

LAPORAN WORKSHOP PEMROGRAMAN PERANGKAT BERGERAK

WEEK 7 News App

Dosen Pengampu : Prasetyo Wibowo S.ST., M.Kom



Disusun oleh :

Nama : Muhammad Daffa Erfiansyah
NRP : 3123500006
Kelas : 2 D3 IT A

**Program Studi Teknik Informatika
Departemen Teknik
Informatika dan Komputer
Politeknik Elektronika Negeri
Surabaya**

2024/2025

A. Dasar Teori

API (Application Programming Interface) merupakan antarmuka yang memungkinkan aplikasi untuk saling berkomunikasi, terutama dalam pengambilan dan pengiriman data. Salah satu jenis API yang umum digunakan dalam pengembangan aplikasi mobile adalah Public API, yaitu API yang dapat diakses secara terbuka oleh siapa saja, biasanya tanpa memerlukan otentikasi yang kompleks. Public API sering dimanfaatkan untuk mendapatkan data dari layanan pihak ketiga seperti data cuaca, berita, negara, dan lainnya. Data yang diambil dari API umumnya berformat JSON (JavaScript Object Notation) karena ringan, mudah dipahami, dan fleksibel untuk diolah ke dalam struktur data aplikasi mobile.

Fetching Public API adalah mengambil data dari server melalui HTTP request (umumnya menggunakan metode GET). Data yang diterima kemudian di-parse dan ditampilkan ke dalam antarmuka pengguna aplikasi. Proses ini dapat dilakukan menggunakan berbagai tools dan library tergantung platform yang digunakan, seperti Retrofit atau Volley pada Android, serta http package pada Flutter. Praktikum ini bertujuan untuk membekali mahasiswa dengan pemahaman dasar tentang komunikasi antara aplikasi mobile dan server eksternal serta keterampilan dalam mengelola data yang diterima dari API agar dapat digunakan secara efektif di dalam aplikasi.

B. Pratikum

1. Client.dart

a) Source Code :



```
1 import 'dart:convert';
2 import 'package:http/http.dart' as http;
3 import 'package:news_apps/response_articles.dart';
4
5 class Client {
6   static Future<List<Article>> fetchArticles() async {
7     const url =
8       'https://newsapi.org/v2/everything?q=tesla&from=2025-03-07&sortBy=publishedAt&apiKey=a3f9f584c6864ca097abf2d5b7f86162'; //pilih bebas dari newsapi,ga harus sama di pdf!
9     final response = await http.get(Uri.parse(url));
10
11     if (response.statusCode == 200) {
12       // If the server did return a 200 OK response
13       // then parse the JSON
14       Map<String, dynamic> responseBody = jsonDecode(response.body);
15       ResponseArticles responseArticles = ResponseArticles.fromJson(
16         responseBody,
17       );
18       return responseArticles.articles;
19     } else {
20       // If the server did not return a 200 OK response
21       // then throw an exception
22       throw Exception('Failed to load article');
23     }
24   }
25 }
26
```

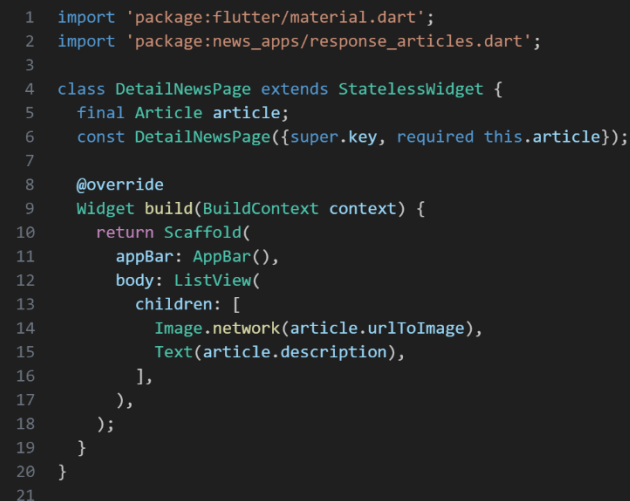
b) Analisa :

Kode di atas merupakan implementasi class Client yang berfungsi untuk mengambil data artikel dari News API secara asynchronous menggunakan

package http. Class ini memiliki dua metode utama: fetchArticles dan searchArticles. Metode fetchArticles mengambil artikel berdasarkan kata kunci, halaman, bahasa, dan waktu terbit, sedangkan searchArticles memungkinkan pencarian lebih lanjut dengan filter tanggal (from dan to). Keduanya membentuk URL menggunakan Uri.parse().replace(), lalu mengirim request ke endpoint https://newsapi.org/v2/everything. Setelah mendapat respon, JSON akan di-decode dan dikonversi menjadi objek ResponseArticles (dari file response_articles.dart). Terdapat pengecekan status HTTP dan penanganan error untuk kode 401, 429, serta log apabila terjadi kegagalan fetching.

2. Detail News Page

a) Source Code :



```
1 import 'package:flutter/material.dart';
2 import 'package:news_apps/response_articles.dart';
3
4 class DetailNewsPage extends StatelessWidget {
5   final Article article;
6   const DetailNewsPage({super.key, required this.article});
7
8   @override
9   Widget build(BuildContext context) {
10    return Scaffold(
11      appBar: AppBar(),
12      body: ListView(
13        children: [
14          Image.network(article.urlToImage),
15          Text(article.description),
16        ],
17      ),
18    );
19  }
20 }
21
```

b) Analisa :

Kode tersebut merupakan implementasi halaman detail berita di Flutter melalui StatelessWidget bernama DetailNewsPage, yang menerima objek Article berisi data berita. Tampilan dibuat dengan CustomScrollView dan SliverAppBar untuk menampilkan gambar berita dengan efek parallax dan animasi transisi Hero. Konten berita ditampilkan melalui SliverList, termasuk judul, penulis, tanggal, deskripsi, dan isi artikel. Tag HTML dibersihkan menggunakan metode _stripHtmlTags, dan format tanggal disesuaikan ke format lokal dengan bantuan package intl.

3. News Page

a) Source Code ;

```

1 import 'package:flutter/material.dart';
2 import 'package:news_apps/client.dart';
3 import 'package:news_apps/response_articles.dart';
4 import 'package:news_apps/detail_news_page.dart';
5
6 class NewsPage extends StatefulWidget {
7   const NewsPage({super.key});
8
9   @override
10  State<NewsPage> createState() => _NewsPageState();
11 }
12
13 class _NewsPageState extends State<NewsPage> {
14   late List<Article> listArticle;
15   bool isLoading = false;
16
17   Future<getlistArticle()> async {
18     setState(() {
19       isLoading = true;
20     });
21
22     listArticle = await Client.fetchArticles();
23     print(listArticle);
24
25     setState(() {
26       isLoading = false;
27     });
28   }
29
30   @override
31   void initState() {
32     super.initState();
33     getlistArticle();
34   }
35
36   @override
37   Widget build(BuildContext context) {
38     return Scaffold(
39       body:
40         isLoading
41           ? const Center(child: CircularProgressIndicator())
42           : listArticle.isEmpty
43             ? const Text('Tidak ada data')
44             : ListView.builder(
45               itemCount: listArticle.length,
46               itemBuilder: (context, index) {
47                 final article = listArticle[index];
48                 return GestureDetector(
49                   onTap: () {
50                     Navigator.push(
51                       context,
52                       MaterialPageRoute(
53                         builder:
54                           (context) => DetailNewsPage(article: article),
55                         ),
56                   );
57               },
58               child: Card(
59                 margin: const EdgeInsets.all(10),
60                 shape: RoundedRectangleBorder(
61                   borderRadius: BorderRadius.circular(20),
62                 ),
63                 elevation: 5,
64                 child: Column(
65                   children: [
66                     ClipRect(
67                       borderRadius: const BorderRadius.only(
68                         topLeft: Radius.circular(20),
69                         topRight: Radius.circular(20),
70                       ),
71                       child: Image.network(
72                         article.urlToImage,
73                         width: double.infinity,
74                         fit: BoxFit.cover,
75                       ),
76                     ),
77                     Padding(
78                       padding: const EdgeInsets.all(15.0),
79                       child: Text(
80                         article.title,
81                         style: const TextStyle(
82                           fontSize: 20,
83                           fontWeight: FontWeight.bold,
84                         ),
85                       ),
86                     ),
87                     Padding(
88                       padding: const EdgeInsets.all(15),
89                       child: Row(
90                         mainAxisAlignment: MainAxisAlignment.spaceBetween,
91                         children: [
92                           Expanded(
93                             child: Text(
94                               article.author,
95                               style: const TextStyle(
96                                 fontSize: 15,
97                                 fontWeight: FontWeight.bold,
98                               ),
99                             overflow: TextOverflow.ellipsis,
100                           ),
101                           Text(
102                             article.publishedAt.toString(),
103                             style: TextStyle(
104                               fontSize: 15,
105                               color: Colors.grey[600],
106                             ),
107                           ),
108                         ],
109                       ),
110                     ),
111                   ],
112                 ),
113               ),
114             );
115
116   },
117 );
118
119 }
120
121

```

b) Analisa :

Kode NewsPage merupakan contoh manajemen state dan interaksi pengguna dalam Flutter menggunakan StatefulWidget. Halaman ini mendukung fitur pencarian, pagination, dan refresh data melalui metode getListArticle(), dengan memanfaatkan variabel state seperti currentPage, pageCache, dan searchQuery. Antarmuka dibangun secara modular lewat metode seperti _buildSearchBar() dan _buildArticleCard(), yang menangani berbagai kondisi aplikasi, termasuk loading, error, dan data kosong. Komponen visual dirancang responsif dan menarik, menggunakan elemen seperti ClipRRect, bayangan, serta RefreshIndicator, dan menangani kesalahan gambar dengan baik. Format tanggal ditata menggunakan DateFormat, menjaga tampilan tetap konsisten dan informatif.

4. Response Articles

a) Source Code :

```
1  import 'dart:convert';
2
3  ResponseArticles responseArticlesFromJson(String str) =>
4    ResponseArticles.fromJson(json.decode(str));
5
6  String responseArticlesToJson(ResponseArticles data) =>
7    json.encode(data.toJson());
8
9  class ResponseArticles {
10    final String status;
11    final int totalResults;
12    final List<Article> articles;
13
14    ResponseArticles({
15      required this.status,
16      required this.totalResults,
17      required this.articles,
18    });
19
20    factory ResponseArticles.fromJson(Map<String, dynamic> json) =>
21      ResponseArticles(
22        status: json['status'],
23        totalResults: json['totalResults'],
24        articles: List<Article>.from(
25          json['articles'].map((x) => Article.fromJson(x)),
26        ),
27      );
28
29    Map<String, dynamic> toJson() => {
30      'status': status,
31      'totalResults': totalResults,
32      'articles': List<dynamic>.from(articles.map((x) => x.toJson())),
33    };
34  }
35
36  class Article {
37    final Source source;
38    final String author;
39    final String title;
40    final String description;
41    final String url;
42    final String urlToImage;
43    final DateTime publishedAt;
44    final String content;
```

```

1 Article({
2   required this.source,
3   required this.author,
4   required this.title,
5   required this.description,
6   required this.url,
7   required this.urlToImage,
8   required this.publishedAt,
9   required this.content,
10 });
11
12 factory Article.fromJson(Map<String, dynamic> json) => Article(
13   source: Source.fromJson(json['source']),
14   author: json['author'] ?? 'tidak ada author',
15   title: json['title'] ?? 'tidak ada title',
16   description: json['description'] ?? 'tidak ada description',
17   url: json['url'] ?? 'tidak ada url',
18   urlToImage:
19     json['urlToImage'] ??
20     'https://cdn.vectorstock.com/i/preview-1x/65/30/default-image-icon-missing-picture-page-vector-48546530.jpg',
21   publishedAt: DateTime.parse(json['publishedAt'] ?? '2023-11-20T02:19:48Z'),
22   content: json['content'] ?? 'tidak ada konten',
23 );
24
25 Map<String, dynamic> toJson() => {
26   'source': source.toJson(),
27   'author': author,
28   'title': title,
29   'description': description,
30   'url': url,
31   'urlToImage': urlToImage,
32   'publishedAt': publishedAt.toIso8601String(),
33   'content': content,
34 };
35 }
36
37 class Source {
38   final String? id;
39   final String name;
40
41   Source({required this.id, required this.name});
42
43   factory Source.fromJson(Map<String, dynamic> json) =>
44     Source(id: json['id'], name: json['name']);
45
46   Map<String, dynamic> toJson() => {'id': id, 'name': name};
47 }

```

b) Analisa :

Kode tersebut merupakan bagian dari model data yang digunakan untuk mengonversi respons JSON dari News API menjadi objek Dart. Terdapat tiga kelas utama: `ResponseArticles`, `Article`, dan `Source`. Kelas `ResponseArticles` menjadi induk data yang memuat status respons, total hasil (`totalResults`), serta daftar artikel dalam bentuk list `Article`. Setiap objek `Article` merepresentasikan satu berita dengan atribut seperti `author`, `title`, `description`, `url`, dan `publishedAt`. Sementara itu, kelas `Source` merepresentasikan sumber berita, termasuk ID dan nama medianya.

Setiap kelas dilengkapi dengan metode `fromJson` untuk parsing dari JSON ke objek, serta `toJson` untuk konversi sebaliknya. Pada `Article.fromJson`, nilai default diberikan untuk menghindari error jika ada field yang kosong—sebuah pendekatan defensif yang baik untuk menangani data API yang tidak selalu konsisten. Fungsi tambahan `responseArticlesFromJson` dan `responseArticlesToJson` mempermudah konversi langsung antara string JSON dan objek Dart.

5. Output :



China Halts Export Of Rare Minerals, Magnet As Trade War With US Intensifies

Sanstuti Nath 2025-04-14 03:13:38.000Z



As the trade war between the US and China escalates, Beijing has suspended the export of several critical rare earth elements, metals and magnets, threatening to choke off supplies to the West



A Tesla Patent Might Be The Answer To China's Blanket Restrictions On The Export Of Rare Earth Metals

Rohail Saleem 2025-04-14 03:03:46.000Z

C. Kesimpulan

Praktikum tentang fetching public API menekankan pentingnya pengambilan data dari sumber eksternal melalui HTTP dalam pengembangan aplikasi modern. Mahasiswa mempelajari cara memahami format JSON, melakukan parsing data, dan menampilkannya di antarmuka aplikasi. Selain itu, penanganan error menjadi hal penting untuk menjaga kestabilan aplikasi saat terjadi kegagalan koneksi atau data yang tidak valid. Penggunaan package seperti http, intl, dan url_launcher turut menunjang fungsionalitas, seperti membuka tautan eksternal dan memformat tanggal. Praktikum ini memperdalam pemahaman terkait integrasi API dalam membangun aplikasi yang responsif dan informatif.

