

PRAKTIKUM 5

CLIENT – SERVER (FUTURE BUILDER)

A. TUJUAN PEMBELAJARAN

1. Mahasiswa dapat mengimplementasikan

B. DASAR TEORI

Dasar teori.

C. TUGAS PENDAHULUAN

1. Update Android Studio atau Visual Code yang Anda pakai.
2. Update Flutter SDK ke versi yang terbaru
3. Pastikan proses update tidak ada yang bermasalah.

D. PERCOBAAN

Percobaan 1 : FutureBuilder

Langkah Percobaan:

1. Buatlah project baru dengan nama `news_app`
2. Buatlah file dart baru dengan nama `article.dart` dan isi dengan class `Article`.

```
1  class Article {  
2      late String author;  
3      late String title;  
4      late String description;  
5      late String url;  
6      late String urlToImage;  
7      late String publishedAt;  
8      late String content;  
9  
10     Article({  
11         required this.author,  
12         required this.title,  
13         required this.description,  
14         required this.url,  
15         required this.urlToImage,  
16         required this.publishedAt,  
17         required this.content,  
18     });  
19 }
```

3. Kemudian kita akan membuat halaman detail dari aplikasi news. Buatlah file dart baru dengan nama `detail_page.dart` dan isi dengan class `ArticleDetailPage`

```

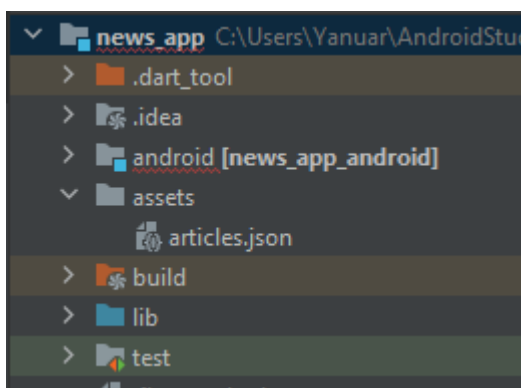
5 class ArticleDetailPage extends StatelessWidget {
6   static const routeName = '/article_detail';
7
8   final Article article;
9   const ArticleDetailPage({required this.article});
10
11   @override
12   Widget build(BuildContext context) {
13     return Scaffold(
14       appBar: AppBar(
15         title: Text(article.title),
16       ), // AppBar
17       body: SingleChildScrollView(
18         child: Column(
19           children: [
20             Image.network(article.urlToImage),
21             Padding(
22               padding: EdgeInsets.all(10),
23               child: Column(
24                 crossAxisAlignment: CrossAxisAlignment.start,
25                 children: [
26                   Text(article.description),
27                   Divider(color: Colors.grey),
28                   Text(
29                     article.title,
30                     style: TextStyle(
31                       color: Colors.black,
32                       fontWeight: FontWeight.bold,
33                       fontSize: 24,
34                     ), // TextStyle
35                   ), // Text
36                   Divider(color: Colors.grey),
37                   Text('Date: ${article.publishedAt}'),
38                   SizedBox(height: 10),
39                   Text('Author: ${article.author}'),
40                   Divider(color: Colors.grey),
41                   Text(
42                     article.content,
43                     style: TextStyle(fontSize: 16),
44                   ), // Text
45                   SizedBox(height: 10),
46                   ElevatedButton(
47                     child: Text('Read more'),
48                     onPressed: () {},
49                   ), // ElevatedButton
50                 ],
51               ), // Column
52             ), // Padding
53           ],
54         ), // Column
55       ), // SingleChildScrollView
56     ); // Scaffold
57   }
58 }

```

4. Sekarang kita membuat daftar news. Buatlah file dart baru dengan nama `list_page.dart` kemudian buat class dengan nama `NewsListPage`:

```
6 class NewsListPage extends StatelessWidget {
7   static const routeName = '/article_list';
8
9   @override
10  Widget build(BuildContext context) {
11    return Scaffold(
12      appBar: AppBar(
13        title: Text('News App'),
14      ), // AppBar
15      body: FutureBuilder<String>(
16        future: DefaultAssetBundle.of(context).loadString('assets/articles.json'),
17        builder: (context, snapshot) {
18          final List<Article> articles = parseArticles(snapshot.data);
19          return ListView.builder(
20            itemCount: articles.length,
21            itemBuilder: (context, index) {
22              return _buildArticleItem(context, articles[index]);
23            },
24          ); // ListView.builder
25        },
26      ), // FutureBuilder
27    ); // Scaffold
28  }
29 }
```

5. Buat folder baru dengan nama `assets` dan tambahkan file `articles.json`. File `articles.json` bisa didownload di alamat url ini <https://gist.githubusercontent.com/LittleFireflies/216da1b914103f53744303e4f928cbe5/raw/e178d582718985a44579e6f4cc1e60a065f4d984/articles.json>



6. Edit file `pubspec.yaml` dan tambahkan `articles.json` dibagian `assets` seperti berikut:

```

54 flutter:
55
56   # The following line ensures that the Material Icons fo
57   # included with your application, so that you can use t
58   # the material Icons class.
59   uses-material-design: true
60
61   # To add assets to your application, add an assets sect
62   assets:
63     - assets/articles.json
64     # - images/a_dot_burr.jpeg
65     # - images/a_dot_ham.jpeg
66
67   # An image asset can refer to one or more resolution-sp
68   # https://flutter.dev/assets-and-images/#resolution-awa
69
70   # For details regarding adding assets from package depe
71   # https://flutter.dev/assets-and-images/#from-packages

```

7. Tambahkan fungsi parseArticles dan _buildArticleItem di list_page.dart

```

31 List<Article> parseArticles(String? json) {
32   if (json == null) {
33     return [];
34   }
35   final List parsed = jsonDecode(json);
36   return parsed.map((json) => Article.fromJson(json)).toList();
37 }
38
39 Widget _buildArticleItem(BuildContext context, Article article) {
40   return ListTile(
41     contentPadding:
42       const EdgeInsets.symmetric(horizontal: 16.0, vertical: 8.0),
43     leading: Image.network(
44       article.urlToImage,
45       width: 100,
46     ), // Image.network
47     title: Text(article.title),
48     subtitle: Text(article.author),
49     onTap: () {},
50   ); // ListTile
51 }

```

8. Berikutnya kita akan mengedit `main.dart` dan menambahkan `routes` di `MaterialApp` nya.

```
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: 'News App',
18       theme: ThemeData(
19         primarySwatch: Colors.blue,
20         visualDensity: VisualDensity.adaptivePlatformDensity,
21       ), // ThemeData
22       initialRoute: NewsListPage.routeName,
23       routes: {
24         NewsListPage.routeName: (context) => NewsListPage(),
25         ArticleDetailPage.routeName: (context) => ArticleDetailPage(
26           article: ModalRoute.of(context)?.settings.arguments as Article,
27         ), // ArticleDetailPage
28       },
29     ); // MaterialApp
30   }
31 }
```

9. Sekarang jalankan aplikasi `news_app`. Anda akan melihat aplikasi tersebut berjalan dan menampilkan daftar berita (news) tetapi ketika daftar berita tersebut diklik masih belum bisa masuk ke detail beritanya.

10. Tambahkan action `onTap` di `ListTile` yang terletak di fungsi `_buildArticleItem`

```
onTap: () {
  Navigator.pushNamed(context, ArticleDetailPage.routeName, arguments: article);
},
```

11. Jalankan aplikasi `news_app`. Sekarang aplikasi kita sudah bisa menampilkan halaman detail tetapi tombol `Read more` di halaman detail tidak bisa diklik.

12. Buat file dart baru dengan nama `web_view.dart` dan buat class dengan nama `ArticleWebView`.

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

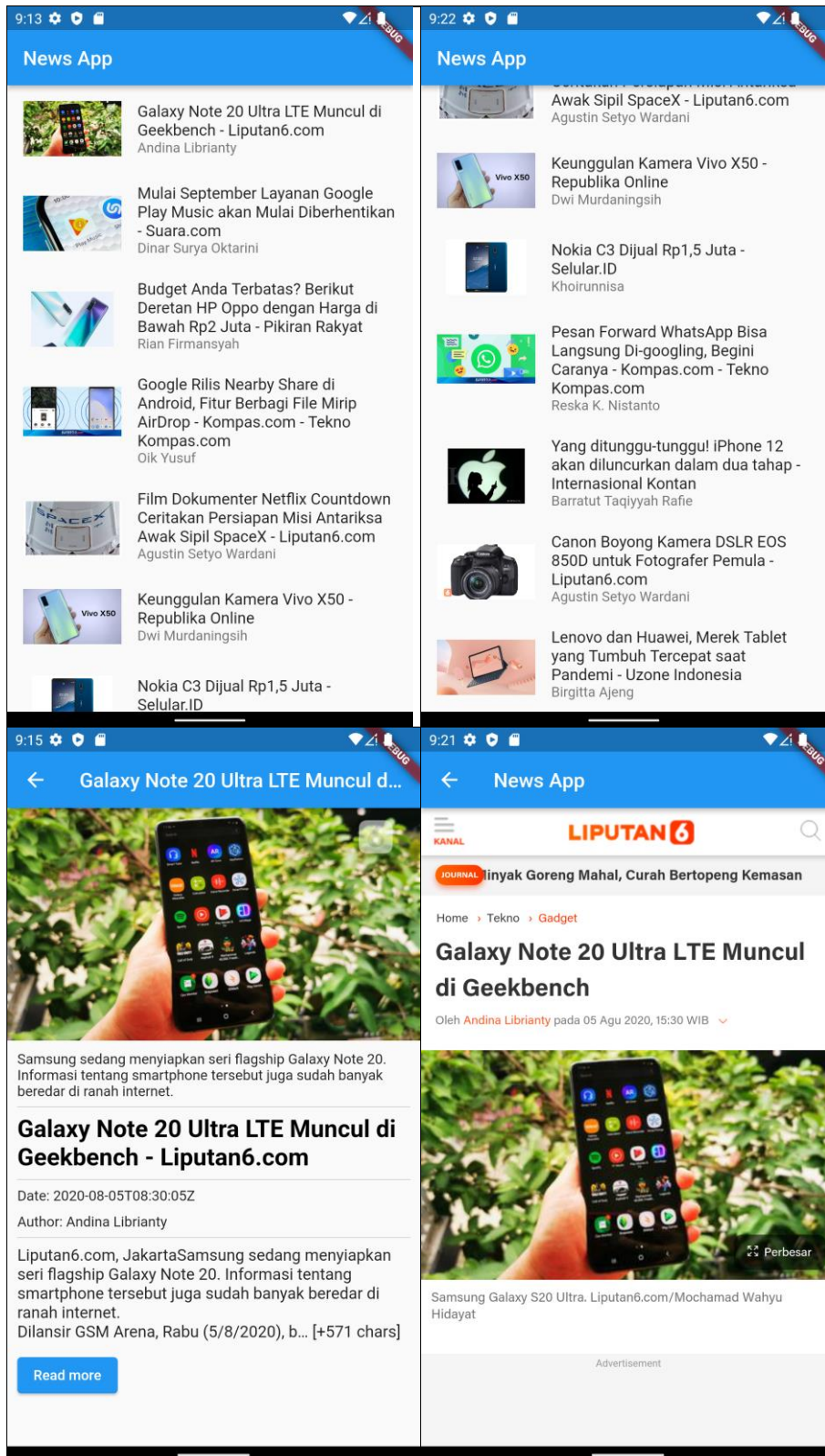
13. Tambahkan action `onPressed` di button Read more yang terletak di class `ArticleDetailPage`

```
onPressed: () {
  Navigator.pushNamed(context, ArticleWebView.routeName, arguments: article.url);
},
```

14. Edit file `main.dart` dan tambahkan baris berikut di `routes`

```
ArticleWebView.routeName: (context) => ArticleWebView(
  url: ModalRoute.of(context)?.settings.arguments as String,
), // ArticleWebView
```

15. Hasil tampilan aplikasi ketika sudah dijalankan seperti ini:



E. TUGAS

1. Tambahkan animasi Hero pada gambar di daftar berita sehingga menyambung dengan gambar yang ada di halaman detail:

F. LAPORAN RESMI

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