkan hasil Percobaan, Latihan dan Tugas di atas dan tambahkan analisa untuk tiap percobaan, latihan, dan tugas yang telah dibuat.