PEMROGRAMAN BERBASIS MOBILE 2

PERTEMUAN 4



**DESKRIPSI MATAKULIAH**

**Mata Kuliah ini mempelajari konsep bahasa pemrograman Dart, membuat aplikasi mobile dengan flutter menggunakan bahasa Dart, menerapkan widget pada flutter untuk membuat aplikasi mobile**

**CAPAIAN PEMBELAJARAN**

**Mahasiswa dapat memahami penggunaan Framework Flutter dengan IDE Visual Studio Code untuk membuat aplikasi mobile.**

**DOSEN PENGAMPU**

**Bambang Ismanto, M.Kom**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

**(STMIK) WIDYA PRATAMA**

[Pokok bahasan/bab/pertemuan]

**CAPAIAN PEMBELAJARAN**

**Mahasiswa dapat memahami penggunaan Framework Flutter dengan IDE Visual Studio Code untuk membuat aplikasi mobile.**

**MATERI PEMBELAJARAN**

1. **Menyiapkan Halaman Input**
2. **Menyiapkan Halaman Ubah dan Hapus**

**MATERI**

1. **Menyiapkan Halaman Input**

Buat file baru dengan nama **tambah\_data.dart**. Isi lengkap file tersebut adalah sebagai berikut:

import 'package:flutter/material.dart';

import 'package:cloud\_firestore/cloud\_firestore.dart';

import 'home.dart';

class TambahData extends StatefulWidget {

  @override

  \_TambahDataState createState() => \_TambahDataState();

}

class \_TambahDataState extends State<TambahData> {

  TextEditingController nimController = TextEditingController();

  TextEditingController namaController = TextEditingController();

  TextEditingController prodiController = TextEditingController();

  void addData() {

    DocumentReference documentReference = FirebaseFirestore.instance.collection('mahasiswa').doc(nimController.text);

    // create Map to send data in key:value pair form

    Map<String, dynamic> mhs = ({

      "nim": nimController.text,

      "nama": namaController.text,

      "prodi": prodiController.text

    });

    // send data to Firebase

    documentReference.set(mhs).whenComplete(() => print('${nimController.text} created'));

    Navigator.pop(context);

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: Text("ADD DATA"),

      ),

      body: Padding(

        padding: EdgeInsets.all(10.0),

        child: ListView(

          children: <Widget>[

            Text(

              "Input Data Mahasiswa",

              style: TextStyle(

                color: Colors.red,

                fontStyle: FontStyle.italic,

                fontWeight: FontWeight.bold,

                fontSize: 25,

              ),

            ),

            SizedBox(

              height: 40,

            ),

            TextFormField(

              controller: nimController,

              decoration: InputDecoration(

                labelText: "NIM",

              ),

            ),

            TextFormField(

              controller: namaController,

              decoration: InputDecoration(labelText: "Nama"),

            ),

            TextFormField(

              controller: prodiController,

              decoration: InputDecoration(labelText: "Prodi"),

            ),

            SizedBox(

              height: 50,

            ),

            ElevatedButton(

              onPressed: () {

                addData();

                Navigator.pushAndRemoveUntil(

                  context,

                  MaterialPageRoute(builder: (context) => Home()),

                  (Route<dynamic> route) => false,

                );

              },

              child: Text("Simpan"),

            ),

          ],

        ),

      ),

    );

  }

}

Untuk bisa menampilkan halaman input, tambahkan import pada **home.dart**

import 'tambah\_data.dart';

tambahkan widget **FloatingActionButton** pada Scaffold di bawah appBar.

floatingActionButton: FloatingActionButton(

        onPressed: () {

          Navigator.push(

            context,

            MaterialPageRoute(

              builder: (context) => TambahData(),

            ),

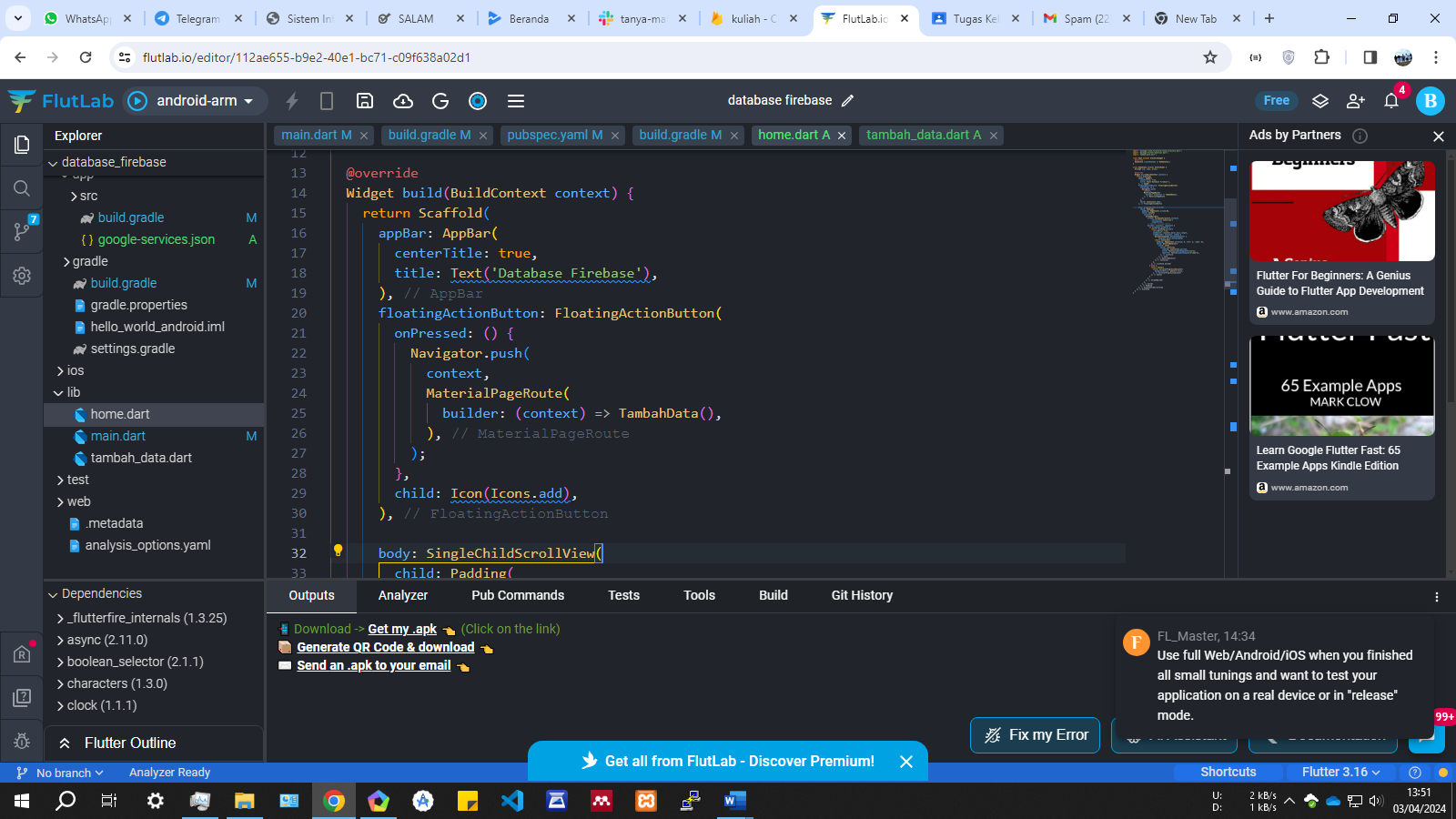
          );

        },

        child: Icon(Icons.add),

      ),

Sehingga menjadi:



**Penjelasan Singkat:**

addData() {

    DocumentReference documentReference = FirebaseFirestore.instance.collection('mahasiswa').doc(nimController.text);

    Map<String, dynamic> mhs = ({

      "nim": nimController.text,

      "nama": namaController.text,

      "prodi": prodiController.text

    });

    // send data to Firebase

    documentReference.set(mhs).whenComplete(() => print('${nimController.text} created'));

    Navigator.pop(context);

  }

Method addData digunakan untuk proses menambahkan data ke database firestore. Method documentReference.set digunakan mengirimkan ke firestore kemudian menambahkan data yang dikirim.

1. **Menyiapkan Halaman Ubah dan Hapus**

Kode pada halaman ubah mahasiswa hampir sama dengan halaman input data. Perbedaannya di sini kita membuat konstruktor dengan beberapa parameter untuk menerima nilai yang akan diubah, yang dikirimkan dari halaman home (diambilkan dari List). Halaman ini juga akan mengimplementasikan fungsi hapus. Berikut adalah kode lengkap dari file **edit\_data.dart**

import 'package:flutter/material.dart';

import 'package:cloud\_firestore/cloud\_firestore.dart';

import 'home.dart';

class EditData extends StatefulWidget {

  final String? nim;

  final String? nama;

  final String? prodi;

  EditData({this.nim, this.nama, this.prodi});

  @override

  \_EditDataState createState() => \_EditDataState();

}

class \_EditDataState extends State<EditData> {

  TextEditingController nimController = new TextEditingController();

  TextEditingController namaController = new TextEditingController();

  TextEditingController prodiController = new TextEditingController();

  void editData() {

    DocumentReference documentReference =

        FirebaseFirestore.instance.collection('mahasiswa').doc(widget.nim);

    Map<String, dynamic> mhs = ({

      "nim": widget.nim,

      "nama": namaController.text,

      "prodi": prodiController.text

    });

    // update data to Firebase

    documentReference

        .update(mhs)

        .whenComplete(() => print('${widget.nim} updated'));

  }

  void deleteData() {

    DocumentReference documentReference =

        FirebaseFirestore.instance.collection('mahasiswa').doc(widget.nim);

    // delete data from Firebase

    documentReference

        .delete()

        .whenComplete(() => print('${widget.nim} deleted'));

  }

  void konfirmasi() {

    AlertDialog alertDialog = AlertDialog(

      content: Text("Apakah anda yakin akan menghapus data '${widget.nama}'"),

      actions: <Widget>[

        ElevatedButton(

          style: ElevatedButton.styleFrom(

            foregroundColor: Colors.red,

          ),

          child: Text(

            "OK DELETE!",

            style: TextStyle(color: Colors.black),

          ),

          onPressed: () {

            deleteData();

            Navigator.pushAndRemoveUntil(

              context,

              MaterialPageRoute(builder: (context) => Home()),

              (Route<dynamic> route) => false,

            );

          },

        ),

        ElevatedButton(

          style: ElevatedButton.styleFrom(

            foregroundColor: Colors.green,

          ),

          child: Text("CANCEL", style: TextStyle(color: Colors.black)),

          onPressed: () => Navigator.pop(context),

        ),

      ],

    );

    showDialog(

      context: context,

      builder: (BuildContext context) {

        return alertDialog;

      },

    );

  }

  @override

  void initState() {

    nimController = TextEditingController(text: widget.nim);

    namaController = TextEditingController(text: widget.nama);

    prodiController = TextEditingController(text: widget.prodi);

    super.initState();

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: Text("EDIT DATA"),

      ),

      body: Padding(

        padding: EdgeInsets.all(10.0),

        child: ListView(

          children: <Widget>[

            Text(

              "Ubah Data Mahasiswa",

              style: TextStyle(

                color: Colors.red,

                fontStyle: FontStyle.italic,

                fontWeight: FontWeight.bold,

                fontSize: 25,

              ),

            ),

            SizedBox(

              height: 40,

            ),

            TextFormField(

              controller: nimController,

              decoration: InputDecoration(

                labelText: "NIM",

              ),

            ),

            TextFormField(

              controller: namaController,

              decoration: InputDecoration(labelText: "Nama"),

            ),

            TextFormField(

              controller: prodiController,

              decoration: InputDecoration(labelText: "Prodi"),

            ),

            SizedBox(

              height: 50,

            ),

            Row(

              mainAxisAlignment: MainAxisAlignment.spaceAround,

              children: [

                ElevatedButton(

                  style: ElevatedButton.styleFrom(

                    foregroundColor: Colors.orange,

                  ),

                  onPressed: () {

                    editData();

                    Navigator.pushAndRemoveUntil(

                      context,

                      MaterialPageRoute(builder: (context) => Home()),

                      (Route<dynamic> route) => false,

                    );

                  },

                  child: Text("Ubah"),

                ),

                ElevatedButton(

                  style: ElevatedButton.styleFrom(

                    foregroundColor: Colors.red,

                  ),

                  onPressed: () {

                    konfirmasi();

                  },

                  child: Text("Hapus"),

                ),

              ],

            ),

          ],

        ),

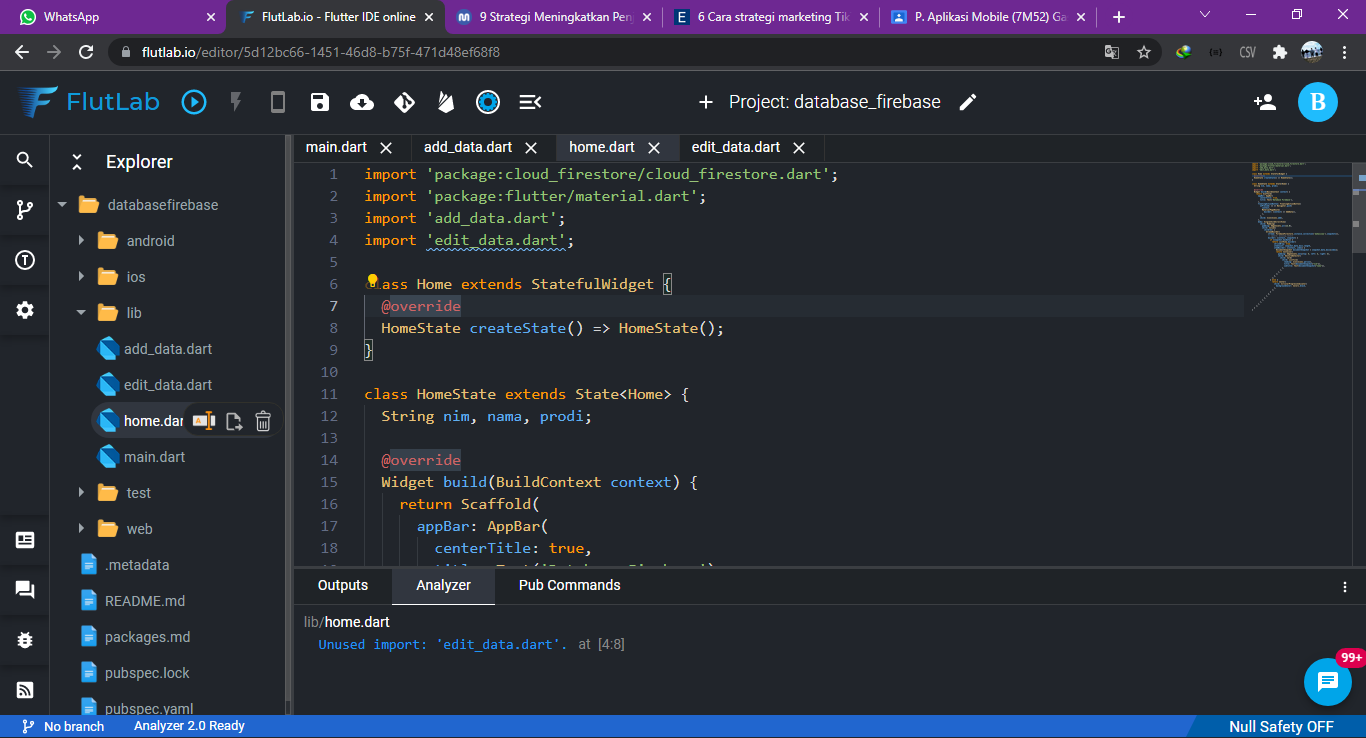
      ),

    );

  }

}

Tambahkan **import edit\_data.dart** pada halaman **home.dart**



Tambahkan juga atribut onTap pada widget GestureDetector

onTap: () => Navigator.push(

                            context,

                            MaterialPageRoute(

                              builder: (context) => EditData(

                                nim: documentSnapshot["nim"],

                                nama: documentSnapshot["nama"],

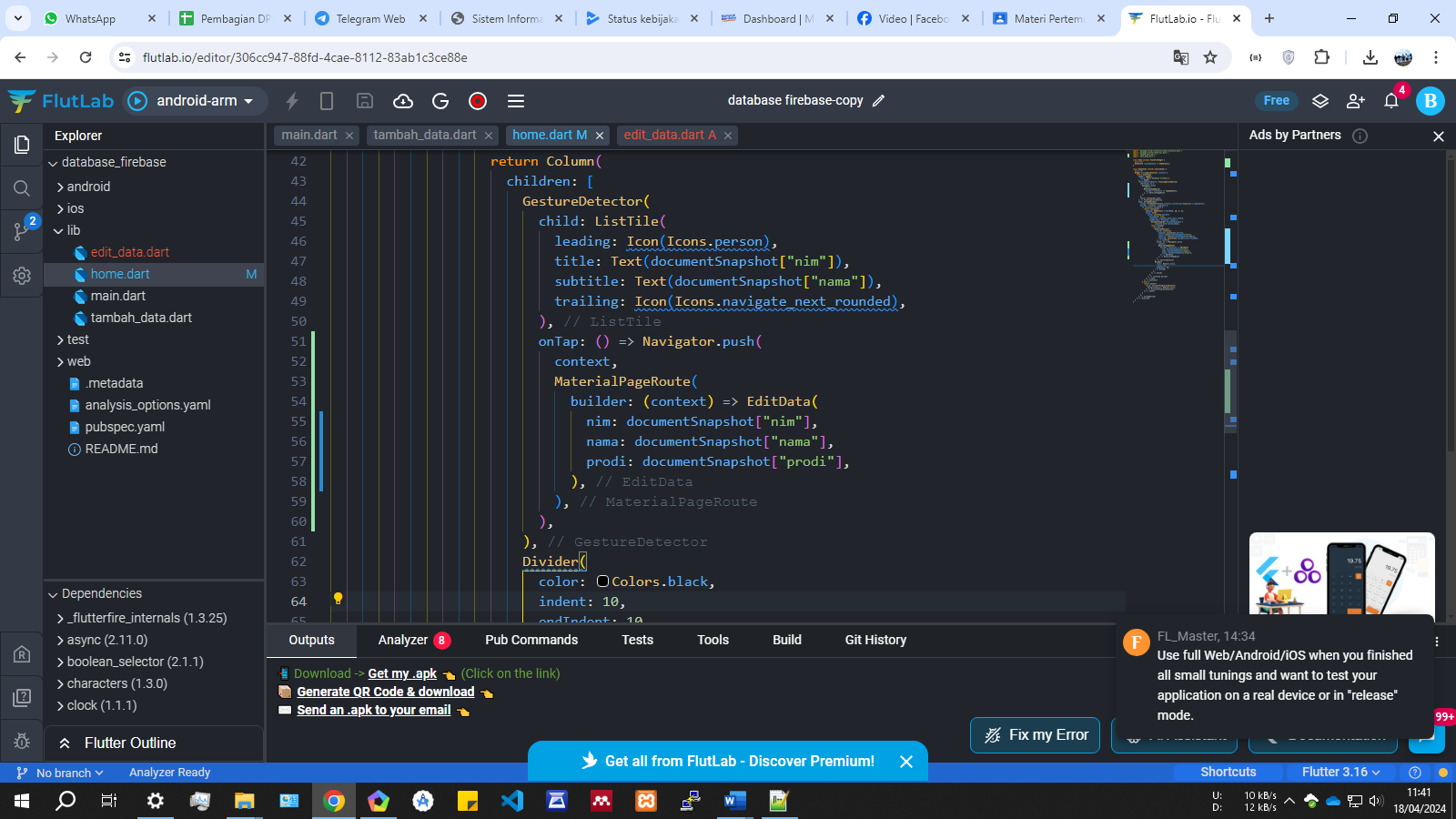
                                prodi: documentSnapshot["prodi"],

                              ),

                            ),

                          ),

Sehingga menjadi:



**Penjelasan Singkat:**

void editData() {

    DocumentReference documentReference =

        FirebaseFirestore.instance.collection('mahasiswa').doc(widget.nim);

    Map<String, dynamic> mhs = ({

      "nim": widget.nim,

      "nama": namaController.text,

      "prodi": prodiController.text

    });

    // update data to Firebase

    documentReference

        .update(mhs)

        .whenComplete(() => print('${widget.nim} updated'));

  }

Method editData digunakan untuk proses mengubah data pada database firestore. Method documentReference.update digunakan mengubah data sesuai dengan data yang dikirim.

void deleteData() {

    DocumentReference documentReference =

        FirebaseFirestore.instance.collection('mahasiswa').doc(widget.nim);

    // delete data from Firebase

    documentReference

        .delete()

        .whenComplete(() => print('${widget.nim} deleted'));

  }

Method deleteData digunakan untuk proses menghapus data pada database firestore. Method documentReference.delete digunakan untuk menghapus data sesuai dengan nim yang dipilih.

void konfirmasi() {

    AlertDialog alertDialog = AlertDialog(

      content: Text(

          "Apakah anda yakin akan menghapus data '${widget.nama}'"),

      actions: <Widget>[

        ElevatedButton(

          style: ElevatedButton.styleFrom(

            primary: Colors.red,

          ),

          child: Text(

            "OK DELETE!",

            style: TextStyle(color: Colors.black),

          ),

          onPressed: () {

            deleteData();

            Navigator.pushAndRemoveUntil(

              context,

              MaterialPageRoute(builder: (context) => Home()),

              (Route<dynamic> route) => false,

            );

          },

        ),

        ElevatedButton(

          style: ElevatedButton.styleFrom(

            primary: Colors.green,

          ),

          child: Text("CANCEL", style: TextStyle(color: Colors.black)),

          onPressed: () => Navigator.pop(context),

        ),

      ],

    );

    showDialog(

      context: context,

      builder: (BuildContext context) {

        return alertDialog;

      },

    );

  }

Method konfirmasi digunakan untuk menampilkan dialog untuk mengkonfirmasi user apakah yakin ingin menghapus data atau tidak. Dialog akan menampilkan 2 button yaitu OK dan Cancel. Jika memilih OK, maka akan memanggil method deleteData untuk menghapus data, kemudian kembali ke tampilan home.

@override

  void initState() {

    nimController = TextEditingController(text: widget.nim);

    namaController = TextEditingController(text: widget.nama);

    prodiController = TextEditingController(text: widget.prodi);

    super.initState();

  }

method initState akan dipanggil pertama kali saat class ini ditampilkan. Didalamnya berisi kode untuk mengisi TextFromField melalui controller. Data diambilkan dari list yang dikirmkan dari home melalui parameter.

**RANGKUMAN**

**[Diisi dengan rangkuman materi yang telah diuraikan]**

**TES FORMATIF**

**[Diisi dengan tes formatif sesuai dengan materi yang telah diuraikan]**

**DAFTAR PUSTAKA**

**[Diisi dengan daftar pustaka yang digunakan dalam menyusun modul pembelajaran]**