

25 KISAH NABI

By Kelompok 5

ANGGOTA

- Fitria Rahmawati (19220094)
- Erni Najiatusalamah (19220797)
- Mohammad Ali Hibrizi (19221137)
- Rijal Dzaki Fakhrusalam (19221267)
- Firda Awalia (19221407)
- Adam Baihaqi (19220298)

PENDAHULUAN

Perkembangan teknologi yang semakin pesat saat ini telah membuat banyak perubahan dalam gaya hidup semua kalangan masyarakat. Salah satu perubahannya adalah banyaknya penggunaan smartphone, terutama yang berbasis android untuk kehidupan sehari-hari. Para pengguna smartphone berasal dari berbagai kalangan dan umur, tidak sedikit masyarakat didaerah kita yang menganut agama Islam yang belum memahami lebih dalam tentang agama Islam oleh karena itu pengetahuan agama Islam sangat penting dan wajib di pelajari oleh setiap muslim yang perlu ditanamkan sejak dini.

LANDASAN TEORI

Aplikasi

Aplikasi adalah Program yang telah jadi dan siap untuk digunakan atau program yang dibuat sendiri dengan kumpulan perintah program di dalamnya yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu (khusus).

Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka.

LANJUTAN

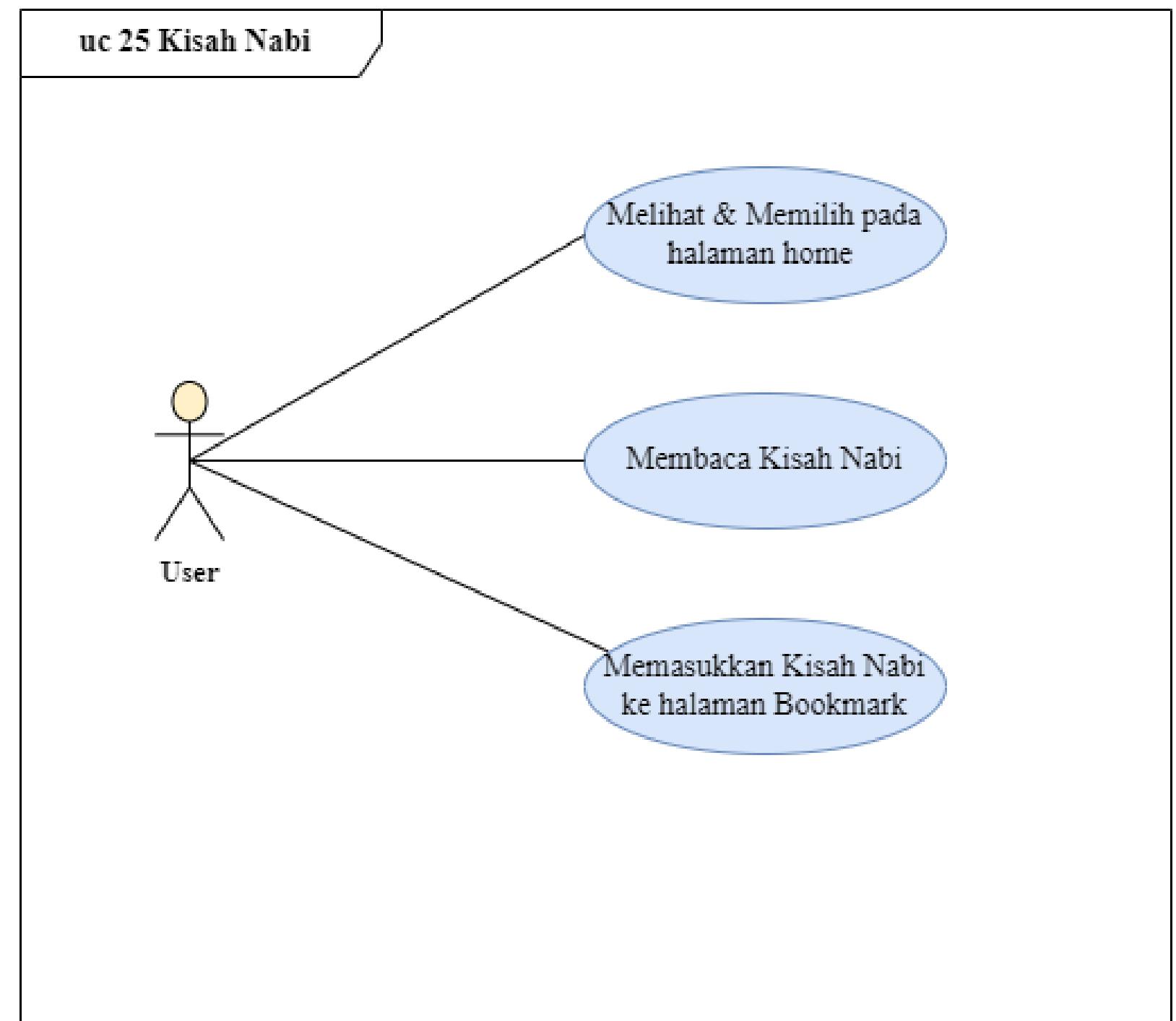
Pembelajaran

Pembelajaran merupakan suatu kegiatan yang melibatkan seseorang dalam upaya memperoleh pengetahuan, keterampilan dan nilai-nilai positif dengan memanfaatkan berbagai sumber untuk belajar.

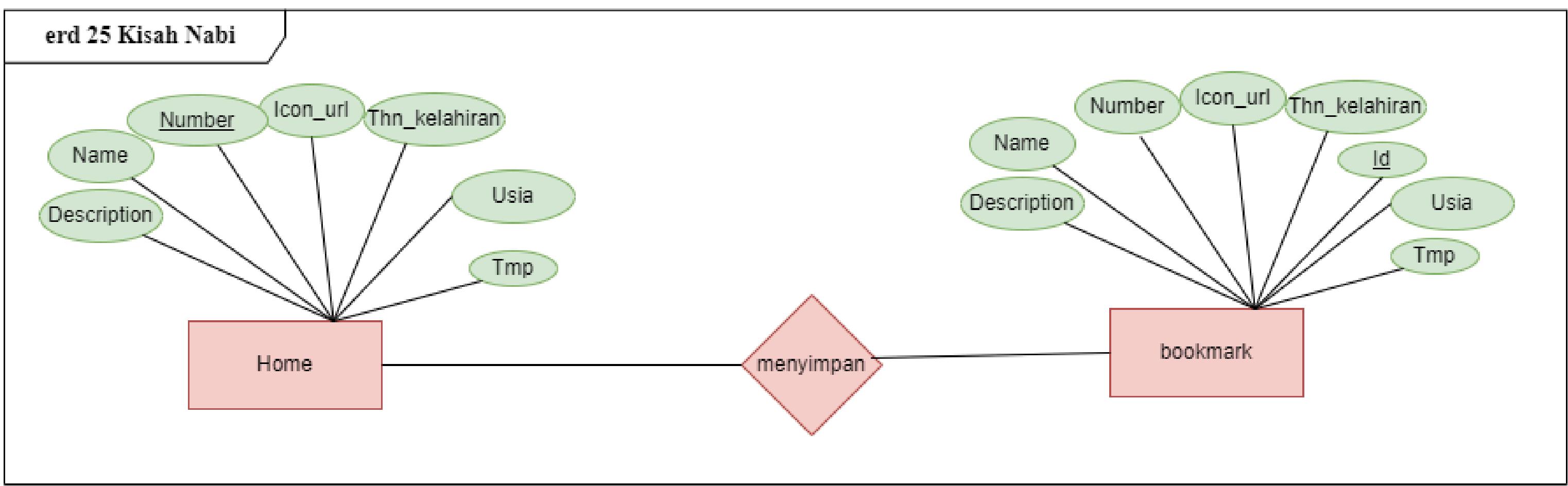
Nabi dan Rasul

Nabi berasal dari kata naba, artinya dari tempat yang tinggi. **Nabi** artinya manusia (lelaki) yang diberi wahyu oleh Allah Swt, tetapi tidak wajib disampaikan kepada umatnya. Sedangkan **Rasul** berasal dari kata risalah, artinya penyampaian.

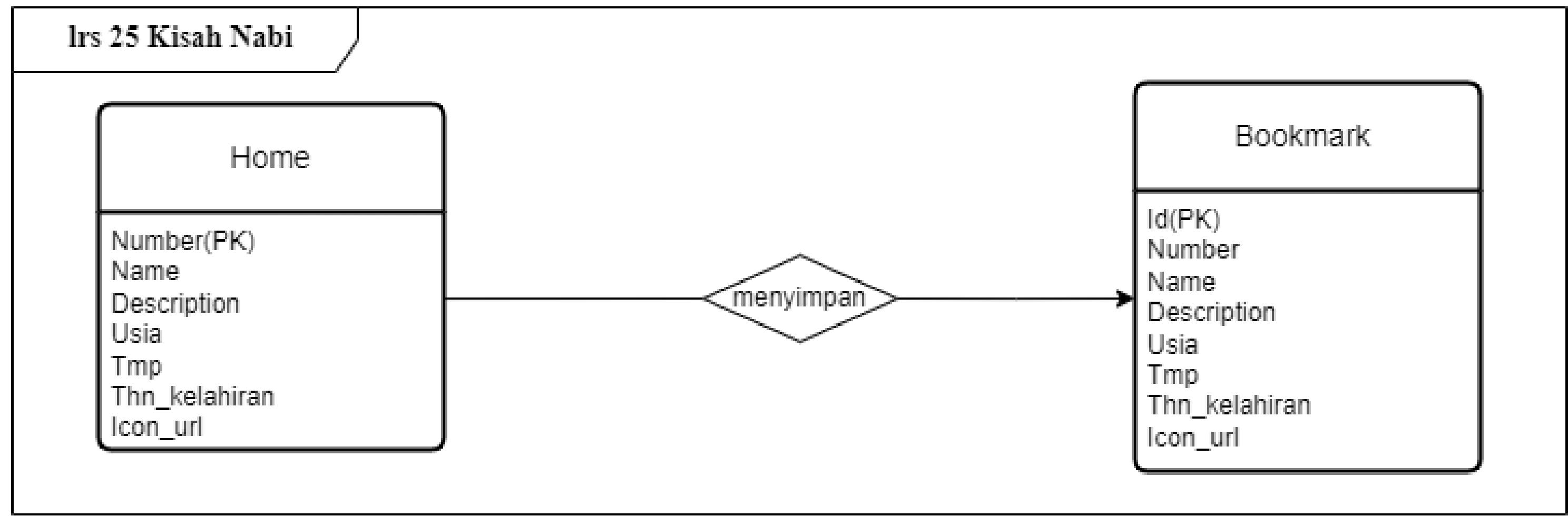
USE CASE DIAGRAM



ERD



LRS



CODING

main.dart

```
1 import 'package:kisah_nabi/ui/bookmark.dart';
2 import 'package:kisah_nabi/ui/home.dart';
3 import 'package:flutter/material.dart';
4
5 void main() {
6   runApp(const MyApp());
7 }
8
9 class MyApp extends StatefulWidget {
10   const MyApp({Key? key}) : super(key: key);
11
12   @override
13   State<MyApp> createState() => _MyAppState();
14 }
15
16 class _MyAppState extends State<MyApp> with SingleTickerProviderStateMixin {
17   TabController? tabController;
18
19   @override
20   void initState() {
21     tabController = TabController(length: 2, vsync: this);
22     super.initState();
23   }
24
25   @override
26   void dispose() {
27     tabController?.dispose();
28     super.dispose();
29   }
30
31   @override
32   Widget build(BuildContext context) {
33     return MaterialApp(
34       theme: ThemeData(
35         useMaterial3: true,
36         scaffoldBackgroundColor: Colors.white,
37         splashColor: Colors.transparent,
38         highlightColor: Colors.transparent,
39         floatingActionButtonTheme: const FloatingActionButtonThemeData(backgroundColor: Color(0xfffffb6b)),
40         cardTheme: const CardTheme(surfaceTintColor: Colors.white),
41         dialogTheme: const DialogTheme(surfaceTintColor: Colors.white, backgroundColor: Colors.white),
42       ),
43       debugShowCheckedModeBanner: false,
44       home: Scaffold(
45         appBar: AppBar(
46           backgroundColor: const Color(0xd0c431ff),
47           title: const Row(
48             children: [
49               Image(
50                 image: AssetImage('assets/images/ic_logo.png'),
51                 height: 50,
52                 width: 50,
53               ),
54               SizedBox(
55                 width: 20,
56               ),
57               Text(
58                 "25 Kisah Nabi",
59                 style: TextStyle(
60                   fontSize: 24,
61                   fontWeight: FontWeight.bold,
62                   color: Colors.white),
63               ),
64             ],
65           ),
66           bottom: setTabBar(),
67         ),
68         body: TabBarView(
69           controller: tabController,
70           children: const [
71             HomePage(),
72             BookmarkPage(),
73           ],
74         ),
75       ),
76     );
77   }
78
79 TabBar setTabBar() {
80   return TabBar(
81     controller: tabController,
82     labelColor: Colors.white,
83     unselectedLabelColor: Colors.black26,
84     indicatorColor: Colors.white,
85     tabs: const [
86       Tab(
87         text: "Home",
88         icon: Icon(Icons.menu_book),
89       ),
89       Tab(
90         text: "Bookmark",
91         icon: Icon(Icons.bookmarks_outlined),
92       ),
93     ],
94   );
95 }
96 }
97 }
```

```
1 import 'package:kisah_nabi/model/bookmark.dart';
2 import 'package:sqlite/sqlite.dart';
3 import 'package:path/path.dart';
4
5 class DatabaseHelper {
6   static final DatabaseHelper _instance = DatabaseHelper._internal();
7   static Database? _database;
8
9   //inisialisasi beberapa variabel yang dibutuhkan
10  final String tableName = 'tbl_bookmark';
11  final String columnId = 'id';
12  final String columnNumber = 'number';
13  final String columnName = 'name';
14  final String columnThn_kelahiran = 'thnKelahiran';
15  final String columnDescription = 'description';
16  final String columnUsia = 'usia';
17  final String columnTmp = 'tmp';
18  final String columnIcon_url = 'iconUrl';
19
20  DatabaseHelper._internal();
21
22  factory DatabaseHelper() => _instance;
23
24  //cek apakah ada database
25  Future<Database?> get checkDB async {
26    if (_database != null) {
27      return _database;
28    }
29    _database = await _initDB();
30    return _database;
31  }
32
33 Future<Database?> _initDB() async {
34   String databasePath = await getDatabasesPath();
35   String path = join(databasePath, 'kisah_nabi.db');
36   return await openDatabase(path, version: 1, onCreate: _onCreate);
37 }
38
39 //membuat tabel dan field-fieldnya
40 Future<void> _onCreate(Database db, int version) async {
41   var sql = "CREATE TABLE $tableName($columnId INTEGER PRIMARY KEY, "
42   "$columnNumber TEXT,"
43   "$columnName TEXT,"
44   "$columnThn_kelahiran TEXT,"
45   "$columnDescription TEXT,"
46   "$columnUsia TEXT,"
47   "$columnTmp TEXT,"
48   "$columnIcon_url TEXT";
49   await db.execute(sql);
50 }
51
52 //insert ke database
53 Future<int?> saveData(Bookmark bookmark) async {
54   var dbClient = await checkDB;
55   return await dbClient!.insert(tableName, bookmark.toMap());
56 }
57
58 //read data bookmark
59 Future<List?> getDataBookmark() async {
60   var dbClient = await checkDB;
61   var result = await dbClient!.rawQuery('SELECT * FROM $tableName ORDER BY number');
62   return result.toList();
63 }
64
65 //cek database bookmark
66 Future<int?> cekDataBookmark() async {
67   var dbClient = await checkDB;
68   return Sqflite.firstIntValue(await dbClient!.rawQuery('SELECT COUNT(*) FROM $tableName'));
69 }
70
71
72 //hapus database bookmark
73 Future<int?> deleteBookmark(int id) async {
74   var dbClient = await checkDB;
75   return await dbClient!.delete(tableName, where: '$columnId = ?', whereArgs: [id]);
76 }
77
78 }
```

CODING MODEL

home.dart

```
1 class Home {  
2   int?number;  
3   String name;  
4   String thnKehadiran;  
5   dynamic usia;  
6   String description;  
7   String tmp;  
8   String iconUrl;  
9  
10  Home({  
11    required this.number,  
12    required this.name,  
13    required this.thnKehadiran,  
14    required this.usia,  
15    required this.description,  
16    required this.tmp,  
17    required this.iconUrl,  
18  });  
19  
20  factory Home.fromJson(Map<String, dynamic> json) => Home(  
21    name: json["name"],  
22    number: json['number'],  
23    thnKehadiran: json["thn_kehadiran"],  
24    usia: json["usia"],  
25    description: json["description"],  
26    tmp: json["tmp"],  
27    iconUrl: json["icon_url"],  
28  );  
29  
30  Map<String, dynamic> toJson() => {  
31    "name": name,  
32    "number": number,  
33    "thn_kehadiran": thnKehadiran,  
34    "usia": usia,  
35    "description": description,  
36    "tmp": tmp,  
37    "icon_url": iconUrl,  
38  };  
39 }  
40  
41  
42
```

```
1 class Bookmark {  
2   int? id;  
3   String? number;  
4   String? name;  
5   String? thnKehadiran;  
6   dynamic usia;  
7   String? description;  
8   String? tmp;  
9   String? iconUrl;  
10  
11  Bookmark({  
12    this.id,  
13    required this.number,  
14    required this.name,  
15    required this.thnKehadiran,  
16    required this.usia,  
17    required this.description,  
18    required this.tmp,  
19    required this.iconUrl,  
20  });  
21  
22  Map<String, dynamic> toMap() {  
23    var map = <String, dynamic>{};  
24    if (id != null) {  
25      map['id'] = id;  
26    }  
27    map['number'] = number;  
28    map['name'] = name;  
29    map['thn_kehadiran'] = thnKehadiran;  
30    map['usia'] = usia;  
31    map['description'] = description;  
32    map['tmp'] = tmp;  
33    map['icon_url'] = iconUrl;  
34  
35    return map;  
36  }  
37  
38  Bookmark.fromMap(Map<String, dynamic> map) {  
39    id = map['id'];  
40    number = map['number'];  
41    name = map['name'];  
42    thnKehadiran = map['thn_kehadiran'];  
43    usia = map['usia'];  
44    description = map['description'];  
45    tmp = map['tmp'];  
46    iconUrl = map['icon_url'];  
47  }  
48  
49 }  
50  
51
```

bookmark.dart

CODING

UI

home.dart

```
1 import 'dart:convert';
2
3 import 'package:kisah_nabi/ui/detail_home.dart';
4 import 'package:kisah_nabi/model/home.dart';
5 import 'package:flutter/material.dart';
6 // ignore: library_prefixes
7 import 'package:flutter/services.dart' as rootBundle;
8 import 'package:flutter_svg/flutter_svg.dart';
9
10 class HomePage extends StatefulWidget {
11   const HomePage({super.key});
12
13   @override
14   State<HomePage> createState() => _HomePageState();
15 }
16
17 class _HomePageState extends State<HomePage> {
18
19   //method get data assets
20   Future<List<Home>> readJsonData() async {
21     final jsonData = await rootBundle.rootBundle.loadString('assets/data/kisah_nabi.json');
22     final listJson = json.decode(jsonData) as List<dynamic>;
23     return listJson.map((e) => Home.fromJson(e).toList());
24   }
25
26   @override
27   Widget build(BuildContext context) {
28     return Scaffold(
29       body: Column(
30         children: [
31           Expanded(
32             child: FutureBuilder(
33               future: readJsonData(),
34               builder: (context, data) {
35                 if (data.hasError) {
36                   return Center(
37                     child: Text('${data.error}');
38                 );
39                 } else if (data.hasData) {
40                   var items = data.data as List<Home>;
41                   return ListView.builder(
42                     itemCount: items.length,
43                     itemBuilder: (context, index) {
44                       return GestureDetector(
45                         onTap: () {
46                           String strNo = items[index].number.toString();
47                           String strTmp = items[index].tmp.toString();
48                           String strName = items[index].name.toString();
49                           String strThn_kelahiran = items[index].thnKelahiran.toString();
50                           String strDescription = items[index].description.toString();
51                           String strUsia = items[index].usia.toString();
52                           String strIcon_url = items[index].iconUrl.toString();
53
54                           Navigator.push(
55                             context,
56                             MaterialPageRoute(
57                               builder: (context) => DetailHome(
58                                 strNo: strNo,
59                                 strDescription: strDescription,
60                                 strName: strName,
61                                 strThn_kelahiran: strThn_kelahiran,
62                                 strUsia: strUsia,
63                                 strIcon_url: strIcon_url,
64                                 strTmp: strTmp
65                               ),
66                             ),
67                           );
68                         },
69                       );
70                     },
71                   );
72                 }
73               );
74             );
75           ],
76         ),
77       ),
78       child: Card(
79         shape: RoundedRectangleBorder(
80           borderRadius: BorderRadius.circular(10)
81         ),
82         elevation: 5,
83         margin: const EdgeInsets.all(10),
84         child: Row(
85           children: [
86             Padding(
87               padding: const EdgeInsets.only(left: 10),
88             ),
89             Stack(
90               alignment: Alignment.center,
91               children: [
92               ClipRRect(
93                 borderRadius: BorderRadius.circular(100),
94                 child: SvgPicture.asset(
95                   'assets/images/no.svg',
96                   width: 50,
97                   height: 50,
98                   fit: BoxFit.cover,
99                 ),
100               ),
101               Positioned(
102                 child: Text(
103                   items[index].number.toString(),
104                   style: const TextStyle(
105                     fontSize: 14,
106                     fontWeight: FontWeight.bold
107                 ),
108               ),
109             ],
110           ),
111           ],
112         ),
113       ),
114       Expanded(
115         child: Column(
116           children: [
117             ListTile(
118               title: Text(
119                 items[index].name.toString(),
120                 style: const TextStyle(
121                   fontSize: 16,
122                   fontWeight: FontWeight.bold
123                 ),
124               ),
125               subtitle: Text(
126                 items[index].thnKelahiran.toString(),
127                 style: const TextStyle(
128                   fontSize: 12
129                 ),
130               ),
131             ),
132             ],
133           ),
134         ),
135       ),
136     ],
137   ),
138 },
139 ],
140 ),
141 );
142 }
143 }
```

lanjutannya

The image shows a person's hand holding a smartphone. The screen of the phone displays a presentation slide. The slide has a white background. At the top, the text "bookmark.dart" is written in a large, bold, brown sans-serif font. In the center, the word "CODING" is written in a large, bold, brown sans-serif font. At the bottom, the letters "UI" are written in a large, bold, brown sans-serif font. To the right of the text, there is a large, thick, orange curved arrow pointing downwards and to the left, indicating a flow or relationship between the text elements.

```
1 import 'package:kisah_nabi/model/bookmark.dart';
2 import 'package:kisah_nabi/database/database_helper.dart';
3 import 'package:flutter/material.dart';
4 import 'package:flutter_svg/flutter_svg.dart';
5
6 class BookmarkPage extends StatefulWidget {
7     const BookmarkPage({super.key});
8
9
10    @override
11        State<BookmarkPage> createState() => _BookmarkPageState();
12 }
13
14 class _BookmarkPageState extends State<BookmarkPage> {
15     List<Bookmark> listBookmark = [];
16     DatabaseHelper databaseHelper = DatabaseHelper();
17     int strCheckDatabase = 0;
18
19    @override
20        void initState() {
21            super.initState();
22            getDatabase();
23            getAllData();
24        }
25
26    //cek database ada data atau tidak
27    Future<void> getDatabase() async {
28        var checkDB = await databaseHelper.cekDataBookmark();
29        setState(() {
30            if (checkDB == 0) {
31                strCheckDatabase = 0;
32            } else {
33                strCheckDatabase = checkDB!;
34            }
35        });
36    }
37
38    //get data untuk menampilkan ke listview
39    Future<void> getAllData() async {
40        var listData = await databaseHelper.getDataBookmark();
41        setState(() {
42            listBookmark.clear();
43            listData!.forEach((data) {
44                listBookmark.add(Bookmark.fromMap(data));
45            });
46        });
47    }
48
49    @override
50        Widget build(BuildContext context) {
51            return Scaffold(
52                body: SingleChildScrollView(
53                    child: Column(
54                        children: [
55                            strCheckDatabase == 0
56                                ? Container(
57                                    padding: const EdgeInsets.only(top: 250),
58                                    child: const Center(
59                                        child: Text("Ups, belum ada bookmark."),
60                                        style: TextStyle(
61                                            fontSize: 20,
62                                            fontWeight: FontWeight.bold,
63                                            color: Colors.black,
64                                        ),
65                                    ),
66                                )
67                                : ListView.builder(
68                                    physics: const NeverScrollableScrollPhysics(),
69                                    shrinkWrap: true,
70                                    itemCount: listBookmark.length,
71                                    itemBuilder: (context, index) {
72                                        Bookmark bookmark = listBookmark[index];
73                                        return GestureDetector(
74                                            onTap: () {
75                                                AlertDialog alertDialog = AlertDialog(
76                                                    title: const Text("Hapus Data",
77                                                    style: TextStyle(fontSize: 18, color: Colors.black),
78                                                ),
79                                                    content: const SizedBox(
80                                                        height: 20,
81                                                        child: Column(
82                                                            children: [
83                                                                Text("Yakin hapus data ini?",
84                                                                style: TextStyle(fontSize: 14, color: Colors.black),
85                                                            ),
86                                                        ],
87                                                    ),
88                                                ),
89                                                actions: [
90                                                    TextButton(
91                                                        onPressed: () {
92                                                            deleteData(bookmark, index);
93                                                            Navigator.of(context, rootNavigator: true).pop();
94                                                        },
95                                                        child: const Text("Hapus",
96                                                        style: TextStyle(fontSize: 14, color: Colors.black),
97                                                    ),
98                                                ),
99                                            ],
100                                        );
101                                    });
102                                }
103                            ],
104                        ),
105                    ),
106                ),
107            );
108        }
109    }
110
111    void deleteData(Bookmark bookmark, int index) {
112        databaseHelper.deleteDataBookmark(bookmark.id);
113        setState(() {
114            listBookmark.removeAt(index);
115        });
116    }
117}
```

```
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173 //untuk hapus data berdasarkan Id
174 Future<void> deleteData(Bookmark bookmark, int position) async {
175   await databaseHelper.deleteBookmark(bookmark.id!);
176   setState(() {
177     getDatabase();
178     listBookmark.removeAt(position);
179   });
180 }
181 }
```

detail_home.dart

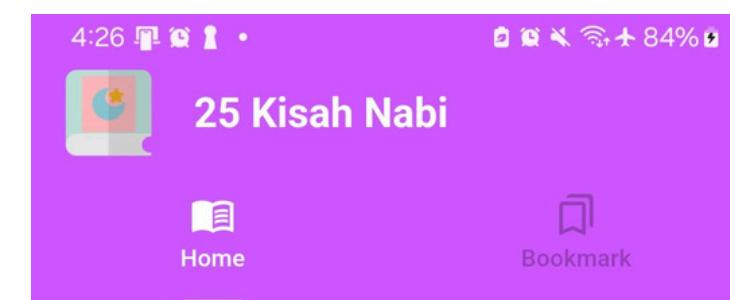
CODING

UI

```
1 import 'dart:async';
2
3 import 'package:kisah_nabi/model/bookmark.dart';
4 import 'package:kisah_nabi/ui/bookmark.dart';
5 import 'package:kisah_nabi/database/database_helper.dart';
6 import 'package:kisah_nabi/main.dart';
7 import 'package:flutter/material.dart';
8
9
10 class DetailHome extends StatefulWidget {
11   final String strNo, strUsia, strName, strThn_kelahiran, strDescription, strTmp, strIcon_url;
12
13   const DetailHome(
14     {super.key, required this.strNo, required this.strUsia,
15      required this.strName, required this.strThn_kelahiran, required this.strDescription,
16      required this.strTmp, required this.strIcon_url});
17
18   @override
19   State<DetailHome> createState() => _DetailHomeState();
20 }
21
22 class _DetailHomeState extends State<DetailHome> {
23   List<Bookmark> listBookmark = [];
24   DatabaseHelper databaseHelper = DatabaseHelper();
25   String? strNo, strUsia, strName, strThn_kelahiran, strDescription, strTmp, strIcon_url;
26
27   @override
28   initState() {
29     super.initState();
30     strNo = widget.strNo;
31     strUsia = widget.strUsia;
32     strName = widget.strName;
33     strThn_kelahiran = widget.strThn_kelahiran;
34     strDescription = widget.strDescription;
35     strTmp = widget.strTmp;
36     strIcon_url = widget.strIcon_url;
37   }
38
39   @override
40   Widget build(BuildContext context) {
41     return Scaffold(
42       body: NestedScrollView(
43         headerSliverBuilder: (BuildContext context, bool innerBoxIsScrolled) {
44           return [
45             const SliverAppBar(
46               backgroundColor: Color(0xd0c431ff)
47             ),
48           ],
49         },
50         body: Padding(
51           padding: const EdgeInsets.all(10),
52           child: SingleChildScrollView(
53             child: Column(
54               crossAxisAlignment: CrossAxisAlignment.start,
55               children: [
56                 const SizedBox(
57                   height: 20,
58                 ),
59                 Center(
60                   child: Text(
61                     strName!,
62                     style: const TextStyle(
63                       fontSize: 32, fontWeight: FontWeight.bold, color: Colors.black
64                     ),
65                   ),
66                 ),
67               ],
68             ),
69           ),
70         ),
71       ),
72     ),
73   ),
74 }
75
76   Center(
77     child: Text(
78       strTmp!,
79       style: const TextStyle(
80         fontSize: 16, color: Colors.black
81       ),
82     ),
83   ),
84   Center(
85     child: Text(
86       strUsia!,
87       style: const TextStyle(
88         fontSize: 16, color: Colors.black
89       ),
90     ),
91   ),
92   Center(
93     child: Text(
94       strThn_kelahiran!,
95       style: const TextStyle(
96         fontSize: 16, color: Colors.black
97       ),
98     ),
99   ),
100  const SizedBox(
101    height: 35,
102  ),
103  Text(
104    strDescription!,
105    style: const TextStyle(
106      fontSize: 15, color: Colors.black
107    ),
108  ),
109  const SizedBox(
110    height: 20,
111  ),
112  ],
113  ),
114  ),
115  ),
116  floatingActionButton: FloatingActionButton(
117   onPressed: () {
118     bool isTapped = false;
119     setState(() async {
120       if (!isTapped) {
121         isTapped = true;
122         await Future.delayed(const Duration(milliseconds: 3000));
123
124         databaseHelper.saveData(Bookmark(
125           number: strNo,
126           name: strName,
127           thnKelahiran: strThn_kelahiran,
128           description: strDescription,
129           usia: strUsia,
130           tmp: strTmp,
131           iconUrl: strIcon_url,
132         ));
133
134         Navigator.push(context, MaterialPageRoute(builder: (context) => const MyApp()));
135
136         isTapped = true;
137       }
138     });
139   },
140   child: const Icon(
141     Icons.favorite,
142     color: Colors.white,
143   ),
144 ),
145 );
146 }
```

lanjutannya

SCREENSHOT HASIL



1 Nabi Adam AS
5876

2 Nabi Idris AS
4533

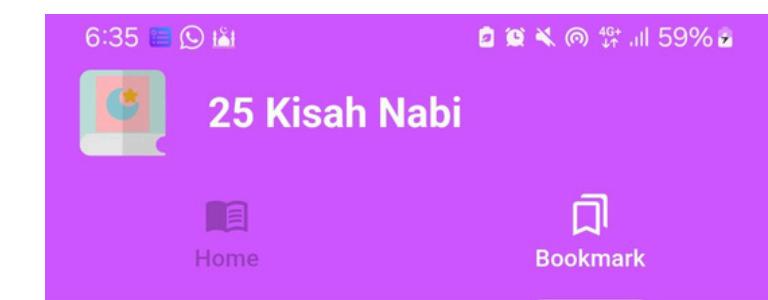
3 Nabi Nuh AS
3993

4 Nabi Hud AS
2450

5 Nabi Sholeh AS
2450

6 Nabi Ibrahim AS
1997

7 Nabi Luth AS
1897



Ups, belum ada bookmark.

KESIMPULAN

Semua orang telah mengalami banyak perubahan gaya hidup karena perkembangan teknologi yang semakin pesat saat ini. Banyaknya penggunaan smartphone, terutama yang berbasis Android. Pengguna smartphone berasal dari berbagai kalangan dan umur, dan banyak orang di daerah kita yang belum memahami lebih dalam tentang agama Islam. Oleh karena itu, pengetahuan tentang agama Islam sangat penting dan wajib dipelajari oleh setiap muslim sejak kecil. Sementara kata "nabi" berasal dari kata "nab", yang berarti "dari tempat yang tinggi", "rasul" berasal dari kata "risalah", yang berarti "penyampaian". "Nabi" juga berarti "manusia" yang diberi wahyu oleh Allah Swt, tetapi tidak harus disampaikan kepada orang lain.

SARAN

- Menyediakan narasi audio untuk setiap kisah, yang dapat didengarkan oleh pengguna.
- Sertakan ilustrasi atau gambar untuk memperkaya pengalaman membaca.
- Fitur pencarian untuk memudahkan pengguna menemukan kisah atau informasi tertentu.
- Sertakan ilustrasi atau animasi untuk mendukung cerita dan membuatnya lebih menarik bagi berbagai kelompok usia.



A central white rectangular area contains the text "THANK YOU" in a bold, brown, sans-serif font. The background is white and features abstract, hand-drawn style organic shapes in shades of orange, brown, and tan. These shapes include wavy lines, blob-like forms, and small white dots scattered across the surface.

THANK YOU