

**LAPORAN TEORI MOBILE PROGRAMMING**  
**MODUL 10**



Nama : Erai Bagusalm  
NIM : 240605110088  
Kelas : Mobile Programming B  
Tanggal : 6 September 2025

**JURUSAN TEKNIK INFORMATIKA**  
**FAKULTAS SAINS DAN TEKNOLOGI**  
**UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM**  
**MALANG**  
**GANJIL 2025/2026**

i. **Tujuan**

1. Mendesain antarmuka (GUI) aplikasi Flutter yang interaktif dan responsif.
2. Menerapkan sistem navigasi antarhalaman menggunakan mekanisme named route.
3. Membuat model data dengan JSON serialization dan deserialization.
4. Mengintegrasikan data dari API eksternal ke dalam aplikasi menggunakan paket http.
5. Menampilkan data hasil permintaan API ke layar aplikasi dengan tata letak yang rapi dan dinamis.

ii. **Langkah Kerja**

1. Membuat proyek Flutter baru bernama named\_route\_app.
2. Membuat beberapa halaman (screens), misalnya HomePage, dan DetailPage.
3. Menambahkan konfigurasi named route pada file main.dart menggunakan MaterialApp dengan properti routes.
4. Mengimplementasikan navigasi antar halaman menggunakan Navigator.pushNamed() dan Navigator.pop().
5. Menambahkan tombol navigasi di setiap halaman untuk berpindah ke halaman lain.
6. Menjalankan aplikasi untuk memastikan navigasi berjalan sesuai rute yang telah ditentukan.

### iii. Screenshot Hasil

#### a. Kode Program

```
Expanded(
  child: Container(
    padding: const EdgeInsets.symmetric(
      horizontal: 10,
      vertical: 10,
    ), // EdgeInsets.symmetric
    child: Container(
      padding: const EdgeInsets.all(10),
      width: double.infinity,
      decoration: BoxDecoration(
        color: Colors.blue.shade50,
        borderRadius: BorderRadius.circular(10),
      ), // BoxDecoration
      child: SingleChildScrollView(
        child: Column(
          children: [
            SizedBox(
              width: double.maxFinite,
              child: Text(
                'Minimum System Requirements',
                style: TextStyle(
                  fontSize: 14,
                  fontWeight: FontWeight.bold,
                  color: Colors.black54.withOpacity(0.8), 'with
            ), // TextStyle
          ), // Text
        ), // SizedBox
      ), // Column
    ), // Column
  ), // Column
```

```
1 import 'package:flutter/material.dart';
2 import 'package:test10/viewmodel/fetchgame.dart';
3 import 'package:test10/model/detailgame.dart';
4 import 'package:readmore/readmore.dart';
5
6 class Detail extends StatelessWidget {
7   final int gameId;
8   const Detail({super.key, required this.gameId});
9
10  Future<DetailGame> fetchData() async {
11    final jsonData = await fetchDataFromAPI(gameId);
12    return DetailGame.fromJson(jsonData);
13  }
14
15  @override
16  Widget build(BuildContext context) {
17    return Scaffold(
18      backgroundColor: Colors.amberAccent.shade400,
19      body: FutureBuilder(
20        future: fetchData(),
21        builder: (context, snapshot) {
22          if (snapshot.connectionState == ConnectionState.waiting) {
23            return const Center(child: CircularProgressIndicator());
24          } else if (snapshot.hasError) {
25            return Center(child: Text('Terjadi kesalahan: ${snapshot.error}'));
26          } else if (!snapshot.hasData) {
27            return const Center(child: Text('Data tidak tersedia'));
28          } else {
29            final game = snapshot.data!;
30            return Column(
31              children: [
32                SizedBox(
33                  width: double.infinity,
34                  child: Stack(
35                    children: [
36                      ClipRRect(
37                        borderRadius: const BorderRadius.only(
38                          bottomLeft: Radius.circular(25),
39                          bottomRight: Radius.circular(25),
40                        ), // BorderRadius.only
41                        child: Image.network(
42                          game.thumbnail,
43                          fit: BoxFit.cover,
44                          height: 300,
45                        ), // Image.network
46                      ), // ClipRRect
47                      Positioned(
48                        top: 25,
49                        left: 10,
50                        child: GestureDetector(
51                          onTap: () {
52                            Navigator.pop(context);
53                          },
54                          child: Container(
55                            padding: const EdgeInsets.all(5),
56                            decoration: BoxDecoration(
57                              color: Colors.grey.withOpacity(0.8), 'with
58                            shape: BoxShape.circle,
59                            ), // BoxDecoration
60                            child: const Icon(
61                              Icons.arrow_back_ios_outlined,
62                              color: Colors.white,
63                            ), // Icon
64                          ), // Container
65                        ), // GestureDetector
66                      ), // Positioned
67                    ], // Stack
68                  ), // Stack
69                ), // SizedBox
```

```

1 import 'package:flutter/material.dart';
2 import 'package:flutter/services.dart';
3 import 'package:test10/view/home.dart';
4 import 'package:test10/view/detail.dart';
5
6 Run | Debug | Profile
7 void main() {
8   runApp(const MyApp());
9 }
10
11 class MyApp extends StatelessWidget {
12   const MyApp({super.key});
13
14   @override
15   Widget build(BuildContext context) {
16     SystemChrome.setSystemUIOverlayStyle(
17       const SystemUIOverlayStyle(
18         statusBarColor: Colors.transparent,
19         statusBarIconBrightness: Brightness.dark,
20       ),
21     );
22
23     return MaterialApp(
24       debugShowCheckedModeBanner: false,
25       title: 'Flutter Game Store',
26       initialRoute: '/',
27       routes: {
28         '/': (context) => const Home(),
29         '/detail': (context) {
30           final gameId = ModalRoute.of(context)?.settings.arguments as int;
31           return Detail(gameTerpilih: gameId);
32         },
33       },
34     ); // MaterialApp
35 }

```

```

1 import 'package:json_annotation/json_annotation.dart';
2 part 'detailgame.g.dart';
3
4 @JsonSerializable()
5 class MinimumSystemRequirements {
6   final String os;
7   final String processor;
8   final String memory;
9   final String graphics;
10  final String storage;
11
12  MinimumSystemRequirements({
13    required this.os,
14    required this.processor,
15    required this.memory,
16    required this.graphics,
17    required this.storage,
18  });
19
20  factory MinimumSystemRequirements.fromJson(Map<String, dynamic> json) =>
21    _MinimumSystemRequirementsFromJson(json);
22  Map<String, dynamic> toJson() => _MinimumSystemRequirementsToJson(this);
23 }
24
25 @JsonSerializable()
26 class Screenshot {
27   final int id;
28   final String image;
29
30   Screenshot({required this.id, required this.image});
31
32   factory Screenshot.fromJson(Map<String, dynamic> json) =>
33     _SScreenshotFromJson(json);
34   Map<String, dynamic> toJson() => _SScreenshotToJson(this);
35 }
36

```

```

37 @JsonSerializable()
38 class DetailGame {
39   final int id;
40   final String title;
41   final String thumbnail;
42   final String status;
43   final String short_description; The variable name 'short_desc'
44   final String description;
45   final String game_url; The variable name 'game_url' isn't a
46   final String genre;
47   final String platform;
48   final String publisher;
49   final String developer;
50   final String release_date; The variable name 'release_date'
51   final String freetogame_profile_url; The variable name 'freeto
52   final MinimumSystemRequirements minimum_system_requirements;
53   final List<Screenshot> screenshots;
54
55   DetailGame({
56     required this.id,
57     required this.title,
58     required this.thumbnail,
59     required this.status,
60     required this.short_description, The variable name 'short_c
61     required this.description,
62     required this.game_url, The variable name 'game_url' isn't
63     required this.genre,
64     required this.platform,
65     required this.publisher,
66     required this.developer,
67     required this.release_date, The variable name 'release_date
68     required this.freetogame_profile_url, The variable name 'freeto
69     required this.minimum_system_requirements, The variable name
70     required this.screenshots,
71   });
72
73   factory DetailGame.fromJson(Map<String, dynamic> json) =>
74     _$DetailGameFromJson(json);
75   Map<String, dynamic> toJson() => _$DetailGameToJson(this);
76 }
77
78

```

```

1 import 'package:flutter/material.dart';
2 import 'package:test10/model/game.dart';
3 import 'package:test10/viewmodel/fetchgame.dart';
4
5 class Home extends StatefulWidget {
6   const Home({super.key});
7
8   @override
9   State<Home> createState() => _HomeState();
10 }
11
12 class _HomeState extends State<Home> {
13   late Future<List<Game>> gameList;
14
15   @override
16   void initState() {
17     super.initState();
18     loadGames();
19   }
20
21   Future<void> loadGames() async {
22     gameList = fetchGames();
23   }
24
25   @override
26   Widget build(BuildContext context) {
27     return Scaffold(
28       backgroundColor: Colors.amberAccent.shade400,
29       body: SafeArea(
30         child: Container(
31           padding: const EdgeInsets.symmetric(horizontal: 10, vertical: 5),
32           child: Column(
33             children: [
34               SearchBar(),
35               const SizedBox(height: 10),
36               FutureBuilder<List<Game>>{
37                 future: gameList,
38                 builder: (context, snapshot) {
39                   if (snapshot.connectionState == ConnectionState.waiting) {
40                     return const Expanded(
41                       child: Center(child: CircularProgressIndicator()),
42                     ); // Expanded
43                   } else if (snapshot.hasError) {
44                     return Text("Error: ${snapshot.error}");
45                   } else if (!snapshot.hasData || snapshot.data!.isEmpty) {
46                     return const Text("Tidak ada data game.");
47                   } else {
48                     final games = snapshot.data!.take(25).toList();
49                     return Expanded(
50                       child: ListView.builder(
51                         itemCount: games.length,
52                         itemBuilder: (context, index) {
53                           final game = games[index];
54                           return GestureDetector(
55                             onTap: () {
56                               Navigator.pushNamed(
57                                 context,
58                                 '/detail',
59                                 arguments: game.id,
60                               );
61                             },
62                           child: ListTile(
63                             game.thumbnail,
64                             game.title,
65                             game.genre,
66                           ),
67                         ),
68                       ),
69                     );
70                   }
71                 },
72               ),
73             ],
74           ),
75         ),
76       ),
77     );
78

```

```

82 Card _listItem(String urlCover, String judul, String genre) {
83   return Card(
84     clipBehavior: Clip.antiAlias,
85     elevation: 3,
86     shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(8)),
87     child: ListTile(
88       tileColor: Colors.blue.shade50,
89       leading: SizedBox(
90         width: 75,
91         height: 75,
92         child: ClipRRect(
93           borderRadius: BorderRadius.circular(8),
94           child: Image.network(urlCover, scale: 3, fit: BoxFit.cover),
95         ), // ClipRRect
96       ), // SizedBox
97       title: Text(judul),
98       subtitle: Text(genre),
99       trailing: const Icon(Icons.more_vert),
100      isThreeLine: false,
101      titleAlignment: ListTileTitleAlignment.center,
102    ), // ListTile
103  ); // Card
104 }
105
106 TextField _searchBar() {
107   return TextField(
108     cursorColor: Colors.blue,
109     decoration: InputDecoration(
110       fillColor: Colors.blue.shade50,
111       filled: true,
112       border: OutlineInputBorder(
113         borderRadius: BorderRadius.circular(20),
114         borderSide: const BorderSide(width: 0, style: BorderStyle.none),
115       ), // OutlineInputBorder
116       contentPadding: const EdgeInsets.symmetric(vertical: 0, horizontal: 20),
117       prefixIcon: const Icon(
118         Icons.search_outlined,
119         color: Colors.blue,
120         size: 30,
121       ), // Icon
122       hintText: 'Cari game',
123       hintStyle: TextStyle(fontSize: 14, color: Colors.grey.withOpacity(0.7)),
124     ), // InputDecoration
125   ); // TextField
126 }
127

```

```

1  import 'dart:convert';
2  import 'package:test10/model/game.dart';
3  import 'package:http/http.dart' as http;
4
5  Future<List<Game>> fetchGames() async {
6    final response = await http.get(
7      Uri.parse('https://www.freetogame.com/api/games'),
8    );
9
10   if (response.statusCode == 200) {
11     final List<dynamic> jsonData = json.decode(response.body);
12     return jsonData.map((item) => Game.fromJson(item)).toList();
13   } else {
14     throw Exception('Gagal mengambil data game');
15   }
16 }
17
18 Future<Map<String, dynamic>> fetchDataFromAPI(int idGame) async {
19   final response = await http.get(
20     Uri.parse('https://www.freetogame.com/api/game?id=$idGame'),
21   );
22   if (response.statusCode == 200) {
23     final jsonData = json.decode(response.body);
24     if (jsonData != null && jsonData is Map<String, dynamic>) {
25       return jsonData;
26     } else {
27       throw Exception('Data dari API tidak sesuai dengan yang diharapkan');
28     }
29   } else {
30     throw Exception('Gagal mengambil data dari API');
31   }
32 }
33

```

```

1  class Game {
2    final int id;
3    final String title;
4    final String thumbnail;
5    final String shortDescription;
6    final String gameUrl;
7    final String genre;
8    final String platform;
9    final String publisher;
10   final String developer;
11   final String releaseDate;
12   final String freetogameProfileUrl;
13
14   Game({
15     required this.id,
16     required this.title,
17     required this.thumbnail,
18     required this.shortDescription,
19     required this.gameUrl,
20     required this.genre,
21     required this.platform,
22     required this.publisher,
23     required this.developer,
24     required this.releaseDate,
25     required this.freetogameProfileUrl,
26   });
27
28   factory Game.fromJson(Map<String, dynamic> json) {
29     return Game(
30       id: json['id'],
31       title: json['title'],
32       thumbnail: json['thumbnail'],
33       shortDescription: json['short_description'],
34       gameUrl: json['game_url'],
35       genre: json['genre'],
36       platform: json['platform'],
37       publisher: json['publisher'],
38       developer: json['developer'],
39       releaseDate: json['release_date'],
40       freetogameProfileUrl: json['freetogame_profile_url'],
41     );
42   }
43 }
44

```

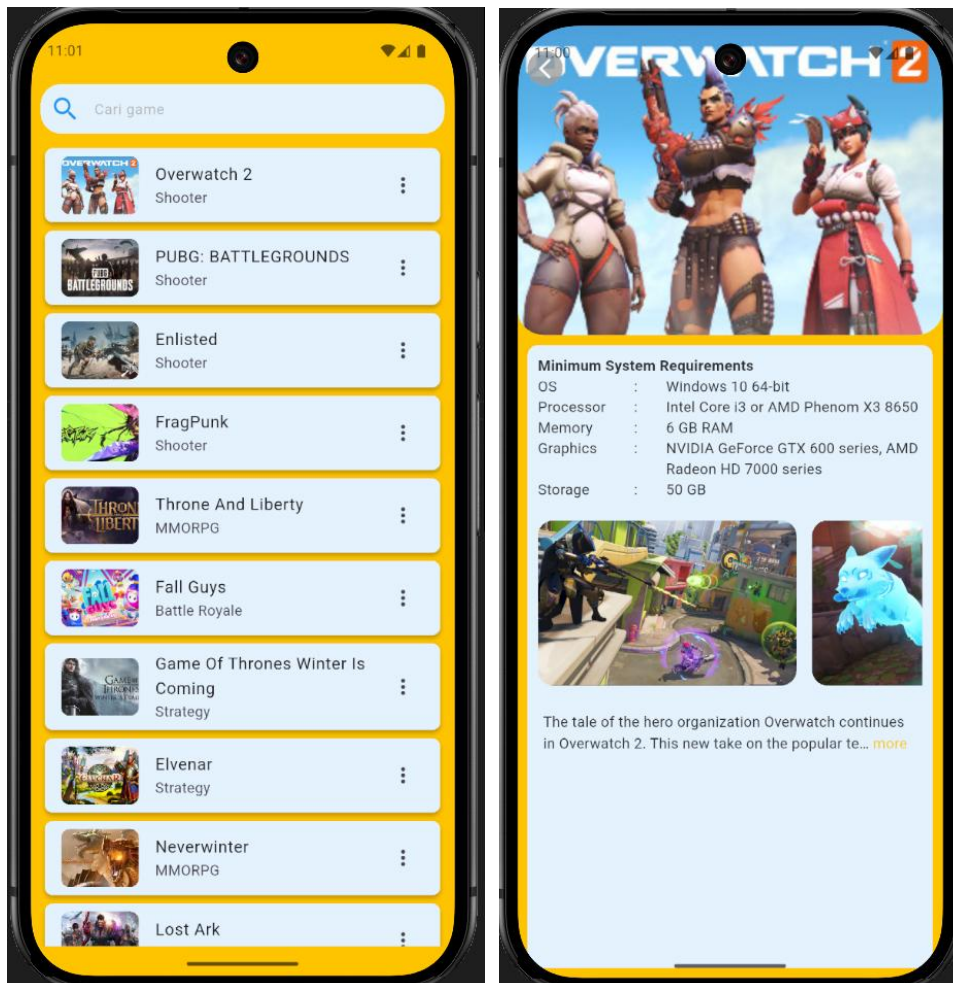
## **b. Penjelasan Kode Program**

Aplikasi galeri game ini dirancang dengan arsitektur yang jelas, memisahkan antara tampilan, data, dan logika. Jantung aplikasi berada di `main.dart`, di mana sistem navigasi `named route` diatur. Halaman utama (`/`) akan menampilkan `Home`, sementara rute `/detail` disiapkan untuk halaman detail game, yang dengan cerdas mengambil ID game yang dikirimkan sebagai argument.

Saat aplikasi dibuka, halaman `Home` langsung memanggil fungsi `fetchGames()` untuk mengambil daftar game dari API `FreeToGame`. Proses ini ditangani dengan elegan oleh `FutureBuilder`, yang akan menampilkan animasi memuat saat data sedang diunduh, sehingga pengguna tahu aplikasi sedang bekerja. Setelah data diterima, `ListView.builder` secara efisien menampilkan daftar game yang diterima, hanya merender item yang terlihat di layar. Setiap item dalam daftar ini dapat ditekan, yang kemudian memicu `Navigator.pushNamed()` untuk berpindah ke halaman detail sambil membawa ID game yang dipilih.

Di halaman detail, `FutureBuilder` kembali digunakan, kali ini untuk memanggil `fetchDataFromAPI()` dengan ID spesifik guna mendapatkan informasi lengkap tentang satu game. Untuk mengubah data mentah JSON dari API menjadi objek Dart yang terstruktur, aplikasi ini mengandalkan kelas-kelas model seperti `Game` dan `DetailGame`. Proses konversi ini diotomatisasi menggunakan `json_serializable`, yang secara signifikan menyederhanakan proses parsing data dan membuat kode lebih bersih serta aman dari kesalahan.

Output :



#### iv. Kesimpulan

Secara keseluruhan, proyek ini berhasil mencapai tujuannya dengan mengintegrasikan beberapa konsep fundamental dalam pengembangan aplikasi Flutter. Penggunaan **named route** terbukti sangat efektif untuk menciptakan alur navigasi yang terstruktur dan mudah dikelola, terutama saat perlu mengirim data antar halaman. Implementasi **FutureBuilder** menjadi kunci dalam menciptakan pengalaman pengguna yang responsif, karena mampu menangani proses pengambilan data dari internet tanpa membuat aplikasi berhenti atau *freeze*, sambil memberikan umpan balik visual yang jelas kepada pengguna.

Lebih dari itu, pendekatan yang digunakan untuk mengelola data—dengan membuat kelas model dan memanfaatkan **json\_serializable**—menunjukkan cara yang kuat dan efisien untuk bekerja dengan API eksternal. Kombinasi dari ketiga elemen ini—navigasi yang rapi, penanganan data asinkron yang mulus, dan pemodelan data yang kokoh—membentuk fondasi yang solid untuk membangun aplikasi modern yang digerakkan oleh data (*data-driven*) dan siap untuk dikembangkan lebih lanjut.