



# Buku Panduan

EasyFlutter

PUTRA PRIMA ARHANDI  
VIVIN AYU LESTARI  
ABDURRASYID MUHASIBI

## Daftar Isi

Abstrak.....	3
1. Software Requirement & Hardware Requirement.....	4
2. Proses Pembelajaran menggunakan aplikasi EasyFlutter.....	4
3. Menjalankan aplikasi EasyFlutter. ....	5
4. Source Code.....	9
4.1 Source Code View .....	9
4.2 Source Code Controller.....	15
5. Kesimpulan.....	22

## Daftar Gambar

Gambar 1: Gambar Aplikasi EasyFlutter.....	4
Gambar 2: Gambar Halaman Login.....	5
Gambar 3: Gambar Halaman Login Sebagai Mahasiswa.....	5
Gambar 4: Gambar Halaman Dashboard Mahasiswa.....	5
Gambar 5: Gambar Daftar Latihan Soal .....	6
Gambar 6: Gambar Latihan Soal .....	6
Gambar 7: Gambar Dialog Konfirmasi Kerjakan Latihan.....	6
Gambar 8: Gambar Kerjakan Latihan Soal.....	7
Gambar 9: Gambar Kerjakan Latihan Soal Peta Konsep.....	7
Gambar 10: Gambar Notifikasi Cek Jawaban Salah.....	8
Gambar 11: Gambar Notifikasi Cek Jawaban Benar .....	8
Gambar 12: Gambar Kerjakan Latihan Soal Peta Konsep Benar .....	8
Gambar 13: Dialog Latihan Soal Selesai .....	8
Gambar 14: Gambar Daftar Latihan Soal Selesai .....	8
Gambar 15: Gambar Bagian Utama Latihan Soal .....	9

## Abstrak

*Flutter* merupakan *framework* pembuatan aplikasi *cross-platform* buatan Google. Menurut data dari *statista.com* pada tahun 2019 hingga tahun 2021 popularitas *flutter* meningkat. *Flutter* menggunakan gaya penulisan deklaratif untuk membuat *layout*. Hal itu menjadikan *layout* pada *flutter* bersifat *immutable*, dan berupa *blueprint* ringan. Penelitian ini mengusulkan metode *construct on scaffold concept map* untuk membantu mahasiswa memahami konsep penyusunan *widget* pada *flutter layout*. *Construct on scaffold* akan memberikan pelajar kerangka dari peta konsep ahli yang tidak lengkap. Sebagian simpul dan relasi penghubung pada kerangka sudah dihilangkan, sehingga pelajar harus mengisi bagian yang hilang dengan beberapa pilihan jawaban yang tersedia untuk melengkapi peta konsep. Untuk membuktikan dampak dari penerapan metode ini, penelitian dilakukan dengan menggunakan rancangan eksperimen *pre-post-test group*. Mahasiswa akan melakukan *pre-test*, penggunaan aplikasi EasyFlutter, dan *post-test*. Hasil *pre-test* dan *post-test* yang didapat menunjukkan bahwa terdapat perbedaan yang signifikan antara nilai *pre-test* dan *post-test*. Penerapan metode *construct on scaffold* memiliki dampak positif yang signifikan terhadap nilai *post-test* mahasiswa terkait konsep penyusunan *widget* pada *flutter*.

**Kata kunci**—*EasyFlutter; Construct on Scaffold, Concept Map; Flutter; Flutter Layout;*

## 1. Software Requirement & Hardware Requirement.

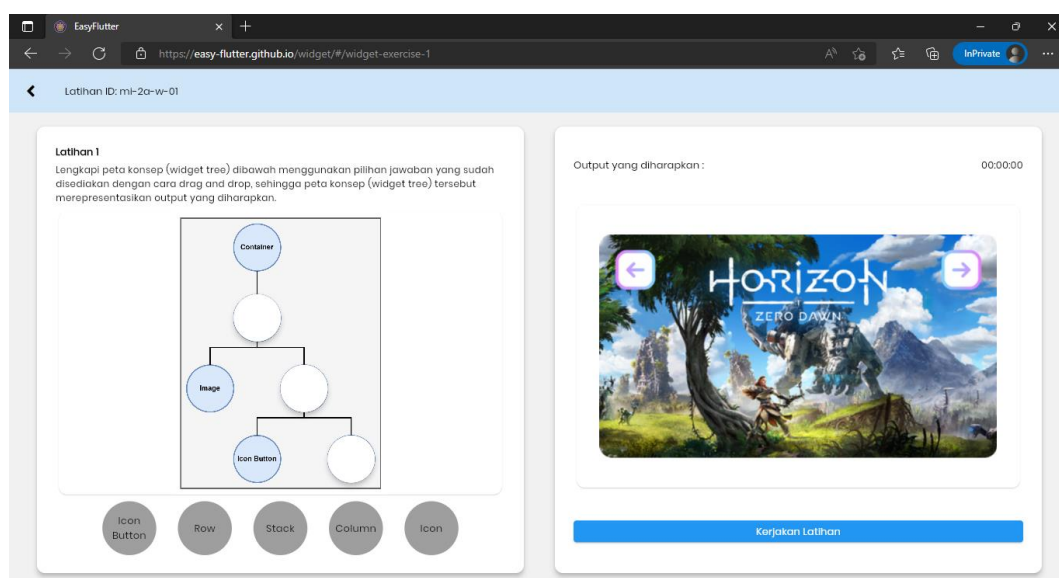
Software yang dibutuhkan untuk dapat menjalankan aplikasi ini adalah sebagai berikut:

1. Koneksi Internet yang stabil.
2. Laptop atau komputer dengan ukuran monitor 1366 x 768.
3. Chromium Based Browsers (Microsoft Edge, Google Chrome).

## 2. Proses Pembelajaran menggunakan aplikasi EasyFlutter.

Aplikasi EasyFlutter menggunakan metode *construct on scaffold concept map* untuk membantu mahasiswa dalam memahami konsep penyusunan *widget* pada *flutter* layout.

Berikut proses pembelajaran mahasiswa saat menggunakan aplikasi EasyFlutter:

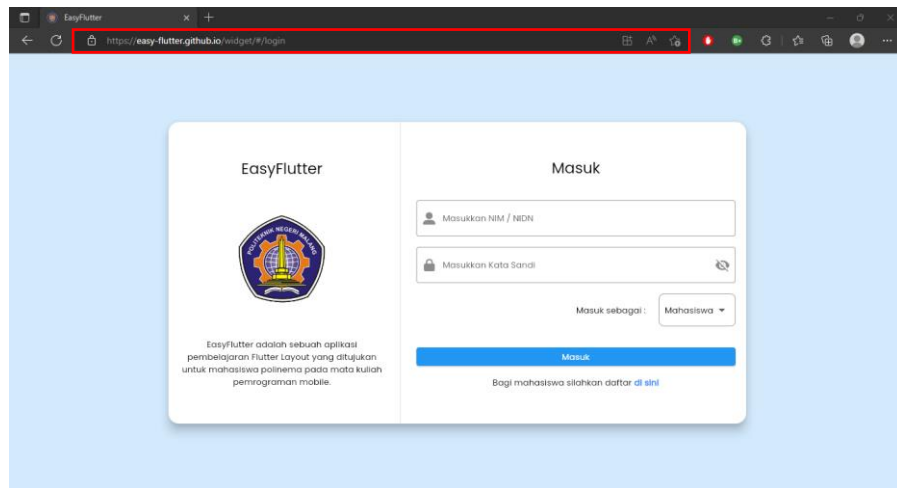


Gambar 1: Gambar Aplikasi EasyFlutter

Pada aplikasi EasyFlutter terdapat dua bagian, yaitu sebelah kiri dan kanan. Pada bagian sebelah kiri, mahasiswa dapat melihat nama latihan, deskripsi latihan, peta konsep dari ahli yang sudah dihilangkan beberapa bagian, dan pilihan jawaban untuk melengkapi peta konsep tersebut. Mahasiswa dapat melengkapi peta konsep dengan cara drag & drop pilihan jawaban ke bagian peta konsep yang kosong. Setiap mahasiswa melakukan drag & drop, jawaban mahasiswa akan disimpan di log jawaban (*firestore*). Pada bagian sebelah kanan, mahasiswa dapat melihat waktu yang menunjukkan seberapa lama mahasiswa mengerjakan latihan, gambar layout output yang diharapkan dari peta konsep, tombol reset, tombol cek jawaban, dan tombol kirim jawaban. Tombol reset digunakan mahasiswa untuk mengatur ulang peta konsep dan pilihan jawaban yang tersedia, tombol cek jawaban digunakan mahasiswa untuk memeriksa apakah peta konsep yang dilengkapi sudah benar, dan tombol kirim jawaban digunakan mahasiswa untuk menyelesaikan latihan.

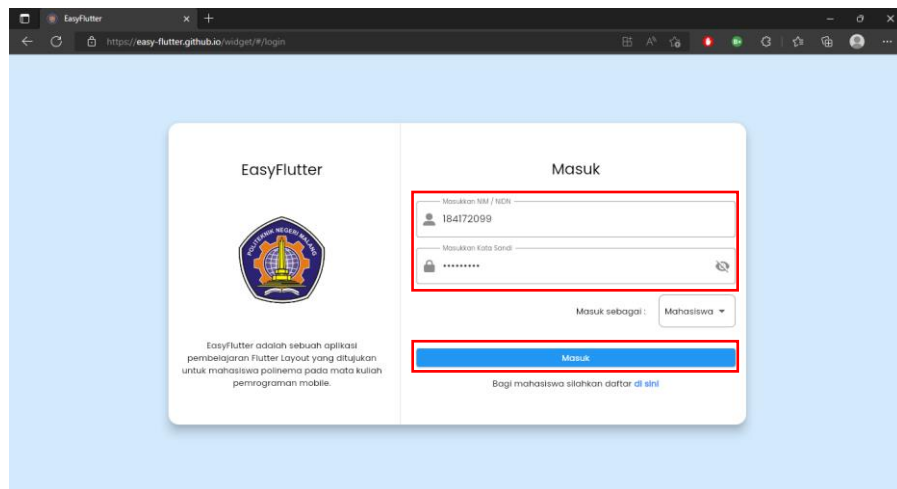
### 3. Menjalankan aplikasi EasyFlutter.

1. Buka website EasyFlutter dengan link: <https://easy-flutter.github.io/widget/>.

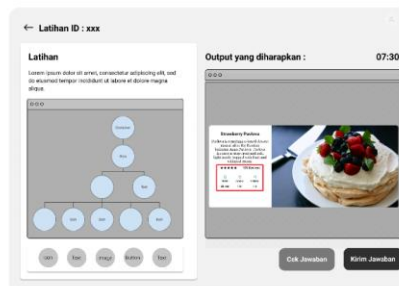
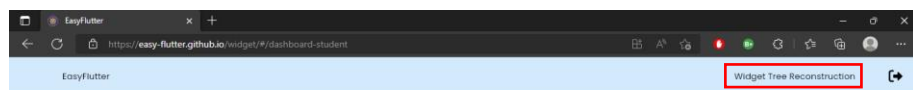


Gambar 2: Gambar Halaman Login

2. Login sebagai mahasiswa dengan username diisi NIM dan password diisi NIM.



Gambar 3: Gambar Halaman Login Sebagai Mahasiswa

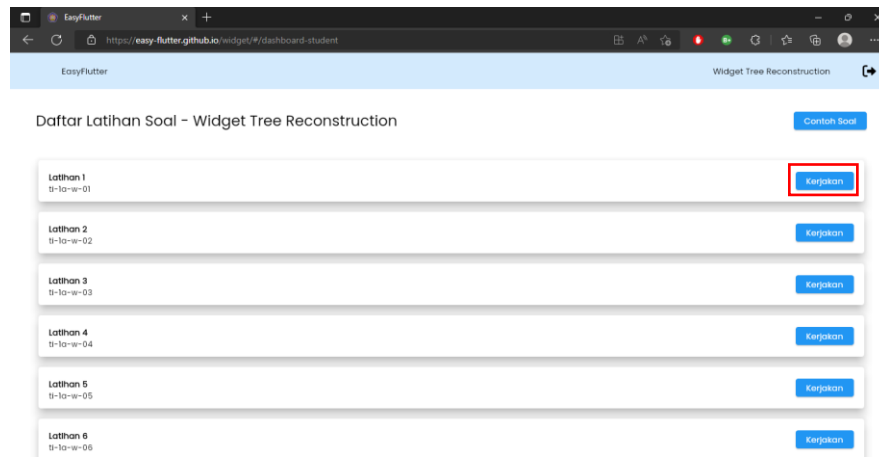


#### Apa itu Widget Tree Reconstruction ?

Widget Tree Reconstruction menerapkan metode Concept Map. Mahasiswa diberikan peta konsep dari dosen (ahli) yang sudah ditayangkan beberapa bagian, sehingga mahasiswa perlu melengkapi peta konsep tersebut dengan jawaban yang sudah disediakan.

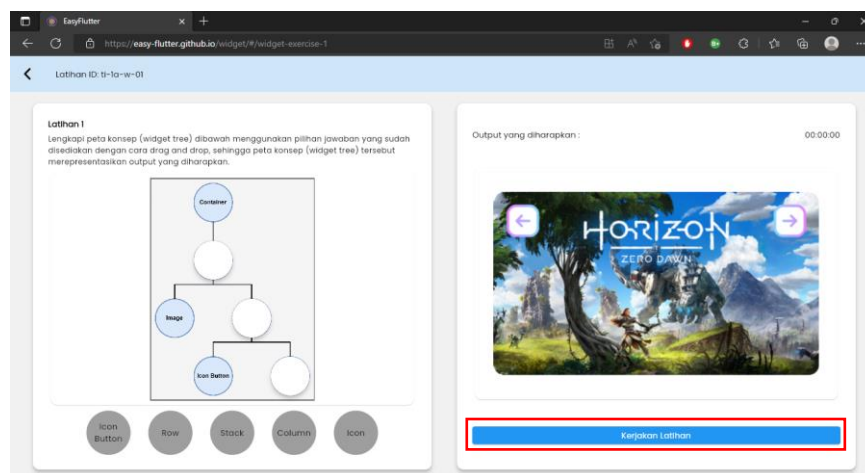
Gambar 4: Gambar Halaman Dashboard Mahasiswa

3. Buka fitur WidgetTreeReconstruction untuk melihat daftar latihan yang akan dikerjakan menggunakan pendekatan metode *construct on scaffold concept map*.



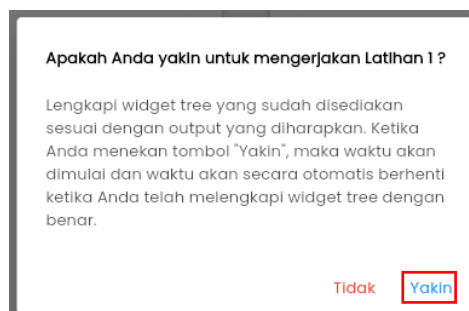
Gambar 5: Gambar Daftar Latihan Soal

4. Buka salah satu latihan soal, contoh: Latihan 1.



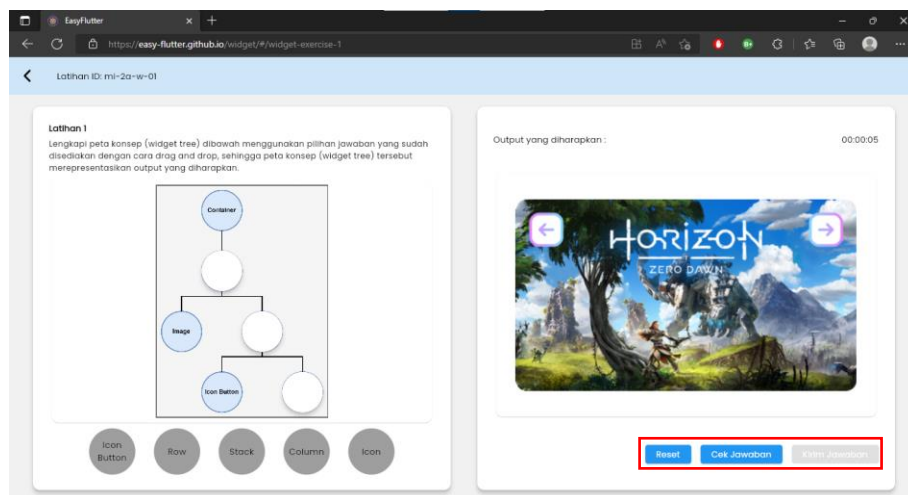
Gambar 6: Gambar Latihan Soal

5. Klik tombol “Kerjakan Latihan” untuk memulai latihan soal. Saat latihan dimulai, waktu akan berjalan dan mahasiswa dapat melakukan *drag & drop* pilihan jawaban ke bagian peta konsep yang masih kosong. Pada dialog konfirmasi mengerjakan latihan, jika mahasiswa menekan tombol “Yakin”, maka latihan soal akan dimulai.



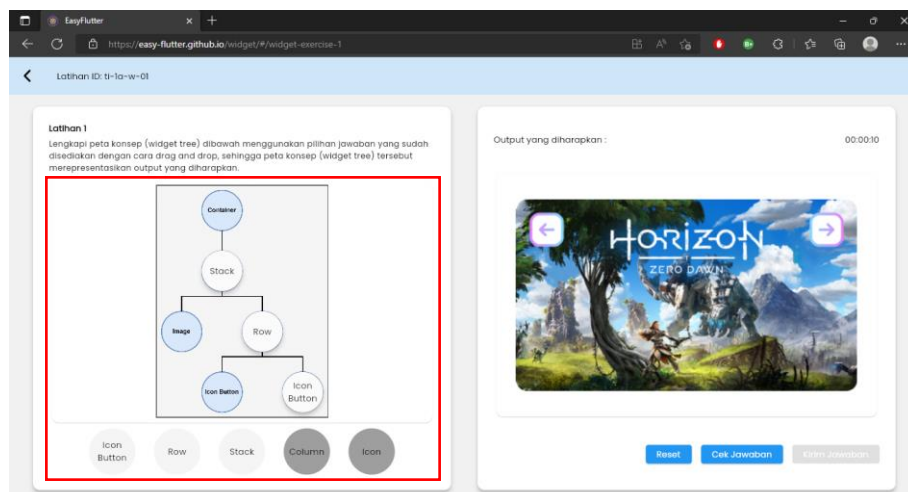
Gambar 7: Gambar Dialog Konfirmasi Kerjakan Latihan

- Setelah menekan tombol “Kerjakan Latihan” latihan soal akan dimulai dan muncul tombol “Reset”, “Cek Jawaban”, dan “Kirim Jawaban”. Tombol “Reset” digunakan mahasiswa untuk mengatur ulang peta konsep dan pilihan jawaban yang tersedia, tombol “Cek Jawaban” digunakan mahasiswa untuk memeriksa apakah peta konsep yang dilengkapi sudah benar, dan tombol “Kirim Jawaban” digunakan mahasiswa untuk menyelesaikan latihan.



Gambar 8: Gambar Kerjakan Latihan Soal

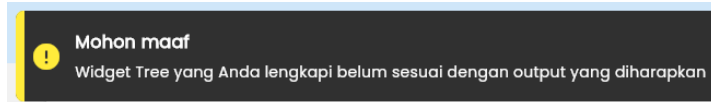
- Lengkapi peta konsep dari ahli yang kosong menggunakan pilihan jawaban yang sudah disediakan sesuai output yang diharapkan.



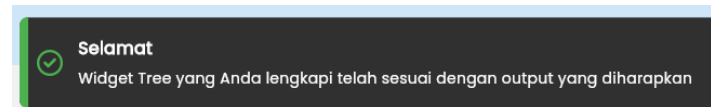
Gambar 9: Gambar Kerjakan Latihan Soal Peta Konsep

- Setelah yakin peta konsep telah dilengkapi dengan baik dan benar, tekan tombol “Cek Jawaban” untuk mengecek jawaban. Terdapat 2 jenis notifikasi pada saat cek jawaban, yaitu notifikasi jawaban salah, dan notifikasi jawaban benar. Pastikan muncul notifikasi jawaban benar “Widget Tree yang Anda lengkapi telah sesuai dengan output yang diharapkan”.



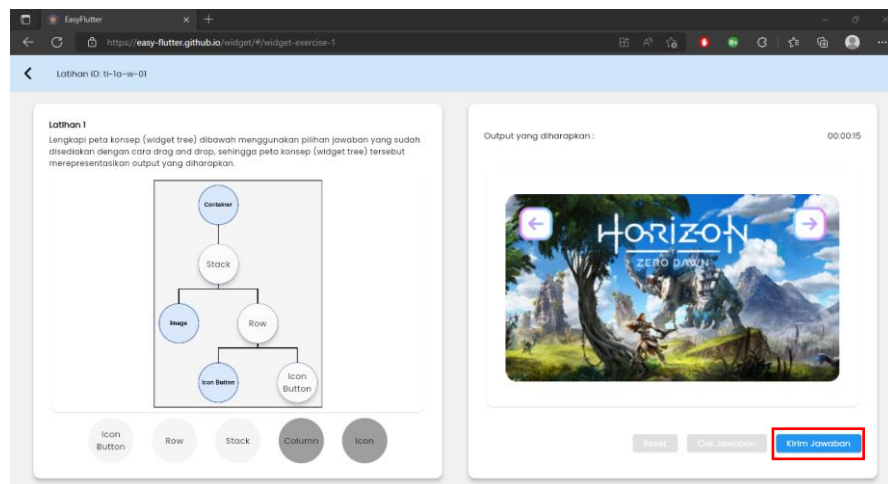


Gambar 10: Gambar Notifikasi Cek Jawaban Salah

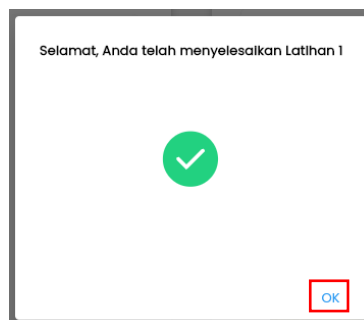


Gambar 11: Gambar Notifikasi Cek Jawaban Benar

9. Setelah proses cek jawaban selesai dan peta konsep dilengkapi dengan benar, tekan tombol “Kirim Jawaban” untuk menyelesaikan latihan.

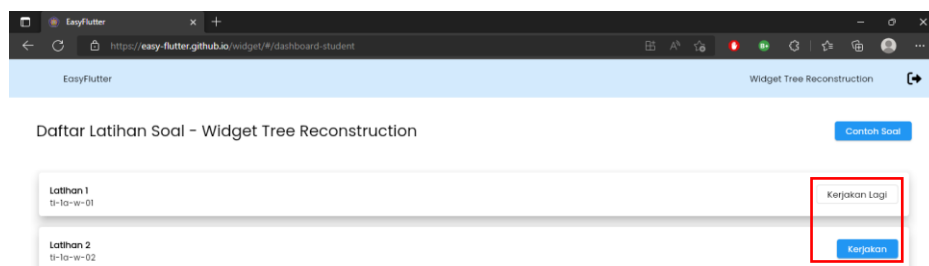


Gambar 12: Gambar Kerjakan Latihan Soal Peta Konsep Benar



Gambar 13: Dialog Latihan Soal Selesai

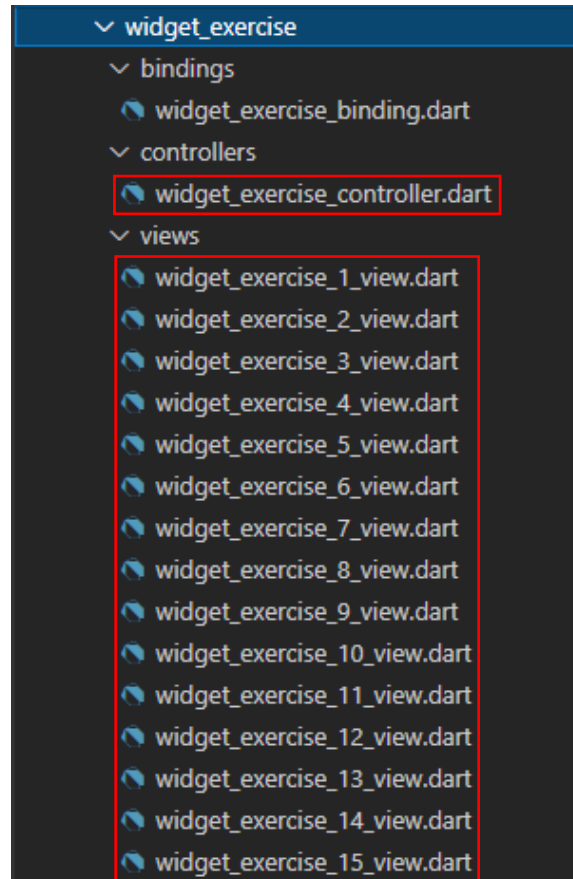
10. Setelah menyelesaikan latihan soal, pastikan tombol “Kerjakan” berubah menjadi “Kerjakan Lagi”. Hal itu menandakan latihan soal telah selesai dikerjakan.



Gambar 14: Gambar Daftar Latihan Soal Selesai

#### 4. Source Code.

Kode program aplikasi EasyFlutter dibuat menggunakan Visual Studio Code dengan bahasa pemrograman *Dart* dan menggunakan *framework Flutter*. Berikut *screenshot* bagian utama kode program yang digunakan pada latihan soal.



Gambar 15: Gambar Bagian Utama Latihan Soal

##### 4.1 Source Code View

Berikut ini isi dari class `widget_exercise_1_view.dart`.

```
import 'package:easyflutter/app/constants/color_constants.dart';
import 'package:easyflutter/app/utils/custom_text_helper.dart';
import
'package:easyflutter/app/views/widget/back_button_exercise_widget.dart';
import 'package:easyflutter/app/views/widget/drag_target_widget.dart';
import 'package:easyflutter/app/views/widget/draggable_widget.dart';
import 'package:easyflutter/app/views/widget/left_content_widget.dart';
import 'package:easyflutter/app/views/widget/right_content_widget.dart';
import 'package:flutter/material.dart';
import 'package:flutter_screenutil/flutter_screenutil.dart';
import 'package:get/get.dart';

import '../controllers/widget_exercise_controller.dart';

class WidgetExerciselView extends StatelessWidget {
  final controller = Get.put(WidgetExerciseController());

  @override
```

```

Widget build(BuildContext context) {
  return ScreenUtilInit(
    designSize: Size(1366, 768),
    builder: (context, child) {
      return Scaffold(
        appBar: AppBar(
          centerTitle: false,
          leading: BackButtonExercise(
            isStart: controller.isStart,
            exerciseName: controller.exerciseName,
          ),
          title: CustomTextHelper.textBody(
            context: context,
            text: "Latihan ID: ${controller.exerciseId}",
          ),
          backgroundColor: ColorConstants.kPrimaryColor,
          elevation: 0,
        ),
        body: Container(
          color: Colors.grey.withOpacity(0.1),
          child: Row(
            children: [
              LeftContentWidget(
                exerciseName: controller.exerciseName,
                exerciseDescription: controller.exerciseDescription,
                dragTargetContent: Center(
                  child: Stack(
                    children: [
                      Card(
                        shape: RoundedRectangleBorder(
                          borderRadius: BorderRadius.circular(12.r),
                        ),
                        child: Center(
                          child: SizedBox(
                            height: 420.h,
                            child: Image.asset(
                              "assets/images/widget/widget_tree1.png",
                            ),
                          ),
                        ),
                      ),
                    ],
                  ),
                ),
              Column(
                children: [
                  SizedBox(height: 127.5.r),
                  Row(
                    children: [
                      SizedBox(width: 264.h),
                      DragTargetWidget(
                        acceptAnswer: controller.acceptAnswer,
                        targetAnswer: controller.targetAnswer,
                        index: 0,
                        size: 70,
                      ),
                    ],
                  ),
                  SizedBox(height: 34.8.r),
                  Row(
                    children: [
                      SizedBox(width: 333.5.h),
                      DragTargetWidget(
                        acceptAnswer: controller.acceptAnswer,

```



Berikut Source Code untuk LeftContentWidget pada widget\_exercise\_1\_view.dart.

```
LeftContentWidget (
  exerciseName: controller.exerciseName,
  exerciseDescription: controller.exerciseDescription,
  dragTargetContent: Center(
    child: Stack(
      children: [
        Card(
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(12.r),
          ),
          child: Center(
            child: SizedBox(
              height: 420.h,
              child: Image.asset(
                "assets/images/widget/widget_tree1.png",
              ),
            ),
          ),
        ),
      ],
    ),
  ),
  Column(
    children: [
      SizedBox(height: 127.5.r),
      Row(
        children: [
          SizedBox(width: 264.h),
          DragTargetWidget(
            acceptAnswer: controller.acceptAnswer,
            targetAnswer: controller.targetAnswer,
            index: 0,
            size: 70,
          ),
        ],
      ),
      SizedBox(height: 34.8.r),
      Row(
        children: [
          SizedBox(width: 333.5.h),
          DragTargetWidget(
            acceptAnswer: controller.acceptAnswer,
            targetAnswer: controller.targetAnswer,
            index: 1,
            size: 70,
          ),
        ],
      ),
      SizedBox(height: 34.8.r),
      Row(
        children: [
          SizedBox(width: 403.3.h),
          DragTargetWidget(
            acceptAnswer: controller.acceptAnswer,
            targetAnswer: controller.targetAnswer,
            index: 2,
            size: 70,
          ),
        ],
      ),
    ],
  ),
),
```



```

    ],
  ),
),
);
}
}

```

Berikut Source Code untuk RightContentWidget pada widget\_exercise\_1\_view.dart.

```

RightContentWidget (
  stopWatchTimer: controller.stopWatchTimer,
  image: "assets/images/widget/latihan1.png",
  exerciseName: controller.exerciseName,
  isStart: controller.isStart,
  isAnswerTrue: controller.isAnswerTrue,
  startExercise: controller.startExercise,
  checkAnswer: controller.checkAnswer,
  dialogMessage: controller.dialogMessage,
  sendAnswer: controller.sendAnswer,
  reset: controller.reset,
),

```

Berikut source code untuk right\_content\_widget.dart

```

class RightContentWidget extends StatelessWidget {
  const RightContentWidget({
    required this.stopWatchTimer,
    required this.dialogMessage,
    required this.image,
    required this.exerciseName,
    required this.isStart,
    required this.isAnswerTrue,
    required this.startExercise,
    required this.checkAnswer,
    required this.sendAnswer,
    required this.reset,
    Key? key,
  }) : super(key: key);

  final Stopwatch stopWatchTimer;
  final String image;
  final String exerciseName;
  final String dialogMessage;
  final RxBool isStart;
  final RxBool isAnswerTrue;
  final Function startExercise;
  final Function checkAnswer;
  final Function sendAnswer;
  final Function reset;

  @override
  Widget build(BuildContext context) {
    return Expanded(
      flex: 1,
      child: Padding(
        padding: EdgeInsets.only(
          left: 16.w,
          top: 16.h,

```

```

        right: 32.w,
        bottom: 16.h,
    ),
    child: Card(
      elevation: 8.r,
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(12.r),
      ),
      child: Padding(
        padding: EdgeInsets.symmetric(horizontal:24.w,vertical:24.h),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.spaceAround,
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Row(
              mainAxisAlignment: MainAxisAlignment.spaceBetween,
              children: [
                CustomTextHelper.textBody(
                  context: context,
                  text: "Output yang diharapkan : ",
                ),
                StopwatchWidget(
                  stopwatchTimer: stopwatchTimer,
                ),
              ],
            ),
            Center(
              child: Card(
                shape: RoundedRectangleBorder(
                  borderRadius: BorderRadius.circular(12.r),
                ),
                child: SizedBox(
                  height: 420.h,
                  child: Image.asset(image),
                ),
              ),
            ),
            ButtonsWidget(
              exerciseName: exerciseName,
              dialogMessage: dialogMessage,
              isStart: isStart,
              isAnswerTrue: isAnswerTrue,
              startExercise: startExercise,
              checkAnswer: checkAnswer,
              sendAnswer: sendAnswer,
              reset: reset,
            ),
          ],
        ),
      ),
    );
  }
}

```

## 4.2 Source Code Controller

Berikut ini isi dari class `widget_exercise_controller.dart`.



```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:easyflutter/app/data/answer_widget_model.dart';
import 'package:easyflutter/app/data/log_model.dart';
import 'package:easyflutter/app/utils/check_answer_helper.dart';
import 'package:easyflutter/app/utils/send_answer_helper.dart';
import 'package:easyflutter/app/utils/snackbar_helper.dart';
import 'package:easyflutter/app/utils/storage_helper.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:stop_watch_timer/stop_watch_timer.dart';

class WidgetExerciseController extends GetxController {
  final StopwatchTimer stopWatchTimer = StopwatchTimer();

  final _storageHelper = Get.find<StorageHelper>();

  final CollectionReference logReference =
    FirebaseFirestore.instance.collection("log");

  final exerciseId = Get.arguments[0];
  final exerciseName = Get.arguments[1];
  final exerciseDescription =
    "Lengkapi peta konsep (widget tree) dibawah menggunakan pilihan jawaban yang sudah disediakan dengan cara drag and drop, sehingga peta konsep (widget tree) tersebut merepresentasikan output yang diharapkan.";
  final dialogMessage =
    "Lengkapi widget tree yang sudah disediakan sesuai dengan output yang diharapkan. Ketika Anda menekan tombol \"Yakin\", maka waktu akan dimulai dan waktu akan secara otomatis berhenti ketika Anda telah melengkapi widget tree dengan benar.";

  int step = 0;

  var isAnswerTrue = false.obs;

  var isStart = false.obs;
  var isOver = false;

  List<String> answerKey = Get.arguments[2][\"answerKey\"];
  List<AnswerWidgetModel> answerListArg = Get.arguments[2][\"answerList\"];
  List<AnswerWidgetModel> targetAnswerArg =
  Get.arguments[2][\"targetAnswer\"];

  var answerList = <AnswerWidgetModel>[].obs;
  var targetAnswer = <AnswerWidgetModel>[].obs;

  @override
  void onInit() {
    super.onInit();
    answerList = answerListArg.obs;
    targetAnswer = targetAnswerArg.obs;
    reset();
  }

  void acceptAnswer(
    AnswerWidgetModel answer, int indexTargetAnswer, BuildContext
context) {
    // check isStart
    if (isStart.value) {
      if (!isOver) {
        // disable answer

```

```

        final temp = answerList.removeAt(answer.index ?? -1);
        answerList.insert(
            answer.index ?? -1,
            AnswerWidgetModel(
                index: temp.index,
                content: temp.content,
                isUsed: true,
            ),
        );

        // replace answer
        if (targetAnswer[indexTargetAnswer].isUsed ?? true) {
            final tempAnswer = targetAnswer[indexTargetAnswer];
            answerList[tempAnswer.index ?? -1].isUsed = false;
        }

        // set targetAnswer
        targetAnswer[indexTargetAnswer].index = answer.index;
        targetAnswer[indexTargetAnswer].content = answer.content;
        targetAnswer[indexTargetAnswer].isUsed = true;

        createLog();
    }
} else {
    SnackbarHelper.showFlushbarInfo(
        "Informasi",
        'Pastikan Anda sudah menekan tombol "Kerjakan Latihan" untuk
melengkapi peta konsep',
    )..show(context);
}
}

void startExercise() {
    isStart.value = true;
    Get.back();
    startStopwatch();
}

void reset() {
    isAnswerTrue.value = false;

    for (int i = 0; i < answerList.length; i++) {
        final temp = answerList.removeAt(i);
        answerList.insert(
            i,
            AnswerWidgetModel(
                index: temp.index,
                content: temp.content,
                isUsed: false,
            ),
        );
    }

    for (int i = 0; i < targetAnswer.length; i++) {
        targetAnswer.removeAt(i);
        targetAnswer.insert(
            i,
            AnswerWidgetModel(
                index: -1,
                content: "",
                isUsed: false,
            ),
        );
    }
}

```

```

        ),
    );
}

void startStopwatch() {
    stopwatchTimer.onExecute.add(StopWatchExecute.start);
}

void stopStopWatch() {
    stopwatchTimer.onExecute.add(StopWatchExecute.stop);
}

void createLog() {
    step++;

    final log = LogModel(
        logId: "$exerciseId-${DateTime.now()}",
        studentId: _storageHelper.getIdUser(),
        studentName: _storageHelper.getNameUser(),
        exerciseId: exerciseId,
        step: step.toString(),
        time: StopwatchTimer.getDisplayTime(
            stopwatchTimer.rawTime.value,
            hours: true,
        ),
        answer: getTextAnswer().toString(),
        timeStamp: DateTime.now().toString(),
    );

    uploadLog(log);
}

void uploadLog(LogModel log) {
    logReference.add({
        "id_log": log.logId,
        "id_mahasiswa": log.studentId,
        "nama_mahasiswa": log.studentName,
        "id_latihan": log.exerciseId,
        "langkah": log.step,
        "waktu": log.time,
        "jawaban": log.answer,
        "time_stamp": log.timeStamp,
    });
}

bool checkAnswer() {
    if (CheckAnswerHelper.isEqual(getTextAnswer(), answerKey)) {
        isAnswerTrue.value = true;
        isOver = true;
        stopStopWatch();
        return true;
    } else {
        return false;
    }
}

List<String?> getTextAnswer() {
    return targetAnswer.map((e) => e.content).toList();
}

```

```

void sendAnswer() {
    SendAnswerHelper.updateExercise(exerciseId);
    SendAnswerHelper.updateStudent(exerciseId);
}

@override
void dispose() {
    stopwatchTimer.dispose();
    super.dispose();
}
}

```

Terdapat beberapa bagian utama pada code widget\_exercise\_controller.dart, yaitu memulai stopwatch, memberhentikan stopwatch, memulai latihan, tombol reset, tombol cek jawaban, tombol kirim jawaban, membuat log, dan upload log.

Berikut source code untuk memulai stopwatch. Fungsi ini dipanggil ketika memulai latihan.

```

void startStopwatch() {
    stopwatchTimer.onExecute.add(StopWatchExecute.start);
}

```

Berikut source code untuk memberhentikan stopwatch. Fungsi ini dipanggil ketika cek jawaban dan jawaban user benar.

```

void stopStopWatch() {
    stopwatchTimer.onExecute.add(StopWatchExecute.stop);
}

```

Berikut source code untuk memulai latihan.

```

void startExercise() {
    isStart.value = true;
    Get.back();
    startStopwatch();
}

```

Berikut source code untuk tombol reset. Tombol digunakan untuk mengatur ulang peta konsep dan pilihan jawaban yang tersedia.

```

void reset() {
    isAnswerTrue.value = false;

    for (int i = 0; i < answerList.length; i++) {
        final temp = answerList.removeAt(i);
        answerList.insert(
            i,
            AnswerWidgetModel(
                index: temp.index,
                content: temp.content,
                isUsed: false,
            ),
        );
    }
}

```

```

for (int i = 0; i < targetAnswer.length; i++) {
    targetAnswer.removeAt(i);
    targetAnswer.insert(
        i,
        AnswerWidgetModel(
            index: -1,
            content: "",
            isUsed: false,
        ),
    );
}
}

```

Berikut source code untuk tombol cek jawaban. Tombol digunakan untuk mengecek apakah peta konsep sudah dilengkapi dengan benar.

```

bool checkAnswer() {
    if (CheckAnswerHelper.isEqual(getTextAnswer(), answerKey)) {
        isAnswerTrue.value = true;
        isOver = true;
        stopStopWatch();
        return true;
    } else {
        return false;
    }
}

List<String?> getTextAnswer() {
    return targetAnswer.map((e) => e.content).toList();
}

```

Berikut source code untuk tombol kirim jawaban. Tombol digunakan untuk menandakan latihan soal berhasil diselesaikan.

```

void sendAnswer() {
    SendAnswerHelper.updateExercise(exerciseId);
    SendAnswerHelper.updateStudent(exerciseId);
}

```

Berikut source code untuk membuat log. Log dibuat setiap mahasiswa melakukan drag & drop pilihan jawaban ke bagian peta konsep yang masih kosong.

```

void createLog() {
    step++;
    final log = LogModel(
        logId: "$exerciseId-${DateTime.now()}",
        studentId: _storageHelper.getIdUser(),
        studentName: _storageHelper.getNameUser(),
        exerciseId: exerciseId,
        step: step.toString(),
        time: StopwatchTimer.getDisplayTime(
            stopwatchTimer.rawTime.value,
            hours: true,
        ),
        answer: getTextAnswer().toString(),
        timeStamp: DateTime.now().toString(),
    );
}

```

```
);  
uploadLog(log);  
}
```

Berikut source code untuk kirim log. Log dikirim ke database setelah log dibuat.

```
void uploadLog(LogModel log) {  
    logReference.add({  
        "id_log": log.logId,  
        "id_mahasiswa": log.studentId,  
        "nama_mahasiswa": log.studentName,  
        "id_latihan": log.exerciseId,  
        "langkah": log.step,  
        "waktu": log.time,  
        "jawaban": log.answer,  
        "time_stamp": log.timeStamp,  
    });  
}
```

Berikut source code untuk menerima jawaban. Fungsi ini akan dipanggil ketika user melakukan drag & drop pilihan jawaban ke bagian peta konsep yang masih kosong.

```
void acceptAnswer(  
    AnswerWidgetModel answer, int indexTargetAnswer, BuildContext  
    context) {  
    if (isStart.value) {  
        if (!isOver) {  
            // disable answer  
            final temp = answerList.removeAt(answer.index ?? -1);  
            answerList.insert(  
                answer.index ?? -1,  
                AnswerWidgetModel(  
                    index: temp.index,  
                    content: temp.content,  
                    isUsed: true,  
                ),  
            );  
            if (targetAnswer[indexTargetAnswer].isUsed ?? true) {  
                final tempAnswer = targetAnswer[indexTargetAnswer];  
                answerList[tempAnswer.index ?? -1].isUsed = false;  
            }  
            targetAnswer[indexTargetAnswer].index = answer.index;  
            targetAnswer[indexTargetAnswer].content = answer.content;  
            targetAnswer[indexTargetAnswer].isUsed = true;  
  
            createLog();  
        }  
    } else {  
        SnackbarHelper.showFlushbarInfo(  
            "Informasi",  
            'Pastikan Anda sudah menekan tombol "Kerjakan Latihan" untuk  
            melengkapi peta konsep',  
        )..show(context);  
    }  
}
```

## **5. Kesimpulan.**

Penerapan metode *construct on scaffold concept map* pada aplikasi EasyFlutter, memiliki dampak positif yang signifikan (nyata) terhadap nilai *post-test* mahasiswa pada mata kuliah pemrograman *mobile* dengan topik *flutter layout*. Aplikasi EasyFlutter juga membantu mahasiswa dalam memahami konsep penyusunan *widget* pada flutter layout.