

KODE PROGRAM

main.dart

```
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:sikaran_qris/SplashScreen/splashscreen.dart';

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  SystemChrome.setEnabledSystemUIMode(SystemUiMode.edgeToEdge);
  SystemChrome.setSystemUIOverlayStyle(
    const SystemUiOverlayStyle(
      statusBarColor: Colors.transparent,
      systemNavigationBarColor: Colors.transparent,
      statusBarIconBrightness: Brightness.dark,
    ),
  );
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
```

```
Widget build(BuildContext context) {  
  return const MaterialApp(  
    debugShowCheckedModeBanner: false,  
    home: SplashScreen(),  
  );  
}  
}
```

home.dart

```
import 'dart:async';  
import 'package:firebase_auth/firebase_auth.dart';  
import 'package:flutter/material.dart';  
import 'package:flutter/services.dart';  
import 'package:google_fonts/google_fonts.dart';  
import 'package:sikaran_qris/HomeDrawer/drawer.dart';  
import 'package:sikaran_qris/LocalData/data.dart';  
import 'package:sikaran_qris/NotificationListen/notification_listen.dart';  
import 'package:volume_controller/volume_controller.dart';  
import 'package:flutter_slide_drawer/flutter_slide_widget.dart';  
  
class VolumeSet extends StatefulWidget {  
  const VolumeSet({Key? key}) : super(key: key);  
  @override  
  State<VolumeSet> createState() => _VolumeSetState();  
}
```

```

class _VolumeSetState extends State<VolumeSet> {
  void getEmailCurrentUser() {
    final FirebaseAuth auth = FirebaseAuth.instance;
    final idEmailUser = auth.currentUser?.email;
    DataStatus.currentEmail = idEmailUser.toString();
  }

  void setVolume() {
    setState() {
      if (DataStatus.autoVolumeDevice == true) {
        VolumeController().showSystemUI = !DataStatus.uiVolumeDevice;
        VolumeController().setVolume(DataStatus.volumeDevice);
      }
    });
  }

  static const platformCek =
    MethodChannel('com.nadhila_bazhlina27.sikaran_qris/CekService');
  Future<void> cekService() async {
    try {
      bool status = await platformCek.invokeMethod('CekService');
      if (status == true) {
        DataStatus.izinService = true;
      } else {
        DataStatus.izinService = false;
      }
    }
  }
}

```

```
    } on PlatformException {  
        null;  
    }  
}
```

```
@override  
void initState() {  
    super.initState();  
    getEmailCurrentUser();  
    cekService();  
    setVolume();  
}
```

```
@override  
void dispose() {  
    super.dispose();  
    VolumeController().removeListener();  
}
```

```
@override  
Widget build(BuildContext context) {  
    return Scaffold(  
        backgroundColor: Colors.blue,  
        floatingActionButton: DataStatus.panduanAwal == false  
            ? null  
            : FloatingActionButton(  

```

```

        backgroundColor: Colors.indigo,
        child: const Icon(Icons.navigate_next),
        onPressed: () {
          DataStatus.panduanAwal = false;
          setState(() {});
        },
      ), // This
body: DataStatus.panduanAwal == false
? const HomeDrawer()
: Column(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Container(
      padding: const EdgeInsets.fromLTRB(10, 0, 10, 0),
      child: Center(
        child: Text(
          "Hai, Selamat Datang di",
          style: GoogleFonts.ubuntu(
            color: const Color.fromARGB(255, 255, 255, 255),
            fontSize: 30,
            fontWeight: FontWeight.w600,
          ),
        ),
      ),
    ),
  ],
),
Container(

```

```

padding: const EdgeInsets.fromLTRB(10, 0, 10, 0),
child: Center(
  child: Text(
    "SIKARAN QRIS!",
    style: GoogleFonts.ubuntu(
      color: const Color.fromARGB(255, 255, 255, 255),
      fontSize: 30,
      fontWeight: FontWeight.w600,
    ),
  ),
),
],
),
);
}
}

```

```

//===== Home
=====//

```

```

class HomeDrawer extends StatefulWidget {
  const HomeDrawer({Key? key}) : super(key: key);
  @override
  State<HomeDrawer> createState() => _HomeDrawerState();
}

```

```

class _HomeDrawerState extends State<HomeDrawer> {
  final GlobalKey<SliderDrawerWidgetState> drawerKey = GlobalKey();
  bool jalankan = false;

  @override
  void initState() {
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: WillPopScope(
        onWillPop: () async {
          final value = await showDialog<bool>(
            context: context,
            builder: (context) {
              return AlertDialog(
                title: const Text("Pemberitahuan"),
                content: const Text("Kamu Ingin Keluar Dari Aplikasi?"),
                actions: [
                  Row(
                    mainAxisAlignment: MainAxisAlignment.spaceAround,
                    children: [
                      ElevatedButton(
                        onPressed: () => Navigator.of(context).pop(false),

```

```

        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.blue,
          padding: const EdgeInsets.all(12),
        ),
        child: const Text("Tidak"),
      ),
      ElevatedButton(
        onPressed: () => Navigator.of(context).pop(true),
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.red,
          padding: const EdgeInsets.all(12),
        ),
        child: const Text("Iya"),
      )
    ],
  ),
]);

});

if (value != null) {
  return Future.value(value);
} else {
  return Future.value(false);
}
},
child: SliderDrawerWidget(
  key: drawerKey,

```



```
option: SliderDrawerOption(  
  backgroundImage: Image.asset("assets/images/latar.png"),  
  sliderEffectType: SliderEffectType.Rounded,  
  upDownScaleAmount: 50,  
  radiusAmount: 20,  
),  
drawer: const SingleChildScrollView(  
  child: IsiDrawer(),  
),  
body: GestureDetector(  
  onDoubleTap: () {  
    drawerKey.currentState!;  
  },  
  onTap: () {  
    drawerKey.currentState!.toggleDrawer();  
  },  
  onLongPress: () {  
    drawerKey.currentState!.toggleDrawer();  
  },  
  child: const NotificationListen(),  
),  
),  
),  
);  
}  
}
```

daftar.dart

```
import 'dart:async';
import 'package:awesome_dialog/awesome_dialog.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/services.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:lottie/lottie.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:sikaran_qris/HomeDrawer/home.dart';
import 'package:sikaran_qris/LocalData/data.dart';
import 'package:sikaran_qris/Login/login.dart';

class DaftarPage extends StatefulWidget {
  const DaftarPage({Key? key}) : super(key: key);

  @override
  State<StatefulWidget> createState() => _State();
}

class _State extends State<DaftarPage> {
  TextEditingController masukkanemailController = TextEditingController();
  TextEditingController masukkanpasswordController = TextEditingController();
  TextEditingController masukkanpassword2Controller = TextEditingController();
```

```
FirestoreFirestore firestore = FirestoreFirestore.instance;
```

```
final daftarakun = FirebaseAuth.instance;
```

```
bool jalankan = false;
```

```
bool _passwordVisible = false;
```

```
bool _passwordVisible2 = false;
```

```
bool tombol = false;
```

```
CollectionReference dataStream = firestore  
    .collection('users')  
    .doc(masukkanemailController.text.trim())  
    .collection('qris');  
dataStream.doc(masukkanemailController.text.trim()).set({  
    'app': "Awal",  
    'date': "Awal",  
    'money': "Awal",  
    'appTes': "Tes",  
    'dateTes': "Tes",  
    'moneyTes': "Tes",  
});  
}
```

```
static const platformRun =
```

```
    MethodChannel('com.nadhila_bazhlina27.sikaran_qris/RunService');
```

```
Future<void> runService() async {
```

```
    try {
```

```
    await platformRun.invokeMethod('RunService');
  } on PlatformException {
    null;
  }
}
});
}
```

```
@override
void initState() {
  super.initState();
  DataStatus.loadingSikaran = false;
}
```

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

```
@override
Widget build(BuildContext context) {
  DataStatus.orientationData = MediaQuery.of(context)
    .orientation
    .toString()
    .replaceAll("Orientation.portrait", "Portrait")
    .replaceAll("Orientation.landscape", "Landscape");
```



```

        ),
        child: const Text("Tidak"),
    ),
    ElevatedButton(
        onPressed: () => Navigator.of(context).pop(true),
        style: ElevatedButton.styleFrom(
            backgroundColor: Colors.red,
            padding: const EdgeInsets.all(12),
        ),
        child: const Text("Iya"),
    )
],
)
]);
});
if (value != null) {
    return Future.value(value);
} else {
    return Future.value(false);
}
},
child: AnimatedContainer(
    duration: const Duration(seconds: 6),
    curve: Curves.fastOutSlowIn,
    decoration: BoxDecoration(
        gradient: LinearGradient(

```

```

        colors: const [
Color.fromARGB(255, 202, 202, 225),
Color.fromARGB(255, 234, 234, 239),
Color.fromARGB(255, 242, 241, 252),
],
        end: jalankan
            ? Alignment.bottomLeft
            : AlignmentDirectional.topCenter,
        begin:
            jalankan ? Alignment.centerRight : Alignment.centerLeft)),
child: Center(
    child: ListView(
        physics: const BouncingScrollPhysics(),
        shrinkWrap: true,
        padding: const EdgeInsets.all(30),
        children: [
            // ===== //
            Padding(
                padding: const EdgeInsets.only(bottom: 30),
                child: Row(
                    mainAxisAlignment: MainAxisAlignment.center,
                    children: [
                        Lottie.asset("assets/lottie/loading_loop_qr.json",
                            width: 50, height: 40),
                        Padding(
                            padding: const EdgeInsets.only(bottom: 4),

```

```

child: Text(
  "SIKARAN QRIS",
  style: GoogleFonts.poppins(
    color: const Color.fromRGBO(49, 48, 146, 1.0),
    fontSize: 30,
    fontWeight: FontWeight.w600,
  ),
),
),
],
),
),
// ===== //

DataStatus.loadingSikaran == true
? Container()
: Column(
  children: [
    Container(
      padding: const EdgeInsets.only(bottom: 10, top: 10),
      child: TextField(
        style: GoogleFonts.ubuntu(
          color: Colors.grey[800],
          fontSize: 17,
          fontWeight: FontWeight.w600,
        ),

```



```
controller: masukanemailController,
decoration: InputDecoration(
  isDense: true,
  hintText: 'Masukkan Email',
  hintStyle: GoogleFonts.ubuntu(
    color: Colors.grey[800],
    fontSize: 15,
    fontWeight: FontWeight.w600,
  ),
  enabledBorder: OutlineInputBorder(
    borderSide: const BorderSide(
      color: Color.fromARGB(255, 123, 123, 123),
      width: 2,
    ),
    borderRadius: BorderRadius.circular(8),
  ),
  focusedBorder: OutlineInputBorder(
    borderSide: const BorderSide(
      color: Color(0xFFDBE2E7),
      width: 2,
    ),
    borderRadius: BorderRadius.circular(8),
  ),
  filled: true,
  fillColor: Colors.grey[200],
  suffixIcon: Icon(Icons.email,
```

```
        color: Colors.grey[800])),
    ),
),
// ===== //
```

```
Container(
  padding: const EdgeInsets.only(bottom: 10, top: 10),
  child: TextField(
    style: GoogleFonts.ubuntu(
      color: Colors.grey[800],
      fontSize: 17,
      fontWeight: FontWeight.w600,
    ),
    obscureText: !_passwordVisible,
    controller: masukanpasswordController,
    decoration: InputDecoration(
      isDense: true,
      hintText: 'Masukkan Password',
      hintStyle: GoogleFonts.ubuntu(
        color: Colors.grey[800],
        fontSize: 15,
        fontWeight: FontWeight.w600,
      ),
      enabledBorder: OutlineInputBorder(
        borderRadius: BorderRadius.circular(8),
        borderSide: const BorderSide(
```

```
        color: Color.fromARGB(255, 123, 123, 123),
        width: 2,
    ),
),
focusedBorder: OutlineInputBorder(
    borderRadius: BorderRadius.circular(8),
    borderSide: const BorderSide(
        color: Color(0xFFDBE2E7),
        width: 2,
    ),
),
filled: true,
fillColor: Colors.grey[200],
suffixIcon: IconButton(
    icon: Icon(
        // Based on passwordVisible state choose the icon
        _passwordVisible
        ? Icons.visibility
        : Icons.visibility_off,
        color: Colors.grey[800],
    ),
    onPressed: () {
        setState(() {
            _passwordVisible = !_passwordVisible;
        });
    },
),
```



```

borderSide: const BorderSide(
  color: Color.fromARGB(255, 123, 123, 123),
  width: 2,
),
borderRadius: BorderRadius.circular(8),
),
focusedBorder: OutlineInputBorder(
  borderSide: const BorderSide(
    color: Color(0xFFDBE2E7),
    width: 2,
  ),
  borderRadius: BorderRadius.circular(8),
),
filled: true,
fillColor: Colors.grey[200],
suffixIcon: IconButton(
  icon: Icon(
    _passwordVisible2
      ? Icons.visibility
      : Icons.visibility_off,
    color: Colors.grey[800],
  ),
  onPressed: () {
    // Update the state i.e. toggle the state of passwordVisible

variable
    setState() {

```

```
        _passwordVisible2 = !_passwordVisible2;
    });
  },
),
),
),
),
// ===== //
```

```
Container(
  alignment: Alignment.centerRight,
  padding: const EdgeInsets.only(
    top: 10, bottom: 10, right: 0),
  child: Material(
    color: Colors.transparent,
    child: InkWell(
      borderRadius: BorderRadius.circular(4),
      splashColor: Colors.blueGrey,
      onTap: () {
        login();
      },
      child: Text(
        ' Sudah memiliki akun ? ',
        style: GoogleFonts.ubuntu(
          color:
            const Color.fromRGBO(49, 48, 146, 1.0),
```

```

        fontWeight: FontWeight.w700,
        fontSize: 15,
    ),
),
),
),
),
),
// ===== //

const SizedBox(height: 25),

Container(
  margin: const EdgeInsets.only(top: 15, bottom: 15),
  height: 45,
  decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(10),
    gradient: const LinearGradient(
      colors: [
        Color.fromARGB(255, 66, 66, 150),
        Color.fromARGB(255, 105, 105, 174),
      ],
      begin: Alignment.topLeft,
      end: Alignment.bottomRight),
  child: Material(
    color: Colors.transparent,
    child: InkWell(

```



```
),  
);  
}
```

```
void dialogDaftar(pesanTitle) {  
    AwesomeDialog(  
        context: context,  
        autoHide: const Duration(seconds: 6),  
        dialogType: DialogType.info,  
        animType: AnimType.scale,  
        title: pesanTitle,  
    ).show();  
}
```

```
void dialogError(pesanTitle) {  
    AwesomeDialog(  
        context: context,  
        autoHide: const Duration(seconds: 6),  
        dialogType: DialogType.error,  
        animType: AnimType.scale,  
        title: pesanTitle,  
    ).show();  
}
```

```
void login() {  
    DataStatus.loadingSikaran = true;
```

```
FocusScope.of(context).requestFocus(FocusNode());
```

```
Timer(const Duration(milliseconds: 1500), () {  
  Navigator.of(context)  
    .pop(MaterialPageRoute(builder: (context) => const LoginPage()));  
  Navigator.of(context)  
    .push(MaterialPageRoute(builder: (context) => const LoginPage()));  
  DataStatus.loadingSikaran = false;  
});  
}
```

```
void daftarAkun(String email, String password) async {
```

```
  tombol = true;
```

```
  toastShort("Loading", Colors.black38);
```

```
  await daftarakun
```

```
    .createUserWithEmailAndPassword(email: email, password: password)
```

```
    .then((uid) => {
```

```
      DataStatus.loadingSikaran = true,
```

```
      DataStatus.panduanAwal = true,
```

```
      DataStatus.signOut = false,
```

```
      saveDaftar(email),
```

```
      saveUserID(),
```

```
      setState(() {
```

```
        jalankan = !jalankan;
```

```
        DataStatus.imageVideo = null;
```

```

        saveDaftar(email);
        runService();
    }),
    }),
    })

// ignore: body_might_complete_normally_catch_error
.catchError((err) {
DataStatus.loadingSikaran = false;
String error = err
    .toString()
    .replaceAll("[firebase_auth/unknown] Given String is empty or null",
        "Form Tidak Boleh Kosong!")
    .replaceAll(
        "[firebase_auth/invalid-email] The email address is badly formatted.",
        "Format Email Yang Anda Masukkan Salah!")
    .replaceAll(
        "[firebase_auth/weak-password] Password should be at least 6 characters",
        "Password Minimal Harus 6 Karakter!")
    .replaceAll(
        "[firebase_auth/wrong-password] The password is invalid or the user does
not have a password.",
        "Email atau Password Yang Anda Masukkan Salah!")
    .replaceAll(
        "[firebase_auth/user-not-found] There is no user record corresponding to
this identifier. The user may have been deleted.",
        "Email Yang Anda Masukkan Tidak Terdaftar!")
    .replaceAll(

```

```
        "[firebase_auth/email-already-in-use] The email address is already in use by
another account.",
        "Akun Tersebut Sudah Terdaftar!")
        .replaceAll(
        "[firebase_auth/network-request-failed] A network error (such as timeout,
interrupted connection or unreachable host) has occurred.",
        "Koneksi Internet Anda Terputus, Coba Lagi!");
        dialogError(error);
        tombol = false;
    });
}
}
```

login.dart

```
import 'dart:async';
import 'package:awesome_dialog/awesome_dialog.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/services.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:lottie/lottie.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:sikaran_qris/HomeDrawer/home.dart';
import 'package:sikaran_qris/LocalData/data.dart';
```

```
import 'package:sikaran_qris/Login/daftar.dart';
import 'package:sikaran_qris/Login/lupa_password.dart';

class LoginPage extends StatefulWidget {
  const LoginPage({Key? key}) : super(key: key);

  @override
  State<StatefulWidget> createState() => _State();
}

class _State extends State<LoginPage> {
  TextEditingController emailController = TextEditingController();
  TextEditingController passwordController = TextEditingController();
  final loginakun = FirebaseAuth.instance;
  bool jalankan = false;
  bool _passwordVisible = false;
  bool tombol = false;

  @override
  void initState() {
    super.initState();
    DataStatus.login = true;
    DataStatus.loadingSikaran = false;
  }

  @override
```

```

Widget build(BuildContext context) {
  DataStatus.orientationData = MediaQuery.of(context)
    .orientation
    .toString()
    .replaceAll("Orientation.portrait", "Portrait")
    .replaceAll("Orientation.landscape", "Landscape");

  if (DataStatus.orientationData == "Portrait") {
    SystemChrome.setEnabledSystemUIMode(SystemUiMode.edgeToEdge);
    SystemChrome.setSystemUIOverlayStyle(
      const SystemUiOverlayStyle(
        statusBarColor: Colors.transparent,
        systemNavigationBarColor: Colors.transparent,
        statusBarIconBrightness: Brightness.dark,
      ),
    );
  }

  if (DataStatus.orientationData == "Landscape") {
    SystemChrome.setEnabledSystemUIMode(SystemUiMode.immersiveSticky);
  }

  return Scaffold(
    body: WillPopScope(
      onWillPop: () async {
        final value = await showDialog<bool>(

```

```
context: context,
builder: (context) {
  return AlertDialog(
    title: const Text("Pemberitahuan"),
    content: const Text("Kamu Ingin Keluar Dari Aplikasi?"),
    actions: [
      Row(
        mainAxisAlignment: MainAxisAlignment.spaceAround,
        children: [
          ElevatedButton(
            onPressed: () => Navigator.of(context).pop(false),
            style: ElevatedButton.styleFrom(
              backgroundColor: Colors.blue,
              padding: const EdgeInsets.all(12),
            ),
            child: const Text("Tidak"),
          ),
          ElevatedButton(
            style: ElevatedButton.styleFrom(
              backgroundColor: Colors.red,
              padding: const EdgeInsets.all(12),
            ),
            child: const Text("Iya"),
          )
        ],
      ),
    ],
  )
}
```

```

        ]);
    });
    if (value != null) {
        return Future.value(value);
    } else {
        return Future.value(false);
    }
},
child: AnimatedContainer(
    duration: const Duration(seconds: 6),
    curve: Curves.fastOutSlowIn,
    decoration: BoxDecoration(
        gradient: LinearGradient(
            colors: const [
                Color.fromARGB(255, 202, 202, 225),
                Color.fromARGB(255, 234, 234, 239),
                Color.fromARGB(255, 242, 241, 252),
            ],
            end: jalankan
                ? Alignment.bottomLeft
                : AlignmentDirectional.topCenter,
            begin:
                jalankan ? Alignment.centerRight : Alignment.centerLeft)),
    child: Center(
        child: ListView(
            physics: const BouncingScrollPhysics(),

```



```

shrinkWrap: true,
padding: const EdgeInsets.all(30),
children: [
  // ===== //
  Padding(
    padding: const EdgeInsets.only(bottom: 30),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        Lottie.asset("assets/lottie/loading_loop_qr.json",
          width: 50, height: 40),
        Padding(
          padding: const EdgeInsets.only(bottom: 4),
          child: Text(
            "SIKARAN QRIS",
            style: GoogleFonts.poppins(
              color: const Color.fromRGBO(49, 48, 146, 1.0),
              fontSize: 30,
              fontWeight: FontWeight.w600,
            ),
          ),
        ],
      ),
    ),
  // ===== //

```

```
DataStatus.loadingSikaran == true
? Container()
: Column(
  children: [
    Container(
      padding: const EdgeInsets.only(bottom: 10, top: 10),
      child: TextField(
        style: GoogleFonts.ubuntu(
          color: Colors.grey[800],
          fontSize: 17,
          fontWeight: FontWeight.w600,
        ),
        controller: emailController,
        decoration: InputDecoration(
          isDense: true,
          hintText: 'Masukkan Email | Staf',
          hintStyle: GoogleFonts.ubuntu(
            color: Colors.grey[800],
            fontSize: 15,
            fontWeight: FontWeight.w600,
          ),
          enabledBorder: OutlineInputBorder(
            borderSide: const BorderSide(
              color: Color.fromARGB(255, 123, 123, 123),
              width: 2,
            ),
          ),
```

```
        borderRadius: BorderRadius.circular(8),
    ),
    focusedBorder: OutlineInputBorder(
      borderSide: const BorderSide(
        color: Color(0xFFDBE2E7),
        width: 2,
      ),
      borderRadius: BorderRadius.circular(8),
    ),
    filled: true,
    fillColor: Colors.grey[200],
    suffixIcon: Icon(Icons.email,
      color: Colors.grey[800])),
  ),
),
```

```
Container(
  padding: const EdgeInsets.only(bottom: 10, top: 10),
  child: TextField(
    style: GoogleFonts.ubuntu(
      color: Colors.grey[800],
      fontSize: 17,
      fontWeight: FontWeight.w600,
    ),
    obscureText: !_passwordVisible,
    controller: passwordController,
```

```
decoration: InputDecoration(
  isDense: true,
  hintText: 'Masukkan Password',
  hintStyle: GoogleFonts.ubuntu(
    color: Colors.grey[800],
    fontSize: 15,
    fontWeight: FontWeight.w600,
  ),
  enabledBorder: OutlineInputBorder(
    borderRadius: BorderRadius.circular(8),
    borderSide: const BorderSide(
      color: Color.fromARGB(255, 123, 123, 123),
      width: 2,
    ),
  ),
  focusedBorder: OutlineInputBorder(
    borderRadius: BorderRadius.circular(8),
    borderSide: const BorderSide(
      color: Color(0xFFDBE2E7),
      width: 2,
    ),
  ),
  filled: true,
  fillColor: Colors.grey[200],
  suffixIcon: IconButton(
    icon: Icon(
```

```

// Based on passwordVisible state choose the icon
_passwordVisible
  ? Icons.visibility
  : Icons.visibility_off,
  color: Colors.grey[800],
),
onPressed: () {
  setState(() {
    _passwordVisible = !_passwordVisible;
  });
},
),
//====//
),
),
),
// =====//

```

```

Container(
  padding: const EdgeInsets.only(
    top: 10, bottom: 10, left: 0, right: 0),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      Material(
        color: Colors.transparent,

```



```
child: Text(  
  textAlign: TextAlign.end,  
  ' Belum memiliki akun? ',  
  style: GoogleFonts.ubuntu(  
    color: const Color.fromRGBO(  
      49, 48, 146, 1.0),  
    fontWeight: FontWeight.w700,  
    fontSize: 15,  
  ),  
),  
),  
),  
),  
],  
),  
),  
// ===== //
```

```
const SizedBox(height: 25),  
// ===== //
```

```
Container(  
  margin: const EdgeInsets.only(top: 15, bottom: 15),  
  height: 45,  
  decoration: BoxDecoration(  
    borderRadius: BorderRadius.circular(10),  
    gradient: const LinearGradient(  

```

```
        colors: [
            Color.fromARGB(255, 66, 66, 150),
            Color.fromARGB(255, 105, 105, 174),
        ],
        begin: Alignment.topLeft,
        end: Alignment.bottomRight)),
child: Material(
    color: Colors.transparent,
    child: InkWell(
        borderRadius: BorderRadius.circular(10),
        onTap: () {
            if (tombol == false) {
                masuk();
            }
        },
        child: Center(
            child: Text(
                'MASUK',
                style: GoogleFonts.ubuntu(
                    color: Colors.white,
                    letterSpacing: 1,
                    fontWeight: FontWeight.w700,
                    fontSize: 16,
                ),
            ),
        ),
    ),
```



```
        ),
    ),
),
// ===== //
],
),
// ===== //
],
),
),
),
),
),
);
}
```

```
void toastShort(message, Color colorBox) {
  Fluttertoast.showToast(
    msg: message,
    toastLength: Toast.LENGTH_SHORT,
    gravity: ToastGravity.BOTTOM,
    backgroundColor: colorBox,
  );
}
```

```
void dialogMasuk(pesanTitle) {
  AwesomeDialog(
```

```
    context: context,
    autoHide: const Duration(seconds: 6),
    dialogType: DialogType.info,
    animType: AnimType.scale,
    title: pesanTitle,
  ).show();
}
```

```
void dialogError(pesanTitle) {
  AwesomeDialog(
    context: context,
    autoHide: const Duration(seconds: 6),
    dialogType: DialogType.error,
    animType: AnimType.scale,
    title: pesanTitle,
  ).show();
}
```

```
void masuk() {
  FocusScope.of(context).requestFocus(FocusNode());
  if (emailController.text.trim() == "") {
    dialogMasuk("Masukkan Email Anda!");
  } else {
    if (emailController.text.trim() == ":" ||
        (emailController.text.lastIndexOf(":") + 1) ==
            emailController.text.length) {
```



```

static const platformOff =
    MethodChannel('com.nadhila_bazhlina27.sikaran_qris/OffService');
Future<void> offService() async {
    try {
        await platformOff.invokeMethod('OffService');
    } on PlatformException {
        null;
    }
}

void saveUserID() {
    FirebaseFirestore firestore = FirebaseFirestore.instance;
    CollectionReference loginUser = firestore.collection('users');
    loginUser.doc(emailController.text.trim()).set({
        'idUser': emailController.text.trim(),
        'idDevice': DataStatus.deviceID,
    });
}

void saveLogin(bool notification) async {
    final spDir = await SharedPreferences.getInstance();
    setState(() {
        spDir.setBool('notificationORfirebase', notification);
        spDir.setBool('enabledNotificationListen', true);
        spDir.setBool('statusgetQRISname', true);
        spDir.setString('getQRISname', "com.shopeepay.merchant.id");
    });
}

```

```

spDir.setString('emailAkunStaf', emailController.text.toString().trim());
DataStatus.akunStaf = false;
DataStatus.emailAkunStaf = emailController.text.toString().trim();
DataStatus.currentEmailAkunStaf = "";
DataStatus.notificationORfirebase = false;
});
}

```

```

void login(String email, String password) async {
  tombol = true;
  toastShort("Loading", Colors.black38);
  await loginakun
    .signInWithEmailAndPassword(email: email, password: password)
    .then((uid) => {
      saveLogin(false),
      DataStatus.loadingSikaran = true,
      DataStatus.notificationORfirebase = false,
      DataStatus.notificationORfirebase = false,
      saveUserID(),
      setState() {
        DataStatus.signOut = false;
        DataStatus.signOut = false;
        jalankan = !jalankan;
        DataStatus.imageVideo = null;
        runService();
      },
    },

```

```

Timer(const Duration(milliseconds: 4000), () {
  toastShort("Login Berhasil", Colors.green);
  Navigator.of(context).pushReplacement(
    MaterialPageRoute(builder: (context) => const VolumeSet()));
  DataStatus.loadingSikaran = false;
  tombol = false;
}),
})

// ignore: body_might_complete_normally_catch_error
.catchError((err) {
DataStatus.loadingSikaran = false;
String error = err
.toString()
.replaceAll("[firebase_auth/unknown] Given String is empty or null",
  "Form Tidak Boleh Kosong!")
.replaceAll(
  "[firebase_auth/invalid-email] The email address is badly formatted.",
  "Format Email Yang Anda Masukkan Salah!")
.replaceAll(
  "[firebase_auth/wrong-password] The password is invalid or the user does
not have a password.",
  "Email atau Password Yang Anda Masukkan Salah!")
.replaceAll(
  "[firebase_auth/user-not-found] There is no user record corresponding to
this identifier. The user may have been deleted.",
  "Email Yang Anda Masukkan Tidak Terdaftar!")
.replaceAll(

```

```
        "[firebase_auth/network-request-failed] A network error (such as timeout,
interrupted connection or unreachable host) has occurred.",
```

```
        "Koneksi Internet Anda Terputus, Coba Lagi!");
```

```
        dialogError(error);
```

```
        tombol = false;
```

```
    });
```

```
}
```

```
void saveData(bool notification, String emailAkunStaf,
```

```
    String passwordAkunStaf, bool akunStaf, String currentStaf) async {
```

```
    final spDir = await SharedPreferences.getInstance();
```

```
    setState(() {
```

```
        spDir.setBool('notificationORfirebase', notification);
```

```
        spDir.setString('emailAkunStaf', emailAkunStaf);
```

```
        spDir.setString('currentEmailAkunStaf', currentStaf);
```

```
        spDir.setString('passwordAkunStaf', passwordAkunStaf);
```

```
        spDir.setBool('akunStaf', akunStaf);
```

```
        DataStatus.akunStaf = akunStaf;
```

```
        DataStatus.emailAkunStaf = emailAkunStaf;
```

```
        DataStatus.passwordAkunStaf = passwordAkunStaf;
```

```
        DataStatus.currentEmailAkunStaf = currentStaf;
```

```
        DataStatus.notificationORfirebase = notification;
```

```
    });
```

```
}
```

```
void cekAkunStaf(String email) async {
```

```

tombol = true;
int tanda = email.lastIndexOf(":");
String akun = email.substring(tanda + 1, email.length);
FirebaseFirestore firestore = FirebaseFirestore.instance;
DocumentReference akunStaf =
    firestore.collection('users').doc(akun).collection('staf').doc(email);
dynamic data;
toastShort("Loading", Colors.black38);

if (DataStatus.akunStaf == false) {
    akunStaf.get().then(
        (value) {
            data = value.data();

            if (data?['password'] == passwordController.text.trim()) {
                DataStatus.loadingSikaran = true;
                DataStatus.signOut = false;
                if (emailController.text != "") {
                    DataStatus.currentEmail = akun;
                }
                saveData(true, emailController.text.toString().trim(),
                    passwordController.text.trim(), true, akun);
                setState() {
                    jalankan = !jalankan;
                    DataStatus.imageVideo = null;
                    DataStatus.akunStaf = true;
                }
            }
        }
    );
}

```



```

        saveData(true, emailController.text.toString().trim(),
            passwordController.text.trim(), true, akun);
        offService();
    });
    Timer(const Duration(milliseconds: 4000), () {
        toastShort("Login Staf Berhasil", Colors.green);
        Navigator.of(context).pushReplacement(
            MaterialPageRoute(builder: (context) => const VolumeSet()));
        DataStatus.loadingSikaran = false;
        DataStatus.notificationORfirebase = true;
        DataStatus.notificationORfirebase = true;
        tombol = false;
    });
} else if (data == null) {
    dialogMasuk(
        "Akun Staf Belum Terdaftar! Silahkan "Tambahkan Akun Staf" Pada Menu
        SIKARAN QRIS menggunakan akun utama.");
    tombol = false;
} else if (data?['password'] != null &&
    data?['password'] != emailController.text) {
    dialogMasuk("Password Salah!");
    tombol = false;
}
},
).catchError((error) {
    String err = error.toString().replaceAll(

```

"[cloud_firestore/unavailable] The service is currently unavailable. This is a most likely a transient condition and may be corrected by retrying with a backoff.",

```
"Koneksi Internet Anda Terputus, Coba Lagi!");
```

```
dialogError(err);
```

```
tombol = false;
```

```
});
```

```
}
```

```
}
```

```
void lupaPassword() {
```

```
  DataStatus.loadingSikaran = true;
```

```
  FocusScope.of(context).requestFocus(FocusNode());
```

```
  Timer(const Duration(milliseconds: 1500), () {
```

```
    Navigator.of(context).pop(
```

```
      MaterialPageRoute(builder: (context) => const LupaPasswordPage());
```

```
    Navigator.of(context).push(
```

```
      MaterialPageRoute(builder: (context) => const LupaPasswordPage());
```

```
    DataStatus.loadingSikaran = false;
```

```
  });
```

```
}
```

```
void daftarAkun() {
```

```
  DataStatus.loadingSikaran = true;
```

```
  FocusScope.of(context).requestFocus(FocusNode());
```

```

Timer(const Duration(milliseconds: 1500), () {
  Navigator.of(context)
    .pop(MaterialPageRoute(builder: (context) => const DaftarPage()));
  Navigator.of(context)
    .push(MaterialPageRoute(builder: (context) => const DaftarPage()));
  DataStatus.loadingSikaran = false;
});
}
}

```

histori.dart

```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:intl/intl.dart';
import 'package:sikaran_qris/LocalData/data.dart';

class HistoryPage extends StatefulWidget {
  const HistoryPage({Key? key}) : super(key: key);

  @override
  State<HistoryPage> createState() => _HistoryPageState(
);
}

```

```

class _HistoryPageState extends State<HistoryPage> {
  @override
  Widget build(BuildContext context) {
    final FirebaseAuth auth = FirebaseAuth.instance;
    final idEmailUser = auth.currentUser?.email;

    Stream<QuerySnapshot> collectionHistory = FirebaseFirestore.instance
      .collection('users')
      .doc(DataStatus.akunStaf == false
        ? idEmailUser
        : DataStatus.currentEmailAkunStaf)
      .collection("history")
      .orderBy('date', descending: true)
      .snapshots();

    return Scaffold(
      appBar: AppBar(
        title: Text(
          "Histori Transaksi",
          style: GoogleFonts.ubuntu
        (
          color: Colors.white,
          fontSize: 17,
          fontWeight: FontWeight.w600,
        ),
      ),
    );
  }
}

```

```

    ),
  ),
  body: StreamBuilder<QuerySnapshot>
(
  stream: collectionHistory,
  builder: (context, snapshot) {
    if (snapshot.hasError) {
      return Center(
        child: Text(
          "Ada Masalah Dengan Data!",
          style: GoogleFonts.ubuntu(
            color: Colors.black,
            fontWeight: FontWeight.w600,
            fontSize: 14,
          ),
        ),
      );
    }

    if (snapshot.connectionState == ConnectionState.waiting) {
      return const Center(
        child: CircularProgressIndicator(),
      );
    }

    if (snapshot.data!.docs.isEmpty) {

```

```

return Center(
  child: Text(
    "Tidak Ada Data Histori",
    style: GoogleFonts.ubuntu(
      color: Colors.black,
      fontWeight: FontWeight.w600,
      fontSize: 14,
    ),
  ),
);
} else {
  return ItemHistory(listHistory: snapshot.data!.docs);
}
},
),
);
}
}

```

```

class ItemHistory extends StatefulWidget {
  final List<DocumentSnapshot> listHistory;
  const ItemHistory({Key? key, required this.listHistory}) : super(key: key);
  @override
  State<ItemHistory> createState() => _ItemHistoryState();
}

```

```

class _ItemHistoryState extends State<ItemHistory> {
  @override
  Widget build(BuildContext context) {
    return ListView.builder(
      itemCount: widget.listHistory.length,
      itemBuilder: (context, i) {
        String app = widget.listHistory[i].get("app").toString();
        String timestamp = widget.listHistory[i].get('date').toString();
        String money = widget.listHistory[i].get("money").toString();

        var rupiah = NumberFormat.simpleCurrency(
          locale: 'id', name: "Rp ", decimalDigits: 0)
          .format(int.parse(money));

        var dateFull =
          DateTime.fromMillisecondsSinceEpoch(int.parse(timestamp));
        String convertDate =
          dateFull.toString().substring(0, dateFull.toString().length - 7);

        return ListTile(
          title: Container(
            margin: const EdgeInsets.only(top: 5),
            padding:
              const EdgeInsets.only(top: 8, bottom: 8, left: 0, right: 5),
            decoration: BoxDecoration(
              borderRadius: const BorderRadius.all(

```

```
        Radius.circular(10),
    ),
    color: Colors.grey[300],
  ),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.start,
    children: [
      Container(
        padding: const EdgeInsets.only(left: 10, right: 8),
        child: Text(
          (i + 1).toString(),
          style: GoogleFonts.ubuntu(
            color: Colors.black,
            fontWeight: FontWeight.w700,
            fontSize: 25,
          ),
        ),
      ),
    ],
  ),
  SizedBox(
    width: MediaQuery.of(context).size.width / 1.6,
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Text(
          "$app - ${rupiah.toString()}",
          style: GoogleFonts.ubuntu(
```



```
    color: Colors.black,
    letterSpacing: 0.5,
    fontWeight: FontWeight.w700,
    fontSize: 15,
  ),
),
const SizedBox(height: 3),
Row(
  children: [
    Text(
      "Tanggal: $convertDate",
      style: GoogleFonts.ubuntu(
        color: Colors.black,
        fontWeight: FontWeight.normal,
        fontSize: 13,
      ),
    ),
    i <= 0
      ? Text(
        " New",
        style: GoogleFonts.ubuntu(
          color: Colors.red,
          fontWeight: FontWeight.bold,
          fontSize: 13,
        ),
      )
  ]
)
```

```

        : Container()
        ],
        ),
        ]),
        ),
        ],
        ),
        ),
        );
    },
    );
}
}

```

lupapassword.dart

```

import 'dart:async';
import 'package:awesome_dialog/awesome_dialog.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/services.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:lottie/lottie.dart';
import 'package:sikaran_qris/LocalData/data.dart';
import 'package:sikaran_qris/Login/login.dart';

```

```

class LupaPasswordPage extends StatefulWidget {
  const LupaPasswordPage({Key? key}) : super(key: key);

  @override
  State<StatefulWidget> createState() => _State();
}

class _State extends State<LupaPasswordPage> {
  TextEditingController emailController = TextEditingController();
  final akunAuth = FirebaseAuth.instance;
  bool jalankan = false;
  bool tombol = false;

  @override
  void initState() {
    super.initState();
    DataStatus.loadingSikaran = false;
  }

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

```

```

@override
Widget build(BuildContext context) {
  DataStatus.orientationData = MediaQuery.of(context)
    .orientation
    .toString()
    .replaceAll("Orientation.portrait", "Portrait")
    .replaceAll("Orientation.landscape", "Landscape");

  if (DataStatus.orientationData == "Portrait")
  {
    SystemChrome.setEnabledSystemUIMode(SystemUiMode.edgeToEdge);
  }

  if (DataStatus.orientationData == "Landscape") {
    SystemChrome.setEnabledSystemUIMode(SystemUiMode.immersiveSticky);
  }

  return Scaffold(
    body: WillPopScope(
      onWillPop: () async {
        final value = await showDialog<bool>(
          context: context,
          builder: (context) {
            return AlertDialog(
              title: const Text("Pemberitahuan"),
              content: const Text("Kamu Ingin Keluar Dari Aplikasi?"),

```

```

actions: [
  Row(
    mainAxisAlignment: MainAxisAlignment.spaceAround,
    children: [
      ElevatedButton(
        onPressed: () => Navigator.of(context).pop(false),
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.blue,
          padding: const EdgeInsets.all(12),
        ),
        child: const Text("Tidak"),
      ),
      ElevatedButton(
        onPressed: () => Navigator.of(context).pop(true),
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.red,
          padding: const EdgeInsets.all(12),
        ),
        child: const Text("Iya"),
      )
    ],
  )
];
});
if (value != null) {

```

```

    return Future.value(value);
  } else
  {
    return Future.value(false);
  }
},
child: AnimatedContainer(
  duration: const Duration(seconds: 6),
  curve: Curves.fastOutSlowIn,
  decoration: BoxDecoration(
    gradient: LinearGradient(
      colors: const [
        Color.fromARGB(255, 202, 202, 225),
        Color.fromARGB(255, 234, 234, 239),
        Color.fromARGB(255, 242, 241, 252),
      ],
      end: jalankan
        ? Alignment.bottomLeft
        : AlignmentDirectional.topCenter,
      begin:
        jalankan ? Alignment.centerRight : Alignment.centerLeft)),
  child: Center(
    child: ListView(
      physics: const BouncingScrollPhysics(),
      shrinkWrap: true,
      padding: const EdgeInsets.all(30),

```

```

children: [
  // ===== //
  Padding(
    padding: const EdgeInsets.only(bottom: 30),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        Lottie.asset("assets/lottie/loading_loop_qr.json",
          width: 50, height: 40),
        Padding(
          padding: const EdgeInsets.only(bottom: 4),
          child: Text(
            "SIKARAN QRIS",
            style: GoogleFonts.poppins(
              color: const Color.fromRGBO(49, 48, 146, 1.0),
              fontSize: 30,
              fontWeight: FontWeight.w600,
            ),
          ),
        ],
      ),
    ),
  // ===== //

```

```
DataStatus.loadingSikaran == true
```

```
? Container()
: Column(
  children: [
    Container(
      padding: const EdgeInsets.only(bottom: 10, top: 10),
      child: TextField(
        style: GoogleFonts.ubuntu(
          color: Colors.grey[800],
          fontSize: 17,
          fontWeight: FontWeight.w600,
        ),
        controller: emailController,
        decoration: InputDecoration(
          isDense: true,
          hintText: 'Masukkan Email',
          hintStyle: GoogleFonts.ubuntu(
            color: Colors.grey[800],
            fontSize: 15,
            fontWeight: FontWeight.w600,
          ),
          enabledBorder: OutlineInputBorder(
            borderSide: const BorderSide(
              color: Color.fromARGB(255, 123, 123, 123),
              width: 2,
            ),
            borderRadius: BorderRadius.circular(8),
```



```
    ),  
    focusedBorder: OutlineInputBorder(  
      borderSide: const BorderSide(  
        color: Color(0xFFDBE2E7),  
        width: 2,  
      ),  
      borderRadius: BorderRadius.circular(8),  
    ),  
    filled: true,  
    fillColor: Colors.grey[200],  
    suffixIcon: Icon(Icons.email,  
      color: Colors.grey[800])),  
  ),  
),  
// ===== //
```

```
Container(  
  alignment: Alignment.centerRight,  
  padding: const EdgeInsets.only(  
    top: 10, bottom: 10, right: 0),  
  child: Material(  
    color: Colors.transparent,  
    child: InkWell(  
      borderRadius: BorderRadius.circular(4),  
      splashColor: Colors.blueGrey,  
      onTap: () {
```

```

        loginKembali();
    },
    child: Text(
      ' Login Kembali ? ',
      style: GoogleFonts.ubuntu(
        color:
          const Color.fromRGBO(49, 48, 146, 1.0),
        fontWeight: FontWeight.w700,
        fontSize: 15,
      ),
    ),
  ),
),
),
),
),
// ===== //

const SizedBox(height: 25),

Container(
  margin: const EdgeInsets.only(top: 15, bottom: 15),
  height: 45,
  decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(10),
    gradient: const LinearGradient(
      colors: [

```

```
        Color.fromARGB(255, 66, 66, 150),
        Color.fromARGB(255, 105, 105, 174),
    ],
    begin: Alignment.topLeft,
    end: Alignment.bottomRight)),
child: Material(
    color: Colors.transparent,
    child: InkWell(
        borderRadius: BorderRadius.circular(10),
        onTap: () {
            if (tombol == false) {
                masukReset();
            }
        },
        child: Center(
            child: Text(
                'RESET PASSWORD',
                style: GoogleFonts.ubuntu(
                    color: Colors.white,
                    letterSpacing: 1,
                    fontWeight: FontWeight.w700,
                    fontSize: 16,
                ),
            ),
        ),
    ),
),
```

```
        ),  
        ),  
    ],  
    ),  
    ],  
    ),  
    ),  
    ),  
    ),  
);  
}
```

```
void dialogPesan(pesanTitle) {  
    AwesomeDialog(  
        context: context,  
        autoHide: const Duration(seconds: 6),  
        dialogType: DialogType.info,  
        animType: AnimType.scale,  
        title: pesanTitle,  
    ).show();  
}
```

```
void dialogSukses(title, pesan) {  
    AwesomeDialog(  
        context: context,  
        dialogType: DialogType.success,
```

```
animType: AnimType.scale,  
body: Container(  
  margin: const EdgeInsets.all(20),  
  child: Column(  
    children: [  
      Text(  
        title,  
        style: GoogleFonts.roboto(  
          color: Colors.black,  
          fontSize: 20,  
          fontWeight: FontWeight.bold,  
        ),  
      ),  
      const SizedBox(height: 20),  
      Text(  
        pesan,  
        textAlign: TextAlign.justify,  
        style: GoogleFonts.roboto(  
          color: Colors.black,  
          fontSize: 15.5,  
          fontWeight: FontWeight.w700,  
        ),  
      ),  
    ],  
  )),  
btnOkOnPress: () {
```

```
        loginKembali();
    },
    btnOkText: "Oke")
    .show();
}
```

```
void dialogError(String pesanTitle) {
    AwesomeDialog(
        context: context,
        autoHide: const Duration(seconds: 10),
        dialogType: DialogType.error,
        animType: AnimType.scale,
        title: pesanTitle,
    ).show();
}
```

```
void toastShort(message, Color colorBox) {
    Fluttertoast.showToast(
        msg: message,
        toastLength: Toast.LENGTH_SHORT,
        gravity: ToastGravity.BOTTOM,
        backgroundColor: colorBox,
    );
}
```

```
void loginKembali() {
```

```

DataStatus.loadingSikaran = true;
FocusScope.of(context).requestFocus(FocusNode());

Timer(const Duration(milliseconds: 1500), () {
  Navigator.of(context)
    .pop(MaterialPageRoute(builder: (context) => const LoginPage()));
  Navigator.of(context)
    .push(MaterialPageRoute(builder: (context) => const LoginPage()));
  DataStatus.loadingSikaran = false;
});
}

void masukReset() {
  FocusScope.of(context).requestFocus(FocusNode());
  if (emailController.text == "") {
    dialogPesan("Masukkan Email Anda!");
  } else {
    resetPassword();
  }
}

void resetPassword() async {
  tombol = true;
  try {
    await akunAuth
      .sendPasswordResetEmail(email: emailController.text.toString().trim())

```

```

        .then((value) => dialogSucces("RESET PASSWORD",
            "Berhasil Melakukan Reset Password Pada Email Anda. Silahkan Cek
            Kotak Masuk Email Anda, Jika Tidak Ada Cek Menu Spam Email Anda!"));
        toastShort("Berhasil Reset Password", Colors.green);
        tombol = false;
    } on FirebaseAuthException catch (err) {
        String error = err
            .toString()
            .replaceAll("[firebase_auth/unknown] Given String is empty or null",
                "Form Tidak Boleh Kosong!")
            .replaceAll(
                "[firebase_auth/invalid-email] The email address is badly formatted.",
                "Format Email Yang Anda Masukkan Salah!")
            .replaceAll(
                "[firebase_auth/wrong-password] The password is invalid or the user does
                not have a password.",
                "Email atau Password Yang Anda Masukkan Salah!")
            .replaceAll(
                "[firebase_auth/user-not-found] There is no user record corresponding to
                this identifier. The user may have been deleted.",
                "Email Yang Anda Masukkan Tidak Terdaftar!")
            .replaceAll(
                "[firebase_auth/network-request-failed] A network error (such as timeout,
                interrupted connection or unreachable host) has occurred.",
                "Koneksi Internet Anda Terputus, Coba Lagi!")
            .replaceAll(
                "[firebase_auth/too-many-requests] We have blocked all requests from this
                device due to unusual activity. Try again later.",

```



```
        "Terlalu Sering Melakukan Reset Password di Perangkat Ini, Silahkan Coba  
Lagi Nanti!");
```

```
        dialogError(error.toString());  
        tombol = false;  
    }  
}  
}
```

kelolaakunstaf.dart

```
import 'package:awesome_dialog/awesome_dialog.dart';  
import 'package:cloud_firestore/cloud_firestore.dart';  
import 'package:flutter/material.dart';  
import 'package:fluttertoast/fluttertoast.dart';  
import 'package:google_fonts/google_fonts.dart';  
import 'package:sikaran_qris/LocalData/data.dart';  
  
class KelolaAkunStaf extends StatefulWidget {  
    const KelolaAkunStaf({super.key});  
  
    @override  
    State<KelolaAkunStaf> createState() => _KelolaAkunStafState();  
}  
  
TextEditingController emailController = TextEditingController();  
TextEditingController passwordController = TextEditingController();  
bool add = false;
```

```

bool _passwordVisible = false;
int? jumlahStaf;

void toastShort(message, Color colorBox) {
  Fluttertoast.showToast(
    msg: message,
    toastLength: Toast.LENGTH_SHORT,
    gravity: ToastGravity.BOTTOM,
    backgroundColor: colorBox,
  );
}

class _KelolaAkunStafState extends State<KelolaAkunStaf> {
  // ===== Firestore
  ===== //
  Stream<QuerySnapshot> collectionStaf = FirebaseFirestore.instance
    .collection('users')
    .doc(DataStatus.currentEmail.toString().trim())
    .collection('staf')
    .orderBy('date', descending: true)
    .snapshots();
  // ===== Firestore
  ===== //

  void saveAkunStaf(BuildContext context) {
    FirebaseFirestore firestore = FirebaseFirestore.instance;
    DocumentReference akunStaf = firestore

```

```

.collection('users')
.doc(DataStatus.currentEmail)
.collection('staf')
.doc(
    "${emailController.text.toString().trim()}:${DataStatus.currentEmail}");

var waktuSekarang = Timestamp.now().millisecondsSinceEpoch;

akunStaf.set({
    'email':
        "${emailController.text.toString().trim()}:${DataStatus.currentEmail}",
    'admin': DataStatus.currentEmail,
    'password': passwordController.text.trim().toString(),
    'date': waktuSekarang,
}).then(
    (value) => {
        toastShort("Berhasil Menambahkan Akun Staf", Colors.green),
        setState() {
            FocusScope.of(context).requestFocus(FocusNode());
            emailController.text = "";
            passwordController.text = "";
            add = !add;
        },
    },
);
}

```

```
void cekJumlah() {  
    if (jumlahStaf == 0) {  
        setState() {  
            add = false;  
        });  
    } else {  
        setState() {  
            add = true;  
        });  
    }  
}
```

```
@override  
void initState() {  
    super.initState();  
    cekJumlah();  
}
```

```
@override  
Widget build(BuildContext context) {  
    return Scaffold(  
        appBar: AppBar(  
            title: Text(  
                "Kelola Akun Staf",  
                style: GoogleFonts.ubuntu(  
                    color: Colors.black,                    size: 18,                    weight: FontWeight.normal,                    style: FontStyle.italic,                ),  
            ),  
        ),  
    );  
}
```

```

        color: Colors.white,
        fontSize: 17,
        fontWeight: FontWeight.w600,
    ),
),
),
body: StreamBuilder<QuerySnapshot>(
  stream: collectionStaf,
  builder: (context, snapshot) {
    if (snapshot.hasError) {
      return Container(
        alignment: Alignment.topCenter,
        margin: const EdgeInsets.only(top: 50),
        child: Text(
          "Gagal Mendapatkan Data",
          style: GoogleFonts.ubuntu(
            color: Colors.black,
            fontWeight: FontWeight.w600,
            fontSize: 14,
          ),
        ),
      );
    }

    if (snapshot.connectionState == ConnectionState.waiting) {
      return const Center(

```

```

        child: CircularProgressIndicator(),
      );
    }

    if (snapshot.hasData) {
      jumlahStaf = snapshot.data!.docs.length;
    }

    if (jumlahStaf == 0) {
      return Container(
        alignment: Alignment.topCenter,
        margin: const EdgeInsets.only(top: 50),
        child: Text(
          "Tidak Ada Data User Staf",
          style: GoogleFonts.ubuntu(
            color: Colors.black,
            fontWeight: FontWeight.w600,
            fontSize: 14,
          ),
        ),
      );
    } else {
      return ItemUserStaf(listHistory: snapshot.data!.docs);
    }
  },
),

```

```

floatingActionButton: FloatingActionButton(
  backgroundColor: Colors.indigo,
  child: Icon(add == true ? Icons.add_outlined : Icons.remove),
  onPressed: () {
    setState() {
      add = !add;
      FocusScope.of(context).requestFocus(FocusNode());
    });
  },
),
bottomSheet: AnimatedContainer(
  height: add == false ? 230 : 0,
  margin: const EdgeInsets.only(left: 0),
  duration: const Duration(milliseconds: 700),
  curve: Curves.fastOutSlowIn,
  decoration: BoxDecoration(
    borderRadius: const BorderRadius.only(
      topLeft: Radius.circular(40),
      topRight: Radius.circular(40),
    ),
    color: Colors.grey[300],
  ),
  child: Stack(children: [
    Container(
      padding:
        const EdgeInsets.only(bottom: 10, top: 35, left: 30, right: 30),

```

```
child: TextField(  
  style: GoogleFonts.ubuntu(  
    color: Colors.grey[800],  
    fontSize: 17,  
    fontWeight: FontWeight.w600,  
  ),  
  controller: emailController,  
  decoration: InputDecoration(  
    isDense: true,  
    hintText: 'Masukkan Username',  
    hintStyle: GoogleFonts.ubuntu(  
      color: Colors.grey[800],  
      fontSize: 15,  
      fontWeight: FontWeight.w600,  
    ),  
    focusedBorder: OutlineInputBorder(  
      borderRadius: BorderRadius.circular(8),  
      borderSide: const BorderSide(  
        color: Color.fromARGB(255, 123, 123, 123),  
        width: 2,  
      ),  
    ),  
    filled: true,  
    fillColor: Colors.white,  
    suffixIcon: Icon(Icons.email, color: Colors.grey[800])),  
),
```


),

// ===== //

Container(

padding:

const EdgeInsets.only(bottom: 10, top: 95, left: 30, right: 30),

child: TextField(

style: GoogleFonts.ubuntu(

color: Colors.grey[800],

fontSize: 17,

fontWeight: FontWeight.w600,

),

obscureText: !_passwordVisible,

controller: passwordController,

decoration: InputDecoration(

isDense: true,

hintText: 'Masukkan Password',

hintStyle: GoogleFonts.ubuntu(

color: Colors.grey[800],

fontSize: 15,

fontWeight: FontWeight.w600,

),

focusedBorder: OutlineInputBorder(

borderRadius: BorderRadius.circular(8),

borderSide: const BorderSide(

```

        color: Color.fromARGB(255, 123, 123, 123),
        width: 2,
    ),
),
filled: true,
fillColor: Colors.white,
suffixIcon: IconButton(
    icon: Icon(
        // Based on passwordVisible state choose the icon
        _passwordVisible ? Icons.visibility : Icons.visibility_off,
        color: Colors.grey[800],
    ),
    onPressed: () {
        setState(() {
            _passwordVisible = !_passwordVisible;
        });
    },
),
//====//
),
),
),
// ===== //

Container(
    margin: const EdgeInsets.only(top: 160, left: 30, right: 30),

```

```
height: 45,
decoration: BoxDecoration(
  borderRadius: BorderRadius.circular(10),
  gradient: const LinearGradient(colors: [
    Color.fromARGB(255, 66, 66, 150),
    Color.fromARGB(255, 105, 105, 174),
  ], begin: Alignment.topLeft, end: Alignment.bottomRight)),
child: Material(
  color: Colors.transparent,
  child: InkWell(
    borderRadius: BorderRadius.circular(10),
    onTap: () {
      cekForm(context);
      setState(() {});
    },
    child: Center(
      child: Text(
        'TAMBAH | UPDATE AKUN STAF',
        style: GoogleFonts.ubuntu(
          color: Colors.white,
          letterSpacing: 1,
          fontWeight: FontWeight.w700,
          fontSize: 13.5,
        ),
      ),
    ),
  ),
),
```

```

        ),
    ),
),
// ===== //
]),
),
);
}

```

```

void dialogMasuk(pesanTitle) {
    AwesomeDialog(
        context: context,
        autoHide: const Duration(seconds: 6),
        dialogType: DialogType.info,
        animType: AnimType.scale,
        title: pesanTitle,
    ).show();
}

```

```

void cekForm(BuildContext context) {
    // FocusScope.of(context).requestFocus(FocusNode());
    if (emailController.text.trim() == "") {
        dialogMasuk("Masukkan Username!");
    } else {
        if (emailController.text.contains("@")) {
            dialogMasuk("Username Tidak Boleh Mengandung Karakter ( @ )");
        }
    }
}

```



```

final List<DocumentSnapshot> listHistory;
const ItemUserStaf({Key? key, required this.listHistory}) : super(key: key);
@override
State<ItemUserStaf> createState() => _ItemUserStaf();
}

```

```

void deleteAkunStaf(String data) {
  FirebaseFirestore firestore = FirebaseFirestore.instance;
  DocumentReference akunStaf = firestore
    .collection('users')
    .doc(DataStatus.currentEmail.toString().trim())
    .collection('staf')
    .doc(data);
  akunStaf.delete();
}

```

```

class _ItemUserStaf extends State<ItemUserStaf> {
  @override
  Widget build(BuildContext context) {
    return ListView.builder(
      itemCount: widget.listHistory.length,
      itemBuilder: (context, i) {
        String emailStaf = widget.listHistory[i].get("email").toString();
        String timestamp = widget.listHistory[i].get('date').toString();
        var dateFull =
          DateTime.fromMillisecondsSinceEpoch(int.parse(timestamp));

```

```
String convertDate =
    dateFull.toString().substring(0, dateFull.toString().length - 7);

return ListTile(
  title: Container(
    margin: const EdgeInsets.only(top: 5),
    padding:
      const EdgeInsets.only(top: 8, bottom: 8, left: 10, right: 0),
    decoration: BoxDecoration(
      borderRadius: const BorderRadius.all(
        Radius.circular(10),
      ),
      color: Colors.grey[300],
    ),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        Column(
          mainAxisAlignment: MainAxisAlignment.start,
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            SizedBox(
              width: MediaQuery.of(context).size.width / 1.6,
              child: Text(
                overflow: TextOverflow.ellipsis,
                maxLines: 3,
```

```
    "Email: $emailStaf",
    style: GoogleFonts.ubuntu(
      color: Colors.black,
      letterSpacing: 0.5,
      fontWeight: FontWeight.w700,
      fontSize: 15,
    ),
  ),
),
const SizedBox(height: 3),
Text(
  "Tanggal dibuat: $convertDate",
  style: GoogleFonts.ubuntu(
    color: Colors.black,
    fontWeight: FontWeight.normal,
    fontSize: 13,
  ),
),
]),
ElevatedButton(
  style: ElevatedButton.styleFrom(
    shadowColor: const Color.fromARGB(159, 0, 0, 0),
    elevation: 0,
    backgroundColor: Colors.transparent,
  ),
  onPressed: () {
```



```
        deleteAkunStaf(emailStaf);
    },
    child: const Icon(
      Icons.delete,
      size: 35,
      color: Colors.red,
    ))
  ],
),
);
},
);
}
```

drawer.dart

```
import 'dart:async';
import 'package:awesome_dialog/awesome_dialog.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:flutter_custom_dialog/flutter_custom_dialog.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:google_fonts/google_fonts.dart';
```

```

import 'package:image_picker/image_picker.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:sikaran_qris/History/history.dart';
import 'package:sikaran_qris/HomeDrawer/home.dart';
import 'package:sikaran_qris/LocalData/data.dart';
import 'package:sikaran_qris/Login/login.dart';
import 'package:sikaran_qris/Pengaturan/pengaturan.dart';
import 'package:sikaran_qris/Widget/all_widget.dart';
import 'package:slider_controller/slider_controller.dart';
import 'package:video_player/video_player.dart';

```

```

class IsiDrawer extends StatefulWidget {
  const IsiDrawer({Key? key}) : super(key: key);

  @override
  State<IsiDrawer> createState() => _IsiDrawerState();
}

```

```

class _IsiDrawerState extends State<IsiDrawer> {
  //===== Fungsi Pick Video
  =====//

  void readSP() async {
    final spDir = await SharedPreferences.getInstance();
    setState() {
      if (spDir.getString('fileImageVideo') != null) {
        if (DataStatus.modeBackground == "mp4" ||

```

```
        DataStatus.modeBackground == "mov") {
            modeFile = true;
        } else {
            modeFile = false;
        }

        volumeVideo = DataStatus.volumeVideo.toDouble();
    }
});
}
```

```
void klikfoto() {
    deleteImageVideo();
    DeleteCacheAll().clearCahce();
    DataStatus.imageVideo = null;
    setState() {
        IsiDrawer;
    });
    pickImage();
}
```

```
void klikvideo() {
    deleteImageVideo();
    DeleteCacheAll().clearCahce();
    DataStatus.imageVideo = null;
    setState() {
```

```
    IsiDrawer;  
  });  
  pickVideo();  
}
```

```
void pickImage() async {  
  final ImagePicker picker = ImagePicker();  
  final XFile? pickerImage =  
    await picker.pickImage(source: ImageSource.gallery);  
  if (pickerImage?.path != null) {  
    setState() {  
      DataStatus.ambilimage = pickerImage!.path;  
      saveImage();  
    });  
  }  
}
```

```
void saveImage() async {  
  final saveimage = await SharedPreferences.getInstance();  
  setState() {  
    saveimage.setString('fileImageVideo', DataStatus.ambilimage!);  
    Navigator.pushReplacement(  
      context,  
      MaterialPageRoute(  
        builder: (BuildContext context) => const HomeDrawer());  
      YYDialog.init(context);  
    });  
}
```

```
});  
}
```

```
void pickVideo() async {  
  final ImagePicker picker = ImagePicker();  
  final XFile? pickerVideo =  
    await picker.pickVideo(source: ImageSource.gallery);  
  if (pickerVideo?.path != null) {  
    setState() {  
      DataStatus.ambilvideo = pickerVideo!.path;  
      saveVideo();  
    });  
  }  
}
```

```
void saveVideo() async {  
  final savevideo = await SharedPreferences.getInstance();  
  setState() {  
    savevideo.setString('fileImageVideo', DataStatus.ambilvideo!);  
    Navigator.pushReplacement(  
      context,  
      MaterialPageRoute(  
        builder: (BuildContext context) => const HomeDrawer());  
      YYDialog.init(context);  
    });  
}
```

```
//===== Fungsi Pick Video
=====//
```

```
//===== Variable
=====//
```

```
VideoPlayerController? videoController;
final loginakun = FirebaseAuth.instance;
bool modeFile = false;
double volumeVideo = 0;
```

```
//===== Variable
=====//
```

```
@override
void initState() {
  super.initState();
}
```

```
@override
Widget build(BuildContext context) {
  readSP();
  YYDialog.init(context);
```

```
return SafeArea(
  child: Container(
    padding: const EdgeInsets.all(15),
    child: Column(
```

```

crossAxisAlignment: CrossAxisAlignment.stretch,
children: [
  profile(),
  garis,
  menu("Custom Background", Icons.add_photo_alternate,
    customBackground, FontWeight.normal),
  modeFile == true
    ? menu("Volume Suara Video", Icons.video_settings,
      pengaturanVideo, FontWeight.normal)
    : Container(),
  menu("Histori Transaksi", Icons.history, navigasiHistory,
    FontWeight.normal),
  menu("Penggunaan", Icons.help, penggunaan, FontWeight.normal),
  menu("Pengaturan", Icons.settings, navigasiPengaturan,
    FontWeight.normal),
  DataStatus.akunStaf == false
    ? menu("Akun User", Icons.account_circle, akunUser,
      FontWeight.normal)
    : menu("Akun Staf", Icons.account_circle, akunStaf,
      FontWeight.normal),
  menu("Sign Out", Icons.logout, keluarAkun, FontWeight.normal),
  menu(
    "Keluar", Icons.power_settings_new, keluarApk, FontWeight.bold),
],
),
),

```

```
);  
}
```

```
YYDialog customBackground() {  
  return YYDialog().build()  
    ..width = 300  
    ..borderRadius = 4.0  
    ..text(  
      alignment: Alignment.center,  
      padding:  
        const EdgeInsets.only(left: 18.0, right: 18.0, bottom: 18, top: 30),  
      text: "Custom Background QRIS",  
      color: Colors.black,  
      fontSize: 18.0,  
      fontWeight: FontWeight.w500,  
    )  
    ..text(  
      alignment: Alignment.center,  
      padding: const EdgeInsets.only(left: 18.0, right: 18.0, bottom: 5),  
      text: "Rekomendasi Ukuran Foto/Video",  
      color: Colors.grey[500],  
      fontSize: 15.0,  
    )  
    ..text(  
      alignment: Alignment.center,  
      padding: const EdgeInsets.only(left: 18.0, right: 18.0, bottom: 5),
```



```

text: "File: PNG, JPG, JPEG, MOV, MP4",
color: Colors.grey[500],
fontSize: 15.0,
)
..text(
alignment: Alignment.center,
padding: const EdgeInsets.only(left: 18.0, right: 18.0, bottom: 5),
text:
    "Width: ${MediaQuery.of(context).size.width * 3}.toInt() dan Height:
    ${MediaQuery.of(context).size.height * 3}.toInt()",
color: Colors.grey[500],
fontSize: 15.0,
)
..text(
alignment: Alignment.center,
padding: DataStatus.height != 0 && DataStatus.width != 0
    ? const EdgeInsets.only(left: 18.0, right: 18.0, bottom: 5)
    : const EdgeInsets.all(0),
text: DataStatus.height != 0 && DataStatus.width != 0
    ? "Rasio Layar: ${DataStatus.width.toInt()} : ${DataStatus.height.toInt()}"
    : "",
color: Colors.grey[500],
fontSize: DataStatus.height != 0 && DataStatus.width != 0 ? 15.0 : 0.0,
)
..text(
alignment: Alignment.center,

```

```

padding: const EdgeInsets.only(left: 18.0, right: 18.0, bottom: 14),
text: "Mode: ${DataStatus.orientationData}",
color: Colors.grey[500],
fontSize: 15.0,
)
)..doubleButton(
gravity: Gravity.spaceEvenly,
text1: "Foto",
color1: Colors.deepPurpleAccent,
fontSize1: 14.0,
fontWeight1: FontWeight.bold,
onTap1: () {
klikfoto();
},
text2: "Video",
color2: Colors.deepPurpleAccent,
fontSize2: 14.0,
fontWeight2: FontWeight.bold,
onTap2: () {
klikvideo();
},
)
)..show();
}

```

```

YYDialog pengaturanVideo() {

```

```
return YYDialog().build()

  ..width = MediaQuery.of(context).size.width / 1.5
  ..backgroundColor = Colors.white
  ..borderRadius = 10.0
  ..gravity = Gravity.center
  ..widget(Column(
    children: [
      Container(
        padding: const EdgeInsets.fromLTRB(0, 30, 0, 0),
        child: Text(
          "Volume Video",
          style: GoogleFonts.ubuntu(
            color: Colors.grey[800],
            fontSize: 20,
            fontWeight: FontWeight.bold,
          ),
        ),
      ),
      Container(
        padding: const EdgeInsets.all(30),
        child: SliderController(
          min: 0.0,
          max: 1.0,
          value: volumeVideo,
          onChanged: (value) {
            setState() {
```

```

        saveVolumeVideo(value);
        DataStatus.volumeVideo = value;
    });
},
),
),
],
))
..show();
}

```

```

YYDialog penggunaan() {
return YYDialog().build()
..width = MediaQuery.of(context).size.width / 1.1
..borderRadius = 3.0
..widget(Container(
padding:
const EdgeInsets.only(left: 15, right: 15, bottom: 15, top: 15),
child: Column(
children: [
Text(
"SIKARAN QRIS",
textAlign: TextAlign.center,
style: GoogleFonts.poppins(
color: Colors.black87,
fontWeight: FontWeight.bold,

```

```
        fontSize: 17),
    ),
    const SizedBox(height: 5),
    Text(
```

"merupakan aplikasi yang menyediakan fitur verifikasi pembayaran QRIS. Aplikasi ini dibuat untuk membantu para pelaku usaha atau merchant QRIS dalam memverifikasi pembayaran QRIS dari pelanggan. Aplikasi ini mungkin masih ada kekurangan namun untuk fitur-fitur utamanya sudah bisa digunakan dengan sebaiknya. Aplikasi ini sangat cocok di implementasikan menjadi pajangan di tempat usaha dengan menggunakan Monitor/TV yang memiliki sistem operasi Android 5.0+ sebagai alat verifikasi pembayaran QRIS yang realtime. Jika Android Box/Tv anda tidak support Notifikasi, maka jadikan HP anda sebagai penerima Notifikasi pembayaran dan Android Box/Tv anda sebagai penampil Suara dan Dialog Box Verification. Android Box/Tv anda wajib login menggunakan akun staf yang bisa anda buat di menu pengaturan."

```
        textAlign: TextAlign.center,
        style: GoogleFonts.poppins(
            color: Colors.black87,
            fontWeight: FontWeight.w700,
            fontSize: 13),
    ),
    const SizedBox(height: 5),
```

```
    Text(
```

"*Catatan: 1) Anda Wajib Memberikan Izin Akses Aplikasi Untuk Menangkap Data Notifikasi Aplikasi QRIS. 2) SIKARAN QRIS Saat Ini Hanya Baru Bisa Menangkap Data Notifikasi Dari Aplikasi Shopee Partner, Tentu Kami Akan Terus Update Aplikasi Agar Bisa Menangkap Data Notifikasi Aplikasi QRIS Lainnya."

```
        textAlign: TextAlign.center,
        style: GoogleFonts.poppins(
            color: Colors.black87,
```

```
        fontWeight: FontWeight.w600,  
        fontSize: 13),  
    ),  
  ],  
),  
))  
..show();  
}
```

```
YYDialog akunUser() {  
  return YYDialog().build()  
    ..width = 300  
    ..borderRadius = 4.0  
    ..text(  
      alignment: Alignment.center,  
      padding:  
        const EdgeInsets.only(left: 18.0, right: 18.0, bottom: 18, top: 30),  
      text: "Account User",  
      color: Colors.black,  
      fontSize: 18.0,  
      fontWeight: FontWeight.w500,  
    )  
    ..text(  
      alignment: Alignment.center,  
      padding: const EdgeInsets.only(left: 18.0, right: 18.0, bottom: 30),  
      text: "Email : ${DataStatus.currentEmail}",
```

```

        color: Colors.grey[700],
        fontSize: 15.5,
    )
    // ..widget(
    //   Container(
    //     padding: const EdgeInsets.only(left: 18.0, right: 18.0, bottom: 10),
    //     child: ElevatedButton(
    //       style: ElevatedButton.styleFrom(backgroundColor: Colors.grey),
    //       onPressed: () async {
    //         User? firebaseUser = FirebaseAuth.instance.currentUser;
    //         bool emailKu = firebaseUser!.emailVerified;
    //         if (emailKu == true) {
    //           await firebaseUser.sendEmailVerification();
    //           toastShort("Mengirim Link Email Verifikasi", Colors.black54);
    //         } else {
    //           toastShort("Email Anda Sudah diverifikasi!", Colors.black54);
    //         }
    //       },
    //       child: Text(
    //         "Verifikasi Email",
    //         style: GoogleFonts.ubuntu(
    //           color: Colors.white,
    //           fontSize: 14,
    //           fontWeight: FontWeight.bold,
    //         ),
    //       ),
    //     ),
  
```

```
// ),  
// ),  
// )  
..show();  
}
```

```
YYDialog akunStaf() {  
  return YYDialog().build()  
    ..width = 300  
    ..borderRadius = 4.0  
    ..text(  
      alignment: Alignment.center,  
      padding:  
        const EdgeInsets.only(left: 18.0, right: 18.0, bottom: 18, top: 30),  
      text: "Account Staf",  
      color: Colors.black,  
      fontSize: 18.0,  
      fontWeight: FontWeight.w500,  
    )  
    ..text(  
      alignment: Alignment.center,  
      padding: const EdgeInsets.only(left: 18.0, right: 18.0, bottom: 30),  
      text: "Email : ${DataStatus.emailAkunStaf}",  
      color: Colors.grey[500],  
      fontSize: 15.0,  
    )  
}
```



```
    ..show();  
}
```

```
void keluarAkun() {  
    DataStatus.signOut = true;  
    AwesomeDialog(  
        context: context,  
        dialogType: DialogType.info,  
        animType: AnimType.scale,  
        title: "Keluar Akun?",  
        btnCancelOnPress: () {},  
        btnCancelText: "Tidak",  
        btnOkText: "Iya",  
        btnOkOnPress: logOut)  
    .show();  
}
```

```
void keluarApk() {  
    AwesomeDialog(  
        context: context,  
        dialogType: DialogType.warning,  
        animType: AnimType.scale,  
        title: "Kamu Ingin Keluar Dari Aplikasi?",  
        btnCancelOnPress: () {},  
        btnCancelText: "Tidak",  
        btnOkText: "Iya",
```

```

        btnOkOnPress: () {
            SystemNavigator.pop();
        }).show();
    }

void saveVolumeVideo(double data) async {
    final spDir = await SharedPreferences.getInstance();
    setState(() {
        spDir.setDouble('volumeVideo', data);
    });
}

void navigasiHistory() {
    Navigator.of(context)
        .push(MaterialPageRoute(builder: (context) => const HistoryPage()));
}

void navigasiPengaturan() {
    Navigator.of(context)
        .push(MaterialPageRoute(builder: (context) => const Pengaturan()));
}

Future<void> deleteImageVideo() async {
    final spDir = await SharedPreferences.getInstance();
    DataStatus.imageQR = null;
    await spDir.remove('fileImageVideo');
}

```

```

    await spDir.remove('fileImageQR');
  }

void saveData(bool notification, String emailAkunStaf, bool akunStaf) async {
  final spDir = await SharedPreferences.getInstance();
  setState() {
    spDir.setBool('notificationORfirebase', notification);
    spDir.setBool('akunStaf', akunStaf);
  });
}

void logOut() async {
  await loginakun
    .signOut()
    .then((uid) => {
      Timer(const Duration(milliseconds: 3000), () {
        Fluttertoast.showToast(msg: "Keluar Akun");
        DeleteCacheAll().deleteSharedPreferences();
        DeleteCacheAll().clearCache();
        DataStatus.login = true;
        DataStatus.imageVideo = null;
        DataStatus.notificationORfirebase = false;
        saveData(false, "", false);
        Navigator.of(context).pushReplacement(
          MaterialPageRoute(builder: (context) => const LoginPage()));
        DataStatus.akunStaf = false;
      });
    });
}

```

```

        DataStatus.emailAkunStaf = "";
    },
    })
// ignore: body_might_complete_normally_catch_error
    .catchError((err) {
showDialog(
    context: context,
    builder: (BuildContext context) {
    return AlertDialog(
        title: const Text("Peringatan"),
        content: Text(err.message),
        actions: [
            TextButton(
                child: const Text("Oke"),
                onPressed: () {
                    Navigator.of(context).pop();
                },
            )
        ],
    );
    });
    }
}

loading(context) {

```

```

AlertDialog alert = AlertDialog(
  content: Row(
    crossAxisAlignment: CrossAxisAlignment.center,
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      const CircularProgressIndicator(),
      Container(
        margin: const EdgeInsets.only(left: 25),
        child: const Text("Loading...")),
    ],
  ),
);
showDialog(
  barrierDismissible: false,
  context: context,
  builder: (BuildContext context) {
    return alert;
  },
);
}

// ===== Widget Profile ===== //
Widget profile() {
  return Container(
    margin: const EdgeInsets.only(top: 10, bottom: 10),
    child: Row(

```

```
children: [
  const Icon(
    Icons.qr_code,
    color: Colors.white,
    size: 60,
  ),
  const SizedBox(width: 10),
  Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text(
        "SIKARAN QRIS ",
        style: GoogleFonts.roboto(
          color: Colors.white,
          fontSize: 24,
          letterSpacing: 0.3,
        ),
      ),
      Text(
        "Sistem Verifikasi Pembayaran",
        style: TextStyle(
          color: Colors.white.withOpacity(0.5),
          fontSize: 12,
        ),
      ),
    ],
  ),
  DataStatus.orientationData == "Portrait"
```

```

? Text(
  "QR Code Indonesian Standard",
  style: TextStyle(
    color: Colors.white.withOpacity(0.5),
    fontSize: 12,
  ),
)
: Text(
  "Quick Response Code Indonesian Standard",
  style: TextStyle(
    color: Colors.white.withOpacity(0.5),
    fontSize: 12,
  ),
),
],
),
],
),
);
}
// ===== Widget Profile ===== //

// ===== Widget Menu =====
//
Widget menu(String nama, IconData icon, onTap, FontWeight bold) {
  return Material(

```

```
color: Colors.transparent,
child: InkWell(
  borderRadius: BorderRadius.circular(12),
  onTap: onTap,
  child: Container(
    padding: const EdgeInsets.only(left: 6, top: 14, bottom: 14),
    child: Row(
      children: [
        Icon(
          icon,
          color: Colors.white,
        ),
        const SizedBox(width: 15),
        Text(
          nama,
          style: TextStyle(
            fontSize: 20,
            fontWeight: bold,
            color: Colors.white,
          ),
        ),
      ],
    ),
  )),
);
}
```



```
// ===== Widget Menu =====  
//
```

```
// ===== Membuat Widget Khusus ===== //
```

```
Widget get garis => Container(  
  margin: const EdgeInsets.symmetric(vertical: 10),  
  height: 2,  
  color: Colors.white.withOpacity(0.2),  
);
```

```
class MenuBox extends StatelessWidget {  
  final EdgeInsetsGeometry? padding;  
  final Widget? icon;  
  final Widget menu;  
  final Function()? onTap;
```

```
  const MenuBox({  
    Key? key,  
    required this.menu,  
    this.padding = const EdgeInsets.all(10),  
    this.icon,  
    this.onTap,  
  }) : super(key: key);
```

```
@override
```

```
Widget build(BuildContext context) {
```

```
return GestureDetector(  
  onTap: () {  
    if (onTap != null) {  
      onTap!();  
    }  
  },  
  child: Container(  
    padding: padding,  
    child: Row(  
      children: [  
        icon != null ? icon! : Container(),  
        const SizedBox(width: 15),  
        menu,  
      ],  
    ),  
  ),  
);  
}
```



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN
RISET DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara, Palembang 30139

Telp. 0711-353414 Fax. 0711-355918


Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id



LEMBAR BIMBINGAN LAPORAN TUGAS AKHIR

Nama Mahasiswa	:	Nadhila Bazhlina
NIM	:	062030701619
Jurusan / Program Studi	:	Teknik Komputer / DIII Teknik Komputer
Dosen Pembimbing	:	Yulian Mirza, S.T., M.Kom
Judul	:	Aplikasi Verifikasi Pembayaran <i>Quick Response Code Indonesian Standard</i> Berbasis <i>Android</i>

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1.	06 Juni 2023	Bimbingan Bab 1	
2.	15 Juni 2023	Revisi Bab 1 dan Bimbingan Bab 2 dan 3	
3.	20 Juni 2023	Acc Bab 1, Revisi Bab 2 dan 3	
4.	22 Juni 2023	Acc Bab 2, Revisi Bab 3	
5.	27 Juni 2023	Acc Bab 3	
6.	06 Juli 2023	Bimbingan Bab 4	

7.	11 Juli 2023	Revisi Bab 4 dan bimbingan Bab 5	
8.	14 Juli 2023	Revisi bab 4 dan Revisi Bab 5	
9.	27 Juli 2023	Laporan Bab 1,2,3,4 dan 5 Acc.	

Palembang, Juli 2023
Mengetahui, Ketua Jurusan



Azwardi, S.T., M.T

NIP. 197005232005011004



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN
RISET DAN TEKNOLOGI**

POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara, Palembang 30139

Telp. 0711-353414 Fax. 0711-355918





Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id



LEMBAR BIMBINGAN LAPORAN TUGAS AKHIR

Nama Mahasiswa	:	Nadhila Bazhlina
NIM	:	062030701619
Jurusan / Program Studi	:	Teknik Komputer / DIII Teknik Komputer
Dosen Pembimbing	:	Ir. Alan Novi Tompunu, S.T., M.T., IPM
Judul	:	Aplikasi Verifikasi Pembayaran <i>Quick Response</i> <i>Code Indonesian Standard</i> Berbasis <i>Android</i>

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1.	30 Mei 2023	Pembuatan Software Program software dengan progress 60%.	
2.	05 Juni 2023	Pembuatan Software Program software dengan progress 70%.	
3.	06 Juni 2023	Bimbingan Laporan Bab 1	
4.	08 Juni 2023	Bimbingan dan Membuat Laporan bab 2	
5.	20 Juni 2023	Pembuatan software dan membuat laporan bab 3	
6.	22 Juni 2023	Bimbingan menunjukkan aplikasi	

7.	6 Juli 2023	Pembuatan software dan membuat laporan bab 4	
8.	12 Juli 2023	Melanjutkan laporan akhir hingga bab 5	
9.	24 Juli 2023	Merevisi laporan akhir	
10.	26 Juli 2023	Aplikasi dan laporan selesai ACC.	

Palembang, Juli 2023
Mengetahui, Ketua Jurusan



Azwardi, S.T., M.T

NIP. 197005232005011004



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara, Palembang 30139

Telp. 0711-353414 Fax. 0711-355918

Website: www.polsri.ac.id Email: info@polsri.ac.id



REKOMENDASI UJIAN LAPORAN TUGAS AKHIR

Pembimbing Laporan Tugas Akhir, memberikan rekomendasi kepada,

Nama Mahasiswa : Nadhila Bazhlina
NIM : 062030701619
Jurusan/Program Studi : Teknik Komputer/DIII-Teknik Komputer
Judul Laporan KP : Aplikasi Verifikasi Pembayaran *Quick Response Code*
Indonesian Standard Berbasis *Android*

Mahasiswa tersebut telah memenuhi syarat dan dapat mengikuti Ujian Laporan Tugas Akhir, pada Tahun Akademik 2023/2024.

Palembang, Juli 2023

Pembimbing I

Pembimbing II

Yulian Mirza, S.T., M.Kom

NIP. 196607121990031003

Ir. Alan Novi Tompunu, S.T., M.T., IPM

NIP. 197611082000031002



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA JURUSAN
TEKNIK KOMPUTER

Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414

Website : www.polsri.ac.id E-mail : info@polsri.ac.id



REVISI UJIAN TUGAS AKHIR

Dosen Penguji : Azwardi, S.T., M.T
Nama Mahasiswa : Nadhila Bazhlina
NIM : 062030701619
Jurusan /Program Studi : DIII Teknik Komputer
Judul LA/ Skripsi : Aplikasi Verifikasi Pembayaran *Quick Response Code Indonesian Standard* Berbasis Android

No	Uraian	Paraf

Palembang, Agustus 2023

Dosen Penguji,

Azwardi, S.T., M.T

NIP. 197005232005011004



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA JURUSAN
TEKNIK KOMPUTER

Jalan Sriwijaya Negara, Palembang 30139. Telp. 0711-353414
Website : www.polsri.ac.id E-mail : info@polsri.ac.id



REVISI UJIAN TUGAS AKHIR

Dosen Penguji : Ir. Ahmad Bahri Joni Malyan, M.Kom
Nama Mahasiswa : Nadhila Bazhlina
NIM : 062030701619
Jurusan /Program Studi : DIII Teknik Komputer
Judul LA/ Skripsi : Aplikasi Verifikasi Pembayaran *Quick Response*
Code Indonesian Standard Berbasis Android

No	Uraian	Paraf
1.	Belum ada verifikasi detail atau salah untuk pembayaran, tambahkan.	

Palembang, Agustus 2023
Dosen Penguji,

Ir. Ahmad Bahri Joni Malyan, M.Kom
NIP. 196007101991031001



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA JURUSAN
TEKNIK KOMPUTER

Jalan Sriwijaya Negara, Palembang 30139. Telp. 0711-353414
Website : www.polsri.ac.id E-mail : info@polsri.ac.id



REVISI UJIAN TUGAS AKHIR

Dosen Penguji : Indarto, S.T., M.Cs
Nama Mahasiswa : Nadhila Bazhlina
NIM : 062030701619
Jurusan /Program Studi : DIII Teknik Komputer
Judul LA/ Skripsi : Aplikasi Verifikasi Pembayaran *Quick Response Code Indonesian Standard* Berbasis *Android*

No	Uraian	Paraf
-	Hapus 2.1 penelitian terdahulu.	
-	Aplikasi belum bisa mem "verifikasi"	
-	bab 3, ikuti metode waterfall di bab 2	
-	Simpulan : Seseorang dan masalah awal	
-	tujuan awal.	

Palembang, 6 Agustus 2023
Dosen Penguji,

Indarto, S.T., M.Cs
NIP. 197307062005011003



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA JURUSAN
TEKNIK KOMPUTER**

Jalan Sriwijaya Negara, Palembang 30139. Telp. 0711-353414
Website : www.polsri.ac.id E-mail : info@polsri.ac.id



REVISI UJIAN TUGAS AKHIR

Dosen Penguji : Ali Firdaus, M.Kom
Nama Mahasiswa : Nadhila Bazhlina
NIM : 062030701619
Jurusan /Program Studi : DIII Teknik Komputer
Judul LA/ Skripsi : Aplikasi Verifikasi Pembayaran *Quick Response*
Code Indonesian Standard Berbasis Android

No	Uraian	Paraf
	<p>Persiapkan Materi Laporan Sesuai dengan Judul :</p>	

Palembang, Agustus 2023
Dosen Penguji,

Ali Firdaus, M.Kom
NIP. 197010112001121001



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA JURUSAN
TEKNIK KOMPUTER

Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414
Website : www.polsri.ac.id E-mail : info@polsri.ac.id



REVISI UJIAN TUGAS AKHIR

Dosen Penguji : Ica Admirani, S.Kom., M.Kom
Nama Mahasiswa : Nadhila Bazhlina
NIM : 062030701619
Jurusan /Program Studi : DIII Teknik Komputer
Judul LA/ Skripsi : Aplikasi Verifikasi Pembayaran *Quick Response Code Indonesian Standard* Berbasis Android

No	Uraian	Paraf
1.	Perbaiki Aplikasi	
2.	Tambahkan pangkaja QR code yg lain	

Palembang, Agustus 2023

Dosen Penguji,

Ica Admirani, S.Kom., M.Kom

NIP. 197903282005012001



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
JURUSAN TEKNIK KOMPUTER
Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414
Website : www.polsri.ac.id E-mail : info@polsri.ac.id



PELAKSANAAN REVISI UJIAN TUGAS AKHIR

Nama Mahasiswa : Nadhila Bazhlina
NIM : 062030701619
Jurusan /Program Studi : DIII Teknik Komputer
Judul LA/ Skripsi : Aplikasi Verifikasi Pembayaran *Quick Response Code Indonesian Standard* Berbasis *Android*

Telah melaksanakan revisi terhadap Laporan Tugas Akhir yang diujikan pada hari Selasa tanggal 08 bulan Agustus tahun 2023. Pelaksanaan revisi terhadap Laporan Tugas Akhir tersebut telah disetujui oleh Dosen Penguji yang memberikan revisi:

No	Komentar	Nama Dosen Penguji	Tanggal/ bulan	Tanda Tangan
1.	<i>Diserius</i>	Azwardi, S.T., M.T	<i>29/8/23</i>	<i>[Signature]</i>
2.	<i>RRR</i>	Ir. A. Bahri Joni Malyan, M.Kom	<i>25/8/23</i>	<i>[Signature]</i>
3.	<i>ok</i>	Indarto, S.T., M.Cs	<i>22/8/23</i>	<i>[Signature]</i>
4.	<i>OK</i>	Ali Firdaus, M.Kom	<i>22/8-23</i>	<i>[Signature]</i>
5.	<i>ACC</i>	Ica Admirani, S.Kom., M.Kom	<i>29/8-2023</i>	<i>[Signature]</i>

Palembang, Agustus 2023
Ketua Penguji,

Azwardi, S.T., M.T
NIP. 197005232005011004