



REPUBLIK INDONESIA  
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

## SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202297352, 30 November 2022

**Pencipta**  
Nama : **Untung Subagyo, S.Kom., M.Kom dan Faiz Ibnu Soleh**  
Alamat : **Soka Lor 002/002, Sruweng, Kec. Sruweng, Kebumen, JAWA TENGAH, 54362**  
Kewarganegaraan : **Indonesia**

**Pemegang Hak Cipta**  
Nama : **STMIK EIRahma Yogyakarta**  
Alamat : **Jl. Sisingamangaraja 76, Brontokusuman, Yogyakarta, DI YOGYAKARTA, 55153**  
Kewarganegaraan : **Indonesia**

Jenis Ciptaan : **Program Komputer**  
Judul Ciptaan : **APLIKASI ANDROID SIAKAD JOGJA**  
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : **17 Oktober 2022, di Yogyakarta**  
Jangka waktu perlindungan : **Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.**  
Nomor pencatatan : **000413096**

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.  
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n Menteri Hukum dan Hak Asasi Manusia  
Direktur Jenderal Kekayaan Intelektual  
u.b.  
Direktur Hak Cipta dan Desain Industri

Anggoro Dasananto  
NIP.196412081991031002

Disclaimer:  
Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

**Panduan Aplikasi**  
**SIAKAD (Sistem Informasi Akademik)**  
**STMIK ElRahma Yogyakarta**



**STMIK ELRAHMA Yogyakarta**

**2022**

## DAFTAR ISI

DAFTAR ISI.....	3
DAFTAR GAMBAR.....	4
PANDUAN PENGGUNAAN.....	5
Halaman Depan.....	5
Login ke Aplikasi.....	6
Mendaftar Akun.....	7
Halaman Depan Aplikasi (Dashboard) .....	8
Halaman Berita .....	9
Halaman Jadwal Kuliah .....	10
Halaman Biodata.....	11
Halaman KRS.....	12
Presensi.....	13
Endpoint.....	14
SOURCE CONTROLLER.....	15
Core Controller.....	15
Home Controller .....	16
Login Controller.....	18
StudyPlan Controller .....	20

## DAFTAR GAMBAR

Gambar 1 Tampilan Halaman Depan .....	5
Gambar 2. Halaman Login.....	6
Gambar 3. Tampilan Halaman Registrasi.....	7
Gambar 4. Halaman Dashboard.....	8
Gambar 5. Tampilan Halaman Berita.....	9
Gambar 6. Tampilan Halaman Jadwal Kuliah.....	10
Gambar 7. Tampilan Halaman Biodata .....	11
Gambar 8. Tampilan Halaman KRS .....	12
Gambar 9. Tampilan Halaman Presensi .....	13

## PANDUAN PENGGUNAAN

### Halaman Depan

Aplikasi dapat di download di Google Play Store

Jalankan aplikasi



Gambar 1 Tampilan Halaman Depan

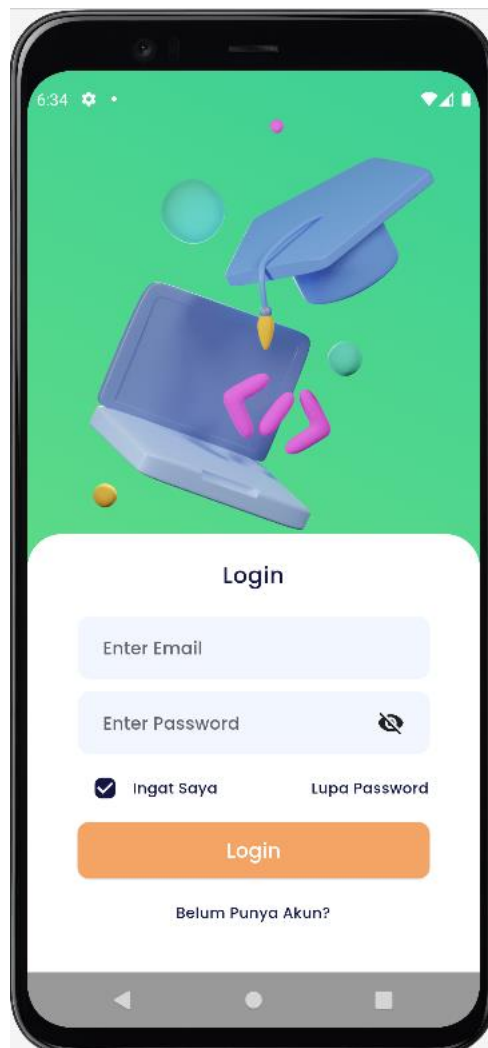
Klik tombol Mulai

## Login ke Aplikasi

Untuk Login:

- masukan Username / email yang sudah terdaftar
- masukan password
- Klik Login

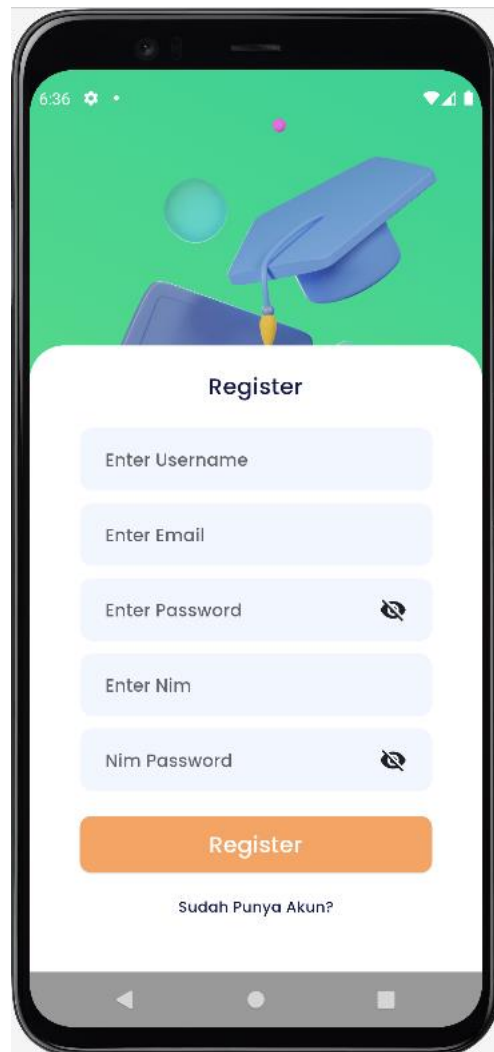
Apabila belum mempunyai akun, silahkan klik link **Belum Punya Akun?**



Gambar 2. Halaman Login

## Mendaftar Akun

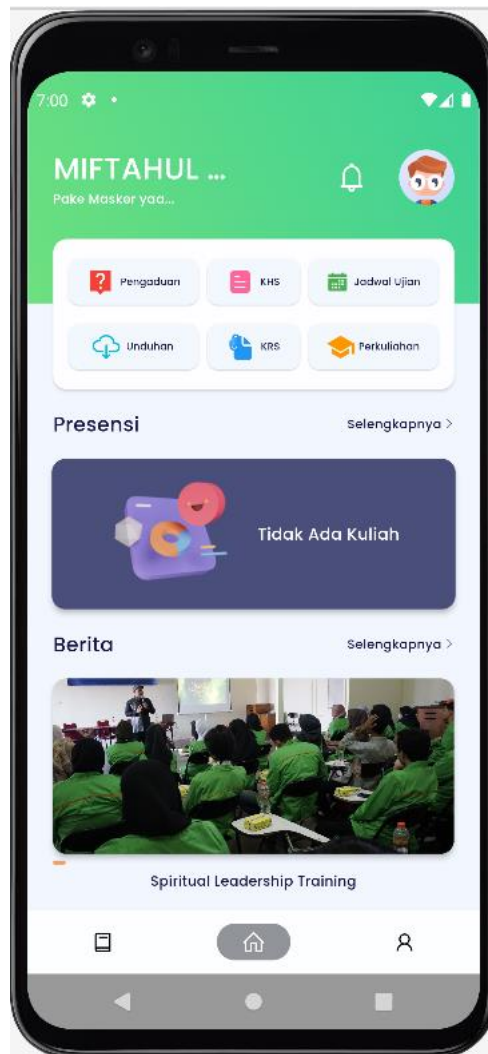
- Klik Belum Punya Akun?
- Masukan username pada kotak teks **Enter Username**
- Masukan alamat email pada kotak teks **Enter Email**
- Masukan password yang akan digunakan untuk login pada kotak teks **Enter Password**
- Masukan NIM pada kotak teks **Enter NIM**
- Masukan password yang digunakan untuk mengakses siacad web pada kotak teks **Nim Password**
- Klik tombol Register



Gambar 3. Tampilan Halaman Registrasi



## Halaman Depan Aplikasi (Dashboard)



Gambar 4. Halaman Dashboard

Pada halaman depan terdapat beberapa bagian:

1. Nama Pengguna
2. Menu Navigation
3. Kartu Presensi
4. Kartu Berita
5. Bottom Navigation

Nama Pengguna menampilkan nama dari pengguna yang login.

Menu Navigation menampilkan daftar menu yang tersedia pada aplikasi

Kartu Presensi menampilkan kuliah yang sedang berjalan, dan terdapat tombol untuk presensi ketika dosen sudah membuat BAP di SIAKAD.

Kartu Berita menampilkan daftar Pengumuman/Berita



Bottom Navigation berisi tiga tombol, yaitu Jadwal Hari Ini, Home, dan Biodata

Halaman Berita



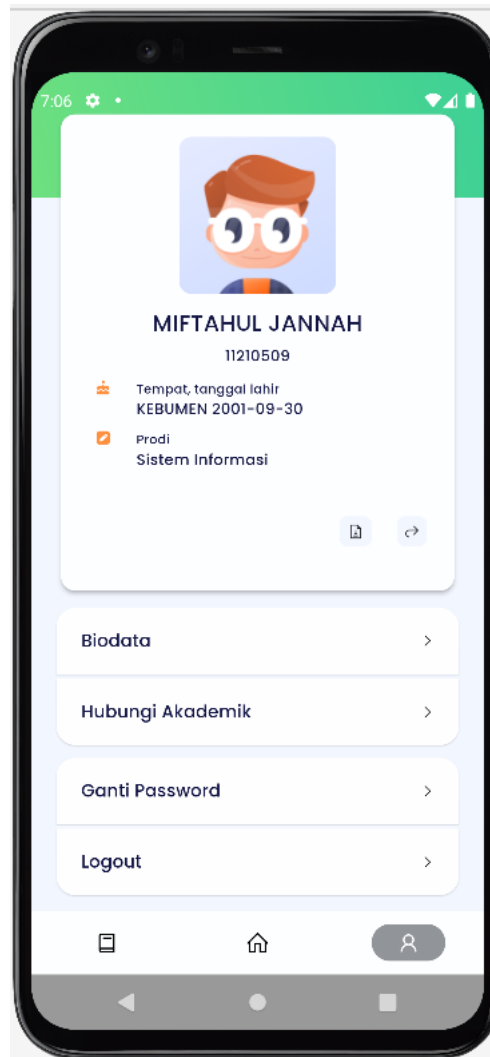
Gambar 5. Tampilan Halaman Berita

## Halaman Jadwal Kuliah



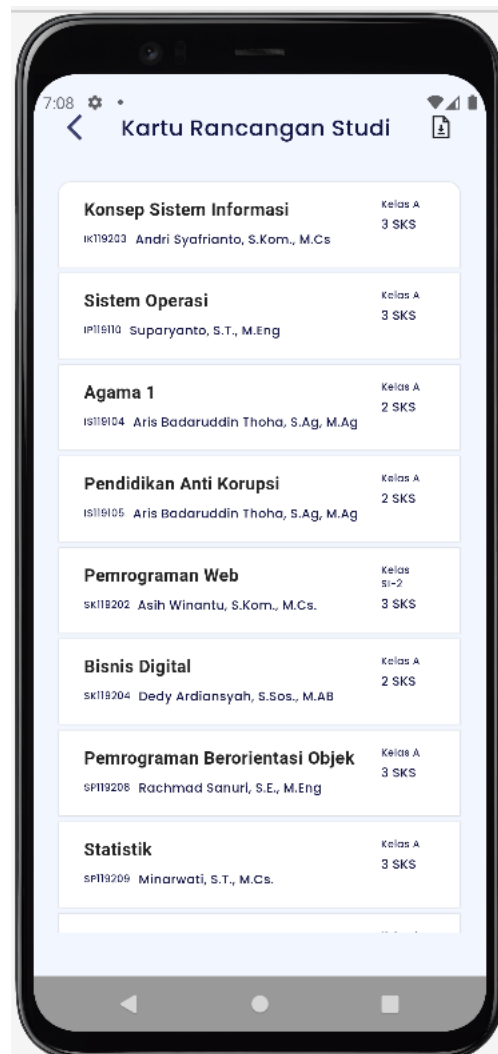
Gambar 6. Tampilan Halaman Jadwal Kuliah

## Halaman Biodata



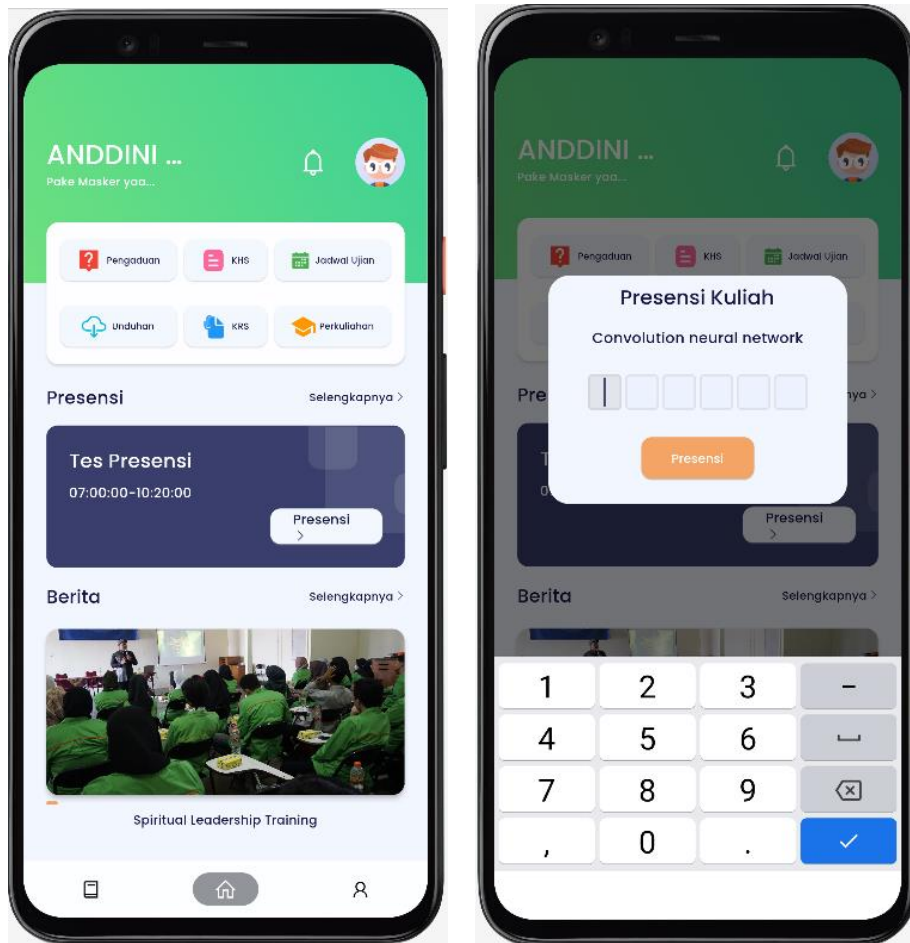
Gambar 7. Tampilan Halaman Biodata

## Halaman KRS



Gambar 8. Tampilan Halaman KRS

## Presensi



Gambar 9. Tampilan Halaman Presensi

## Endpoint

Aplikasi SIAKAD ini mengakses EndPoint berikut:

```
abstract class Endpoints {
    static const api = "https://api.stmikelrahma.ac.id/api/";
    static const login = 'login';
    static const mahasiswa = "get_mhs";
    static const period = "periode/get_periode_aktif";
    static const register = "register_with_nim";
    static const news = 'berita/get_news';
    static const logout = 'logout';
    static const isStudyPlan = 'periode/is_periode_krs';
    static const saveStudyPlan = 'krs/save_krs?kdkmk&kelas';
    static const deleteStudyPlan = 'krs/delete_krs';
    static const schedule = 'schedule/get_schedule';
    static const ampuan = 'ampuan/get_all_mk_ampuan';
    static const activeStudyPlan = 'krs/get_krs_aktif';
    static const present = 'schedule/simpanpresensi';
    static const scheduleToday = "schedule/get_schedule_today";
}
```

# SOURCE CONTROLLER

## Core Controller

```
import 'package:elrahma/src/config/config.dart';
import 'package:elrahma/src/data/data.dart';
import 'package:intl/intl.dart';

class CoreController extends GetxController
  with BiographyRepo, StudyPlanRepo, NewsRepo, ScheduleRepo,
  ScheduleRepo, ScheduleTodayRepo, CredentialRepo {
  RxList newsData = [].obs, schedule = [].obs;
  List scheduleToday = [];
  Rx<BioModel> bioData = BioModel().obs;
  RxList ampuanData = [].obs, studyPlan = [].obs;
  RxBool hide = false.obs;
  DateFormat date = DateFormat.EEEE('ID');

  // Get data From Api, Prevent Page loading When Open Home Page
  Future<void> getData(bool data) async {
    await getBio()
      .then((_) async => await getNews().then((_) async => await
getSchdule().then((_) => getActiveStudyPlan().then((_) =>
getScheduleToday()))));
    data ? FlutterNativeSplash.remove() : Get.back();
    Get.offAndToNamed(GoRoutes.dashboard);
  }

  void scheduleWeek() {
    // ('EEEE',).format(date, id);
  }

  //Get Bio Data
  Future<void> getBio() async {
    await apiBio().then((rampage) => rampage.status ? bioData.value =
rampage.object : null);
  }

  // Get Study Plan data
  Future<void> getStudyPlan() async {
    await apiIsStudyPlanRequest().then((rampage) => rampage.status ?
getAmpuan() : null);
  }

  //Get Active Studyplan
  Future<void> getActiveStudyPlan() async {
    await apiActiveStudyPlan().then((rampage) => rampage.status ?
studyPlan.value = rampage.data : null);
  }

  //Get Ampuan Data
  Future<void> getAmpuan() async {
    await apiAmpuan().then((rampage) => ampuanData.value = rampage.data);
  }

  //Get News Data
  Future<void> getNews() async {
    await apiNews().then((rampage) => newsData.value = rampage.data);
  }
}
```



```

Future<void> getScheduleToday() async {
  await apiScheduleToday().then((rampage) => scheduleToday =
rampage.data);
}

//Get Scedule Today
Future<void> getSchdule() async {
  await apiSchedule().then((rampage) =>
{logger.v(rampage.data[0].hari.toString()), schedule.value =
rampage.data});
}
}

```

## Home Controller

```

import 'package:elrahma/src/config/config.dart';
import 'package:elrahma/src/data/data.dart';
import 'package:elrahma/src/screen/screen.dart';
import 'package:intl/intl.dart';

class HomeController extends CoreController with CredentialRepo,
GetSingleTickerProviderStateMixin, PresentRepo, Universe {
  GlobalKey<ScrollSnapListState> sslKey = GlobalKey();
  RxInt focusedIndex = 0.obs;
  late final AnimationController animationController =
AnimationController(duration: const Duration(seconds: 1), vsync:
this)..repeat();
  BioModel mahasiswa = Get.find<CoreController>().bioData.value;
  List scheduleToday = Get.find<CoreController>().scheduleToday;
  List schedulesd = Get.find<CoreController>().schedule;
  List news = Get.find<CoreController>().newsData;
  var pageController = PageController(initialPage: 0).obs, newsController =
PageController(initialPage: 0).obs;
  RxBool present = true.obs, logoutStatus = false.obs;
  Rx<TabItem> currentTab = TabItem.present.obs;
  RxList schedulesx = [].obs;

  dynamic isFirstRouteInCurrentTab;

  // Call this method when controller get called in init
  @override
  void onReady() async {
    SystemChrome.setSystemUIOverlayStyle(const
SystemUiOverlayStyle(statusBarIconBrightness: Brightness.light));
    !await navigatorKeys[currentTab]!.currentState!.maybePop();
    super.onReady();
  }

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

  @override
  void onInit() {
    schedulesx.value = List.generate(6, (i) => List.filled(1, null,
growable: true));
    scheduleAwek();
    scheduleAwekGen();
    super.onInit();
  }

```

```

}

void onItemClickFocus(int index) {
    focusedIndex.value = index;
    super.refresh();
}

void scheduleAwek() {
    schedulesx.value = List.generate(6, (index) => [], growable: true);
    for (var element in schedulesd) {
        switch (element.hari) {
            case 1:
                schedulesx[0].add(element);
                break;
            case 2:
                schedulesx[1].add(element);
                break;
            case 3:
                schedulesx[2].add(element);
                break;
            case 4:
                schedulesx[3].add(element);
                break;
            case 5:
                schedulesx[4].add(element);
                break;
            case 6:
                schedulesx[5].add(element);
                break;
            default:
                break;
        }
    }
}

void scheduleAwekGen() {
    DateFormat _date = DateFormat.EEEE('ID');
    String dated = _date.format(DateTime.now());
    switch (dated) {
        case "Senin":
            focusedIndex.value = 0;
            update();
            break;
        case "Selasa":
            focusedIndex.value = 1;
            update();
            break;
        case "Rabu":
            focusedIndex.value = 2;
            update();
            break;
        case "Kamis":
            focusedIndex.value = 3;
            update();
            break;
        case "Jumat":
            focusedIndex.value = 4;
            update();
            break;
        case 'Sabtu':
            focusedIndex.value = 5;
    }
}

```

```

        update();
        break;
      default:
        break;
    }
  }
}

//logout Function. clear Hive and logout
Future<void> logout() async {
  logoutStatus.value = true;
  await apiLogout().then((rampage) async {
    rampage ? await box.clear().then((_) =>
{animationController.dispose(), Get.offAllNamed(GoRoutes.login)}) : null;
  });
}

Future<void> presentCheck(int data, String token) async {
  await apiPresent(data.toString(), token).then((value) => value.status ?
{Get.back(), snackBar("succes")} : snackBar("error"));
}
}

```

## Login Controller

```

import 'package:elrahma/src/config/config.dart';
import 'package:elrahma/src/screen/screen.dart';
import 'package:elrahma/src/data/data.dart';

enum SwithPage { signin, register, forgot }

class LoginController extends GetxController with CredentialRepo, Universe
{
  final core = Get.put(CoreController());
  Rx<SwithPage> switched = SwithPage.signin.obs;
  RxDouble keyboard = 0.0.obs, mover = 0.0.obs, bottom = 0.0.obs, animate =
0.0.obs;
  RxBool text = true.obs, obscure = true.obs, shake = false.obs, border =
false.obs, showed = false.obs, status = false.obs;
  Rx<TextEditingController> username = TextEditingController().obs, email =
TextEditingController().obs;
  Rx<TextEditingController> password = TextEditingController().obs, nim =
TextEditingController().obs;
  Rx<TextEditingController> nimpasssword = TextEditingController().obs;

  @override
  void onInit() {
    mover.value = aHeight(220.0);
    super.onInit();
  }

  //send api forgot request to server
  Future<void> forgotChecker() async {
    await apiRegister(username.value.text, email.value.text,
password.value.text, nim.value.text, nimpasssword.value.text).then(
(rampage) => rampage.status ? null : null,
  );
}

  //send api register request to server
  Future<void> registerChecker() async {
    status.value = true;

```

```

    await apiRegister(username.value.text, email.value.text,
password.value.text, nim.value.text, nimpassword.value.text).then((rampage)
{
    rampage.status
    ? core.getData(false)
    : {status.value = false, border.value = true, shake.value = true,
Timer(const Duration(milliseconds: 500), () => shake.value = false)};
});
}

//check login credential
Future<void> loginChecker() async {
    status.value = true;
    await apiLogin(email.value.text, password.value.text).then((rampage) {
        rampage.status
        ? core.getData(false)
        : {status.value = false, border.value = true, shake.value = true,
Timer(const Duration(milliseconds: 500), () => shake.value = false)};
    });
}

// clear all textfield in controller
void clearField() {
    username.value.clear();
    email.value.clear();
    password.value.clear();
    nim.value.clear();
    nimpassword.value.clear();
}

//when user completed register
void onRegisterComplete(context) {
    shake.value = shake.value ? false : false;
    border.value = border.value ? false : false;
    FocusScope.of(context).unfocus();
    animate.value = aHeight(340.0);
    bottom.value = aHeight(485.0);
}

//when user completed login
void onLoginComplete(context) {
    shake.value = shake.value ? false : false;
    border.value = border.value ? false : false;
    FocusScope.of(context).unfocus();
    animate.value = aHeight(345.0);
    bottom.value = aHeight(350.0);
    mover.value = aHeight(345.0);
}

// when user switch to login
void switchLogin() {
    clearField();
    switched.value = SwithPage.register;
    bottom.value = aHeight(500.0);
    shake.value = shake.value ? false : false;
    border.value = border.value ? false : false;
}

//when user switch to register
void switchRegister() {
    clearField();
}

```

```

        switched.value = SwithPage.signin;
        bottom.value = aHeight(350.0);
        shake.value = shake.value ? false : false;
        border.value = border.value ? false : false;
    }
}

```

## StudyPlan Controller

```

import 'package:elrahma/src/config/config.dart';
import 'package:elrahma/src/data/data.dart';
import 'package:elrahma/src/screen/screen.dart';

class StudyPlanController extends GetxController with StudyPlanRepo,
Universe {
    final ampuandata = Get.find<CoreController>().ampuanData;
    final activeStudyplan = Get.find<CoreController>().studyPlan;
    final controller = Get.find<CoreController>();
    RxMap<dynamic, dynamic> studyPlan = {}.obs, backup = {}.obs;
    Rx<TextEditingController> search = TextEditingController().obs;
    RxList confirmListed = [].obs;
    RxInt totalData = 0.obs;
    RxBool exit = false.obs;

    //Call this method on INit
    @override
    void onInit() {
        getDataController();
        super.onInit();
    }

    // Get Api Ampuan data from Api
    Future<void> getStudyPlan() async {
        await apiAmpuan().then((rampage) => rampage.status ?
saveData(rampage.data) : snackBar(rampage.message));
    }

    //save study plan data to controller
    Future<void> saveStudyPlan() async {
        confirmListed.asMap().forEach((key, value) async {
            await apiSaveStudyPlan(confirmListed[key]['kdkmk'],
confirmListed[key]['kelas'], confirmListed[key]['kode'])
                .then((value) => value.status ?
snackBar("${confirmListed[key]['nakmk']} : succes") :
snackBar(value.message));
        });
        await controller.apiActiveStudyPlan().then((value) =>
activeStudyplan.refresh());
        exit.value = true;
    }

    // Caled On init, first : check value in controller if null request to Api
server
    void getDataController() => ampuandata.isNotEmpty ? saveData(ampuandata)
: getStudyPlan();

    //Assign value to variable1 Backup and Studyplan
    void saveData(rampage) => {backup.value = studyPlan.value = rampage,
addCheckListed()};

    //Add check list data to work with checklisted items

```

```

    void addCheckListed() => studyPlan.forEach((key, value) =>
studyPlan[key].addAll({"checklist": false, "detail": false}));

// Expand container for info
    void containerExpand(int index) => {studyPlan[index]["detail"] =
studyPlan[index]["detail"] ? false : true, studyPlan.refresh()};

//Check the value of all sks from selected item
    void counterData() => studyPlan.forEach((key, value) =>
value["checklist"] ? totalData.value = totalData.value += value["sksmk"] as
int : null);

//Search filter
    void serchFilter() => backup.forEach(
    (key, value) =>
value["nakmk"].toString().isCaseInsensitiveContainsAny(search.value.text) ?
studyPlan[studyPlan.length] = backup[key] : null);

//Save the checklisted item to next screen
    void finalCheckListed(String data, classd) => studyPlan.forEach((key,
value) {
        studyPlan[key]["nakmk"] == data && studyPlan[key]["kelas"] ==
classd
        ? {studyPlan[key]["checklist"] = studyPlan[key]["checklist"] ?
false : true, counterData(), studyPlan.refresh(), totalData.refresh()}
        : null;
    });

//save Data to next Page
    void confirmData() => studyPlan.forEach((key, value) =>
value["checklist"] ? confirmListed.add(value) : null);
}

```





REPUBLIK INDONESIA  
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

## SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202297352, 30 November 2022

**Pencipta**

Nama : **Untung Subagyo, S.Kom., M.Kom dan Faiz Ibnu Soleh**  
Alamat : Soka Lor 002/002, Sruweng, Kec. Sruweng, Kebumen, JAWA  
TENGAH, 54362  
Kewarganegaraan : Indonesia

**Pemegang Hak Cipta**

Nama : **STMIK EIRahma Yogyakarta**  
Alamat : Jl. Sisingamangaraja 76, Brontokusuman, Yogyakarta, DI  
YOGYAKARTA, 55153  
Kewarganegaraan : Indonesia

Jenis Ciptaan : **Program Komputer**  
Judul Ciptaan : **APLIKASI ANDROID SIAKAD JOGJA**

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 17 Oktober 2022, di Yogyakarta

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000413096

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.  
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n Menteri Hukum dan Hak Asasi Manusia  
Direktur Jenderal Kekayaan Intelektual  
u.b.  
Direktur Hak Cipta dan Desain Industri

Anggoro Dasananto  
NIP.196412081991031002

**Disclaimer:**

Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.