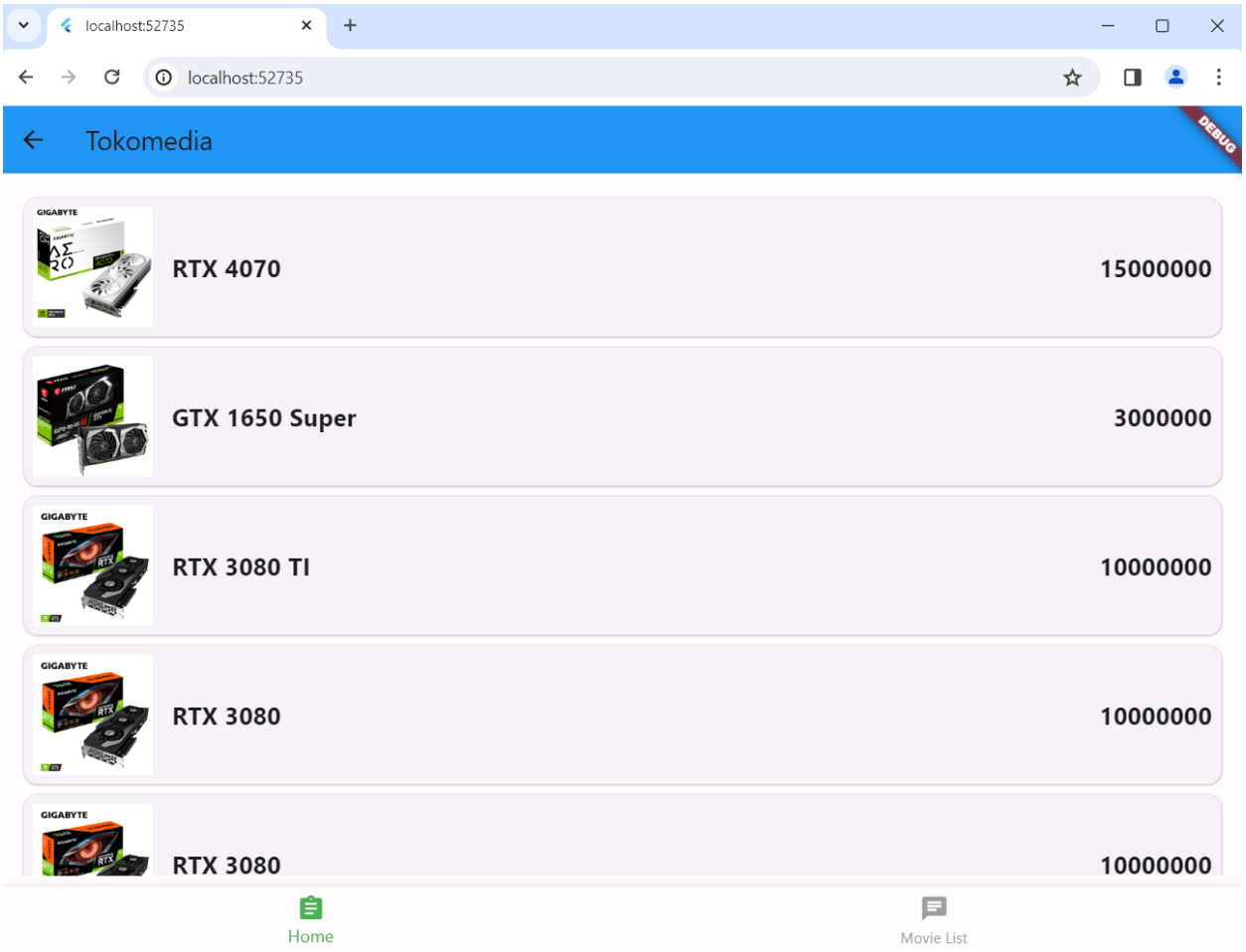
Nama : Muhammad Haikal

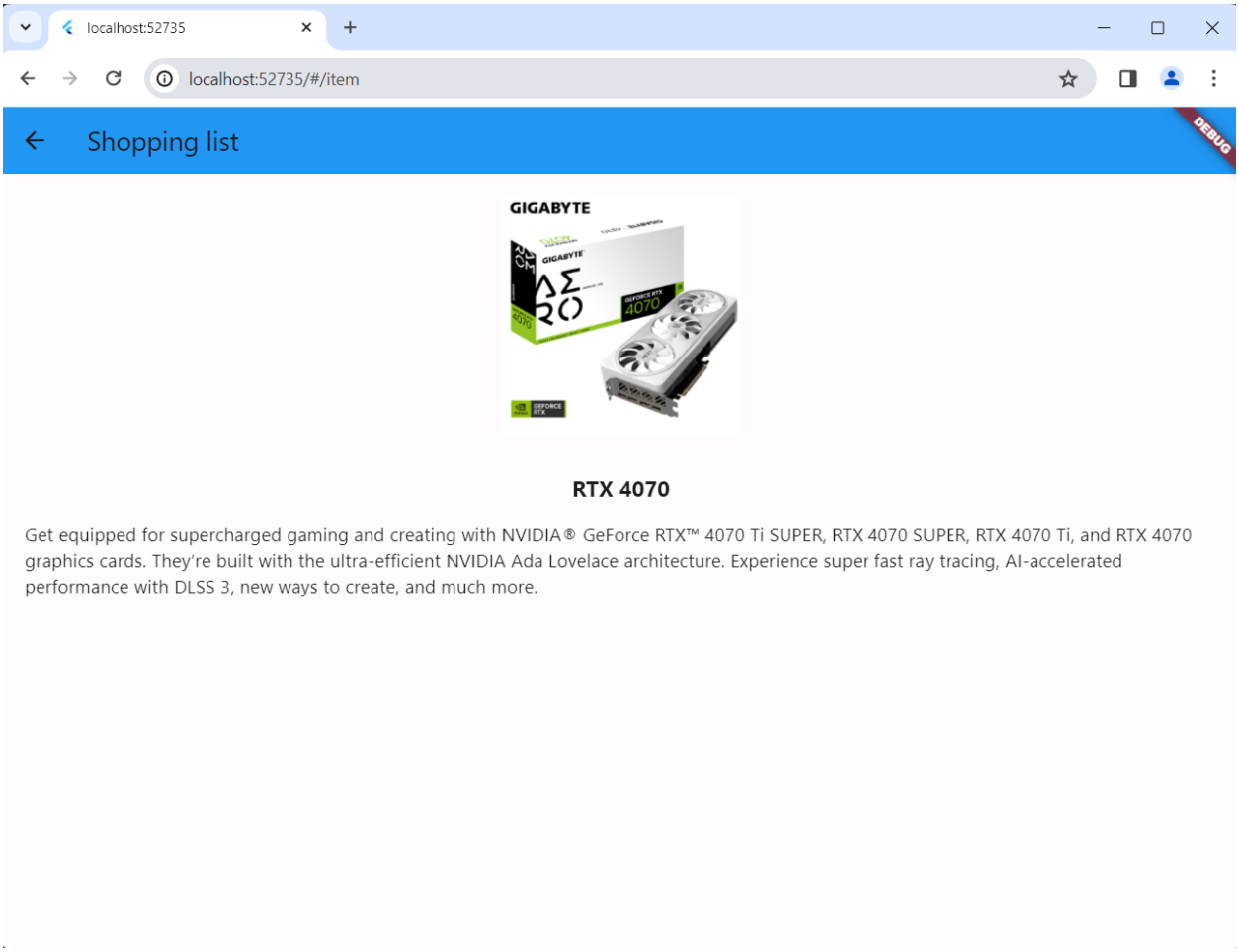
Kelas/Absen : XI RPL4 / 23

Screenshoot

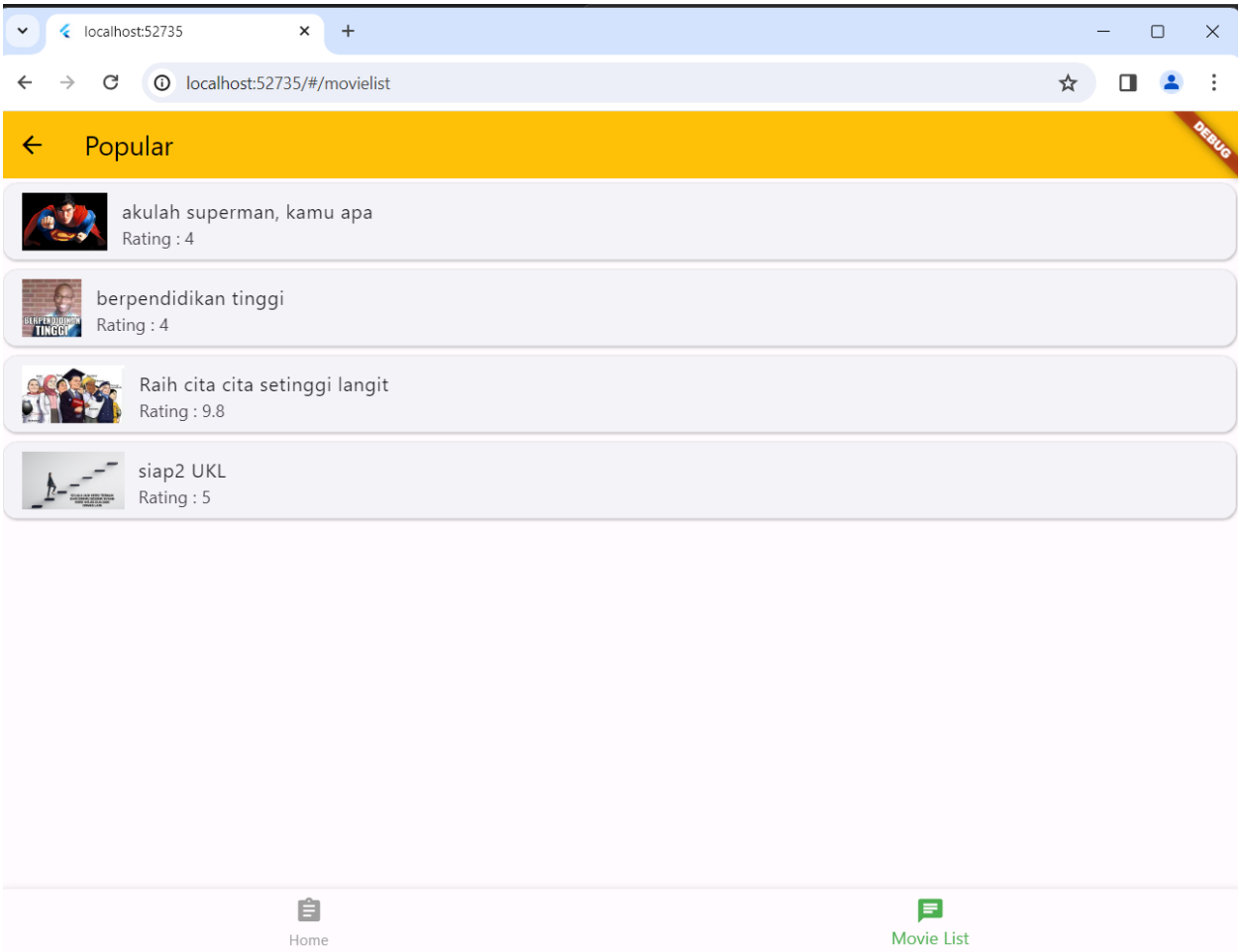
Layar 1 :



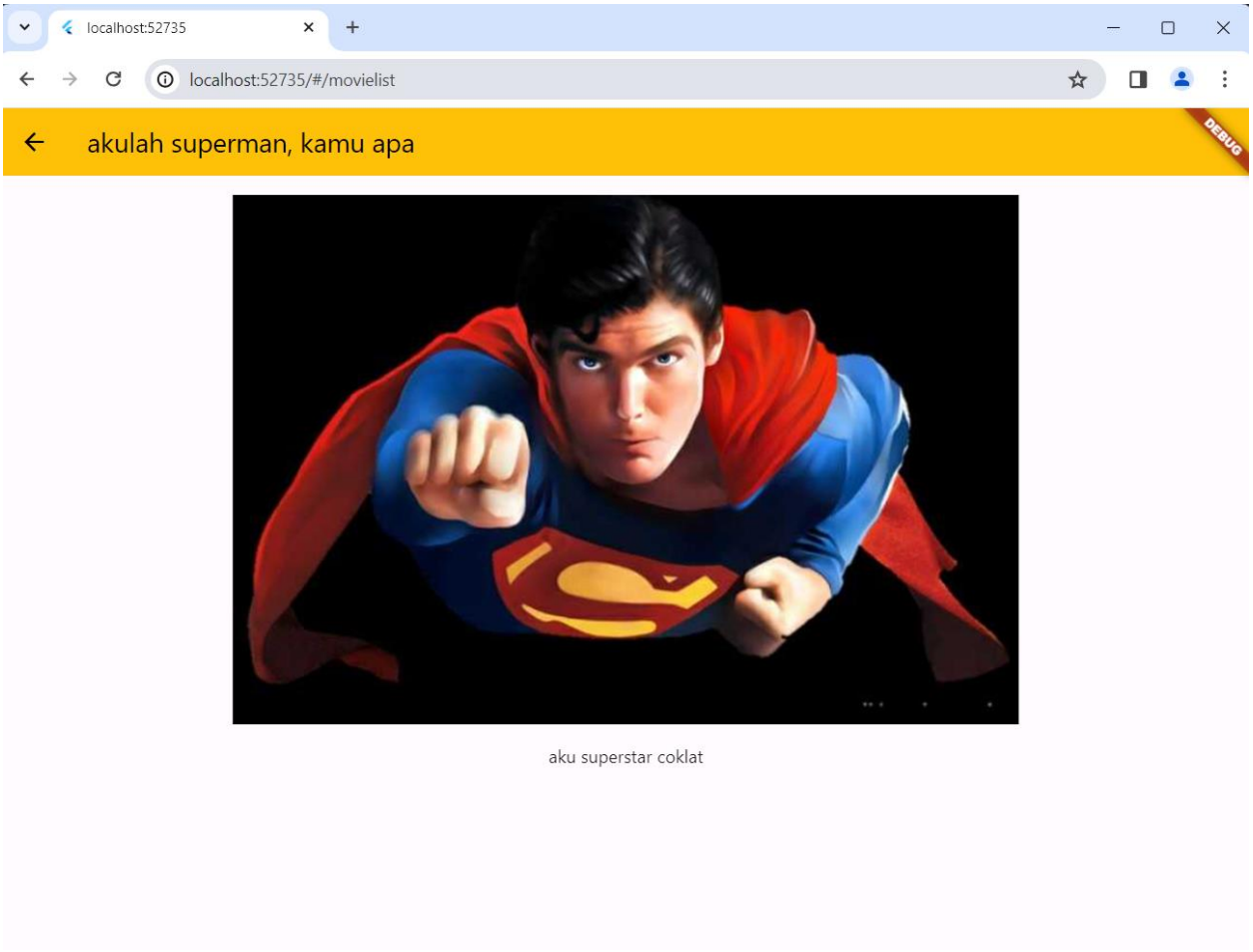
Layar 1 Bagian 2 :



Layar 2 bagian 1 :



Layar 2 bagian 2 :



Penjelasan Script :

```dart
class HttpService {
  final String baseUrl = 'https://movie.tukanginyuk.com/api/getmovie';

  Future<List?> getPopularMovies() async {
    final String uri = baseUrl;

    http.Response result = await http.get(Uri.parse(uri));

    if(result.statusCode == HttpStatus.ok){
      print("Connected");
      final jsonResponse = json.decode(result.body);
      final moviesMap = jsonResponse['data'];
      List movie = moviesMap.map((i) => Film.fromJson(i)).toList();
      return movie;
    }else{
      print("Not Connected");
      return null;
    }

  }
}
```

Penjelasan :

Fungsi ini menunjukkan bagaimana pemrograman asinkron dan pemanggilan API ditangani dalam aplikasi Dart/Flutter. Ini juga menunjukkan penggunaan praktis dari future, penguraian JSON, dan penanganan kesalahan dalam pengembangan aplikasi.

```dart
class Film {
  int? id;
  String? title;
  double? voteAverage;
  String? overview;
  String? posterPath;

  Film(this.id, this.title, this.voteAverage, this.overview, this.posterPath);

  Film.fromJson(Map<String, dynamic> parsedJson){
    id = parsedJson['id'];
    title = parsedJson['title'];
    voteAverage = parsedJson['voteaverage']*1.0;
    overview = parsedJson['overview'];
    posterPath = parsedJson['posterpath'];
  }
}
```

Penjelasan :

Kode ini menunjukkan bagaimana PBO digunakan untuk memodelkan film dalam aplikasi. Ini juga menunjukkan bagaimana data JSON dapat di-deserialisasi menjadi objek dalam aplikasi, yang umum ketika bekerja dengan API yang menyediakan data dalam format JSON.

```dart
class item {
  String name;
  int price;
  String ImagePath;
  String description;
  item({required this.name, required this.price, required this.ImagePath, required th
}
```

Penjelasan :
Kode diatas bertujuan untuk menyimpan nama,harga,gambar,deskripsi pada kelas homepage, kode diatas merupakan bagian dari database lokal yang digunakan dalam homepage

Penjelasan :
kode diatas membuat daftar list, di mana setiap list menampilkan gambar dan teks. Ketika list diklik, aplikasi akan berpindah halaman dan mengirimkan data item. Di bagian bawah, ada navbar yang memungkinkan pengguna untuk beralih antara homepage dan moviepage.

```dart
class itempage extends StatelessWidget {
  itempage({super.key});

  @override
  Widget build(BuildContext context) {
    final itemArgs = (ModalRoute.of(context)!.settings.arguments) as Map;
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.blue,
        title: const Text('Shopping list'),
      ), // AppBar
      body: Container(
        padding: EdgeInsets.all(20),
        child: Column(
          children: [
                Container(
                  width: 200, // Set the desired width for the image
                  height: 200,
                  child: Image.network(itemArgs["ImagePath"]),
                ), // Container
                SizedBox(
                    height: 30), // Add spacing between image and description // SizedBox
                Container(
                  child: Text(
                    itemArgs["name"],
                    style: TextStyle(
                      fontSize: 18,
                      fontWeight: FontWeight.bold // Set the font size for the description
                    ), // TextStyle
                  ), // Text
                ), // Container
            SizedBox(height: 16), // Add spacing between image and name
            Text(
              itemArgs["description"],
              style: TextStyle(
                fontSize: 15, // Set the font size to make the name bigger
              ), // TextStyle
            ), // Text
          ],
        ), // Column
      ), // Container
    ); // Scaffold
  }
```

Penjelasan :
Secara keseluruhan, kode ini membuat daftar kartu, di mana setiap kartu menampilkan gambar dan beberapa teks. Ketika kartu diklik, aplikasi akan melakukan navigasi ke halaman lain dan mengirimkan data item sebagai argumen. Di bagian bawah, ada navigasi bar yang memungkinkan pengguna untuk beralih antara berbagai bagian dari aplikasi.

```dart
@override
void initState(){
  service = HttpService();
  initialize();
  super.initState();
}

@override
Widget build(BuildContext context) {
  service!.getPopularMovies().then((value) => {
    setState(() {
      result =(value != null) as String;
    })
  });
  return Scaffold(
    appBar: AppBar(
      title: const Text("Popular"),
      backgroundColor: Colors.amber,
      foregroundColor: Colors.black,
    ), // AppBar
    body: film!.isEmpty ? Container() :
    ListView.builder(
      itemCount: moviesCount,
      itemBuilder: (context, int position){
        return Card(
          color: Colors.white,
          elevation: 2.0,
          child: ListTile(
            leading: Image(image: NetworkImage(film![position].posterPath)),
            title: Text(film![position].title),
            subtitle: Text("Rating : ${film![position].voteAverage.toString()}"),
            onTap: (){
              MaterialPageRoute route = MaterialPageRoute(
                builder: (_) => MovieDetail(film?[position])); // MaterialPageRoute
                Navigator.push(context, route);
            },
          ), // ListTile
        ); // Card
      }
    ), // ListView.builder
    bottomNavigationBar: BottomNav(selectedItem: 1)
  ); // Scaffold
  }
}
```

Penjelasan :

Kode ini menunjukkan penanganan data asinkron, pembuatan UI dengan widget, dan interaksi dengan layanan eksternal untuk pengambilan data.

```dart
class BottomNav extends StatefulWidget {
  final int selectedItem;
  BottomNav({super.key, required this.selectedItem});

  @override
  State<BottomNav> createState() => _BottomNavState();
}

class _BottomNavState extends State<BottomNav> {
  int selectedNavbar = 0;
  void changeSelectedNavBar(int index) {
    setState(() {
      selectedNavbar = index;
    });
    if (index == 0) {
      Navigator.pushNamed(context, '/');
    } else if (index == 1) {
      Navigator.pushNamed(context, '/movielist');
    }
  }

  @override
  Widget build(BuildContext context) {
    return BottomNavigationBar(
      items: const <BottomNavigationBarItem>[
        BottomNavigationBarItem(
          icon: Icon(Icons.assignment),
          label: 'Home',
        ), // BottomNavigationBarItem
        BottomNavigationBarItem(
          icon: Icon(Icons.chat),
          label: 'Movie List',
        ), // BottomNavigationBarItem
      ], // <BottomNavigationBarItem>[]
      selectedItemColor: Colors.green,
      unselectedItemColor: Colors.grey,
      showUnselectedLabels: true,
      currentIndex: widget.selectedItem,
      onTap: changeSelectedNavBar,
    ); // BottomNavigationBar
  }
}
```

Penjelasan :

kode ini diatas memungkinkan pengguna untuk berpindah antara berbagai layar dalam aplikasi dengan mengklik item navbarlayar.

```dart
class MovieDetail extends StatelessWidget {
  final Film film;
  const MovieDetail(this.film);

  @override
  Widget build(BuildContext context) {
    String path;
    if (film.posterPath != null){
      path = film.posterPath!;
    } else {
      path = 'https://cdn.vectorstock.com/i/preview-1x/82/99/no-image-available-like-missing-picture-vector-43938299.jpg';
    }
    double height = MediaQuery.of(context).size.height;
    return Scaffold(
      appBar: AppBar(
        title: Text(film.title!),
        backgroundColor: Colors.amber,
        foregroundColor: Colors.black,
      ), // AppBar
      body: SingleChildScrollView(
        child: Center(
          child: Column(
            children: [
              Container(
                padding: EdgeInsets.all(16),
                height: height / 1.5,
                child: Image.network(path),
              ), // Container
              Container(
                child: Text(film.overview!),
                padding: EdgeInsets.only(left: 16, right: 16),
              ) // Container
            ],
          ), // Column
        ), // Center
      ), // SingleChildScrollView
    ); // Scaffold
  }
}
```

Penjelasan:

Kode tersebut bertujuan untuk mengganti gambar dengan gambar yang berada di path jika gagal load

```dart
Run | Debug | Profile
void main() {
  runApp(MaterialApp(
    initialRoute: '/',
    routes: {
      '/': (context) => Homepage(),
      '/item': (context) => itempage(),
      '/movielist': (context) => FilmList(),

    },
  )); // MaterialApp
}
```

Penjelasan :

Kode ini menunjukkan bagaimana route dari aplikasi flutterApi ini.