

**DOKUMENTASI MEMBUAT APLIKASI MOBILE
MENGUNAKAN REACT NATIVE**



Disusun oleh:

Adi Mulyana

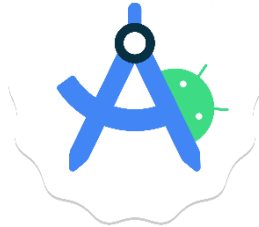
562020122024

**PROGRAM STUDI TEKNIK KOMPUTER
FAKULTAS TEKNIK
UNIVERSITAS WIRALODRA
INDRAMAYU**

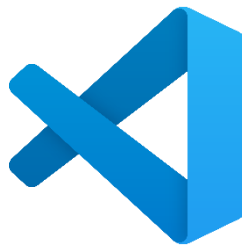
2025

Dokumentasi pembuatan Aplikasi Mobile menggunakan React Native

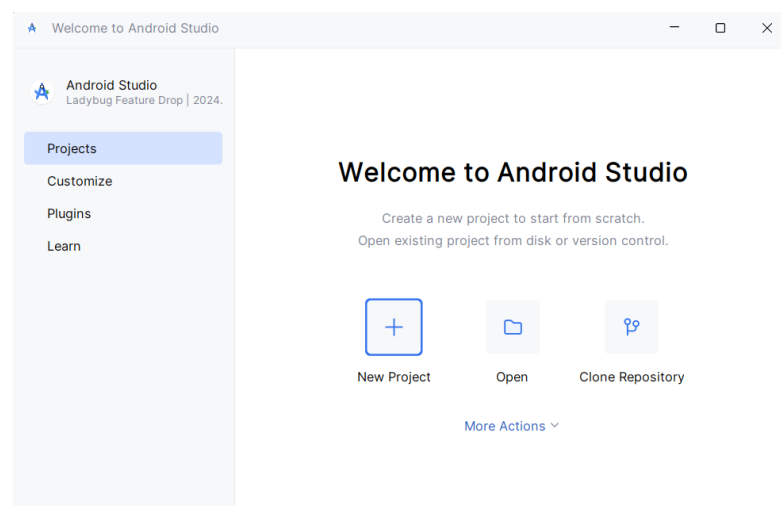
- Siapkan software yang diperlukan untuk membuat aplikasi mobile seperti:
- Android Studio



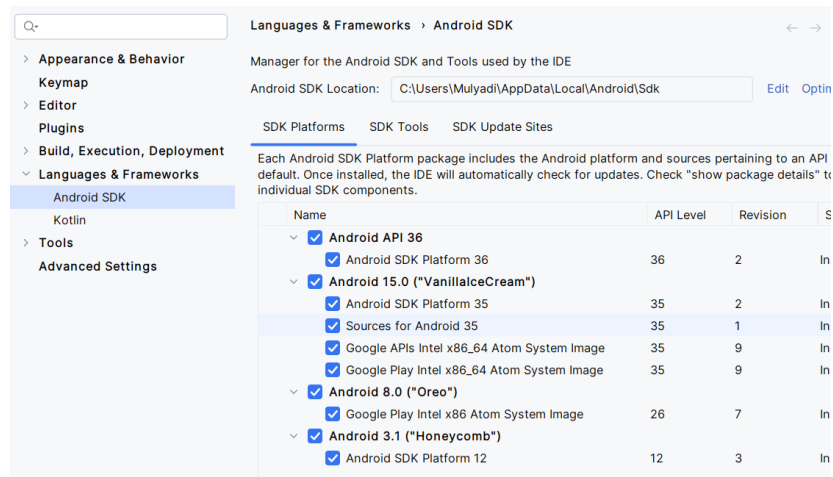
- Visual Studio Code



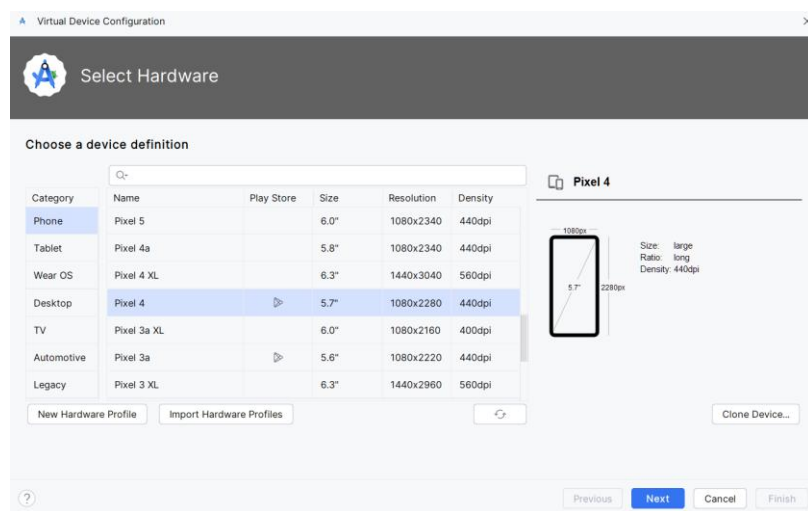
- Setelah software terinstall, langkah selanjutnya adalah Konfigurasi pada Android Studio yang digunakan sebagai emulator



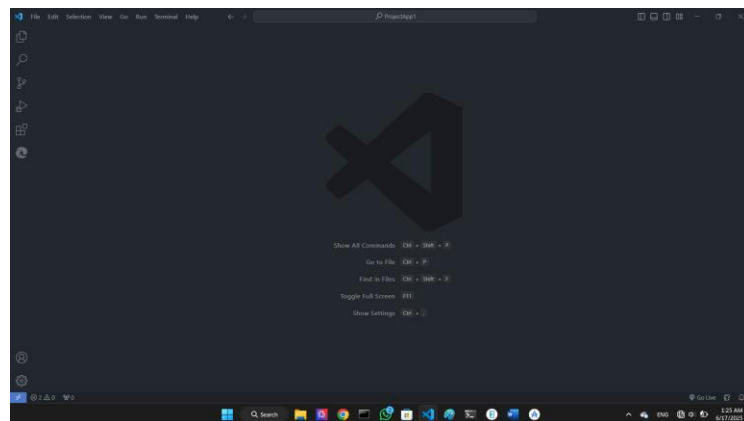
- Install Beberapa komponen di SDK Manager
Temasuk SDK Platform dan SDK Tools



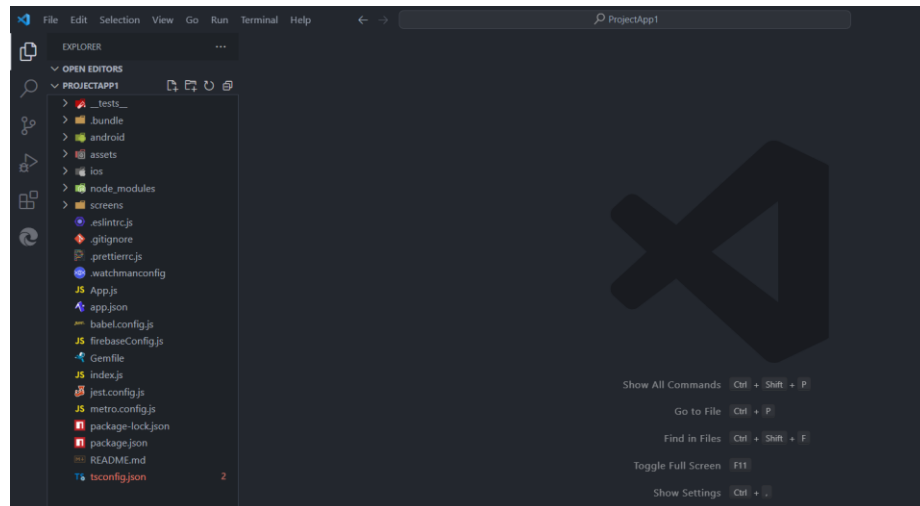
- Install Virtual Device



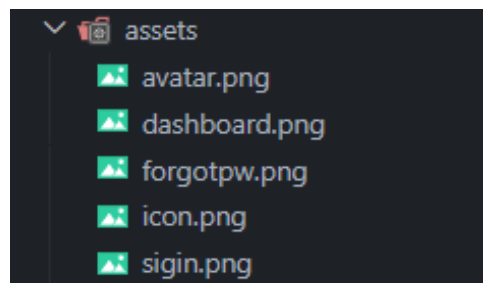
- Buka Visual Studio Code



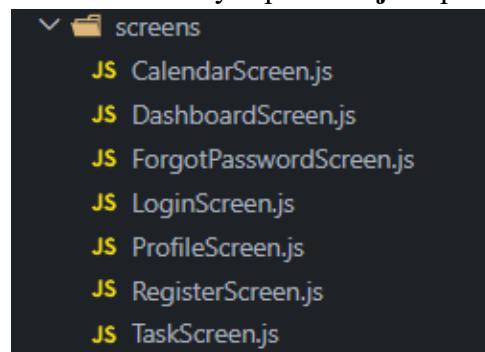
- Buka folder "ProjectApp1"



- Membuat folder *Asset* yang berisikan gambar



- Membuat folder *Screen* untuk menyimpan file **.js** tiap screen nya



- Jalankan **npx react-native start** untuk memulai metro server localhost
Atau **npx react-native start --reset-cache** untuk memulai metro server localhost dengan mereset cache nya

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Welcome to React Native v0.79
Starting dev server on http://localhost:8081

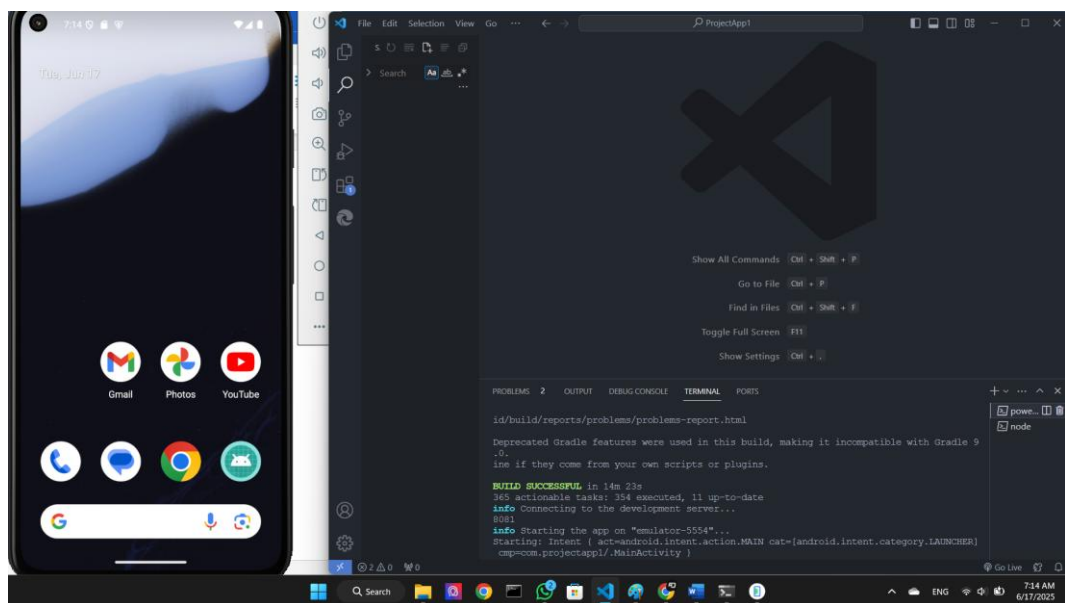
  Welcome to Metro v0.82.3
  Fast - Scalable - Integrated

WARN the transform cache was reset.

INFO Dev server ready. Press Ctrl+C to exit.
INFO Key commands available:

x - reload app(s)
d - open Dev Menu
j - open DevTools
```

- Setelah jalankan metro localhost, buka new terminal lalu masukkan perintah **npx react-native run-android** untuk menjalankan emulator Android



- Membuat file **App.js** yang akan digunakan sebagai Navigasi antar Screen

```

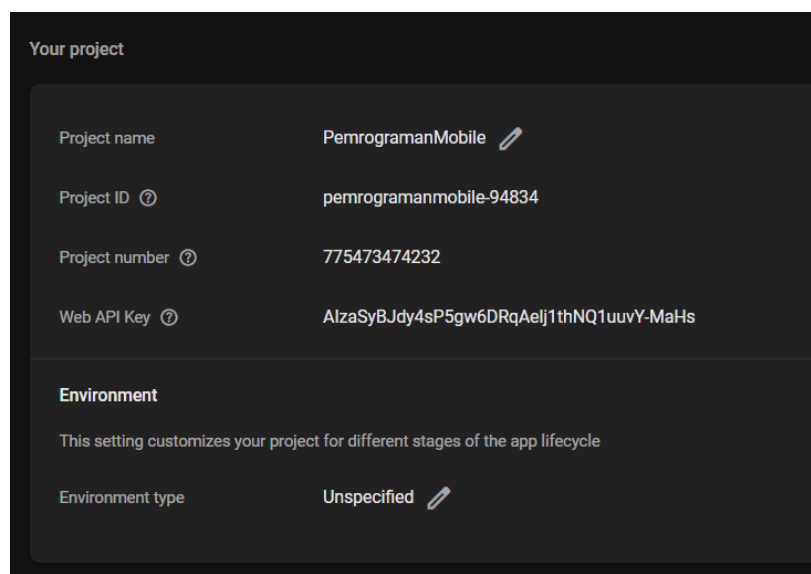
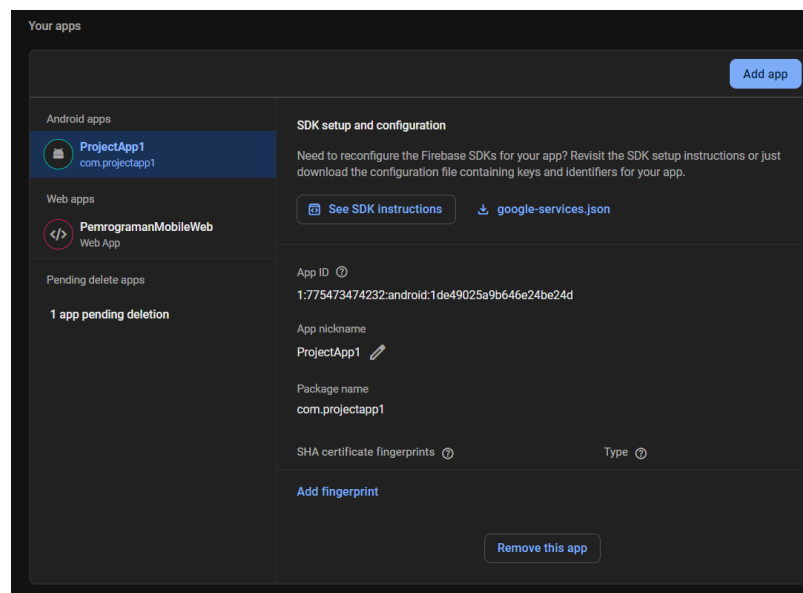
1 // App.js
2
3 import React, {useState, useEffect} from 'react';
4 import {NavigationContainer} from '@react-navigation/native';
5 import {createNativeStackNavigator} from '@react-navigation/native-stack';
6 import auth from '@react-native-firebase/auth';
7
8 // Screens
9 import LoginScreen from './screens/LoginScreen';
10 import RegisterScreen from './screens/RegisterScreen';
11 import ForgotPasswordScreen from './screens/ForgotPasswordScreen';
12 import DashboardScreen from './screens/DashboardScreen';
13 import ProfileScreen from './screens/ProfileScreen';
14 import CalendarScreen from './screens/CalendarScreen';
15 import TasksScreen from './screens/TasksScreen';
16
17 const Stack = createNativeStackNavigator();
18
19 const App = () => {
20   const [user, setUser] = useState(null);
21   const [initializing, setInitializing] = useState(true);
22
23   useEffect(() => {
24     const unsubscribe = auth().onAuthStateChanged(currentUser => {
25       setUser(currentUser);
26       if (initializing) setInitializing(false);
27     });
28
29     return unsubscribe; // Bersihkan listener saat unmount
30   }, [initializing]);
31
32   if (initializing) return null; // Atau bisa tampilkan splash screen
33
34   return (
35     <NavigationContainer>
36       <Stack.Navigator screenOptions={{headerShown: false}}>
37         {user ? (
38           <>
39             <Stack.Screen name="Dashboard" component={DashboardScreen} />
40             <Stack.Screen name="Profile" component={ProfileScreen} />
41             <Stack.Screen name="Calendar" component={CalendarScreen} />
42             <Stack.Screen name="Tasks" component={TasksScreen} />
43           </>
44         ) : (
45           <>
46             <Stack.Screen name="Login" component={LoginScreen} />
47             <Stack.Screen name="Register" component={RegisterScreen} />
48             <Stack.Screen
49               name="ForgotPassword"
50               component={ForgotPasswordScreen}
51             />
49             </>
52           </>
53         )}
54       </Stack.Navigator>
55     </NavigationContainer>
56   );
57 };
58
59 export default App;
60

```

- Membuat file **firebaseConfig.js** yang akan digunakan sebagai koneksi ke Database Firebase untuk menyimpan data user

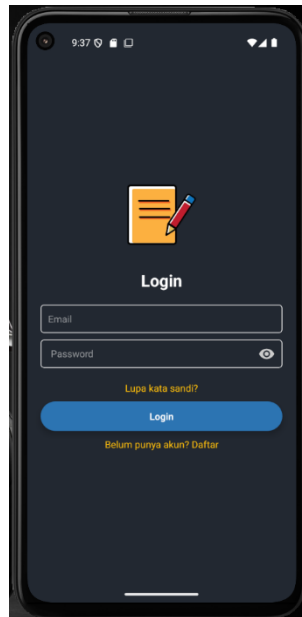
```
JS firebaseConfig.js X
JS firebaseConfig.js
1 // firebaseConfig.js
2 import auth from '@react-native-firebase/auth';
3 import firebase from '@react-native-firebase/app';
4
5
6 export {firebase, auth};
7
```

- Konfigurasi di Firebase



TAMPILAN FORM LOGIN

Disini merupakan tahapan membuat bagian form login



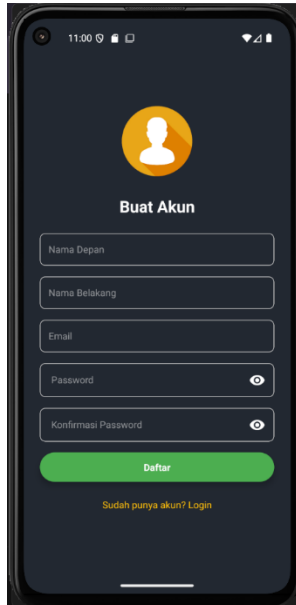
CODE FORM LOGIN

Berikut merupakan bagian Source Code dari Form Login

```
13 import auth from '@react-native-firebase/auth';
14 import Icon from 'react-native-vector-icons/MaterialCommunityIcons';
15
16 const LoginScreen = ({navigation}) => {
17   const [email, setEmail] = useState('');
18   const [password, setPassword] = useState('');
19   const [showPassword, setShowPassword] = useState(false);
20
21   const handleLogin = () => {
22     if (!email || !password) {
23       Alert.alert('Error', 'Email dan password wajib diisi');
24       return;
25     }
26
27     auth()
28       .signInWithEmailAndPassword(email, password)
29       .then(() => {
30         Alert.alert('Sukses', 'Login berhasil');
31         navigation.navigate('Dashboard');
32       })
33       .catch(error => {
34         Alert.alert('Login Gagal', error.message);
35       });
36   };
37 }
```


TAMPILAN REGISTER

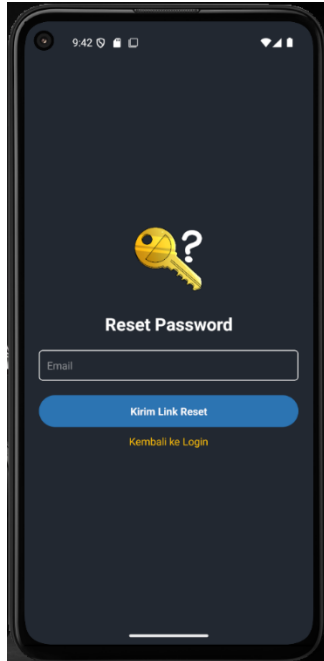
- Berikut merupakan tampilan register/buat akun terdapat 5 placeholder, diantaranya Nama depan, Nama belakang, Email, Password, dan Konfirmasi Password beserta code program nya



```
1  const RegisterScreen = ({navigation}) => {
2    const [firstName, setFirstName] = useState("");
3    const [lastName, setLastName] = useState("");
4    const [email, setEmail] = useState("");
5    const [password, setPassword] = useState("");
6    const [confirmPassword, setConfirmPassword] = useState("");
7
8    const [showPassword, setShowPassword] = useState(false);
9    const [showConfirmPassword, setShowConfirmPassword] = useState(false);
10
11    const isValidPassword = password => {
12      const regex = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d){8,}$/;
13      return regex.test(password);
14    };
15
16    const handleRegister = () => {
17      if (!firstName || !lastName || !email || !password || !confirmPassword) {
18        Alert.alert('Error', 'Semua field wajib diisi :');
19        return;
20      }
21
22      if (!isValidPassword(password)) {
23        Alert.alert(
24          'Password tidak valid',
25          'Password harus minimal 8 karakter, mengandung huruf besar, huruf kecil, dan angka.',
26        );
27        return;
28      }
29
30      if (password !== confirmPassword) {
31        Alert.alert('Error', 'Konfirmasi password tidak cocok');
32        return;
33      }
34
35      auth()
36        .createUserWithEmailAndPassword(email, password)
37        .then(userCredential => {
38          const user = userCredential.user;
39          return user.updateProfile({
40            displayName: `${firstName} ${lastName}`,
41          });
42        })
43        .then(() => {
44          Alert.alert('Berhasil', 'Akun berhasil dibuat');
45          navigation.navigate('Login');
46        })
47        .catch(error => {
48          console.log(error);
49          Alert.alert('Gagal membuat akun', error.message);
50        });
51    };
52  };
53
```

TAMPILAN RESET PASSWORD

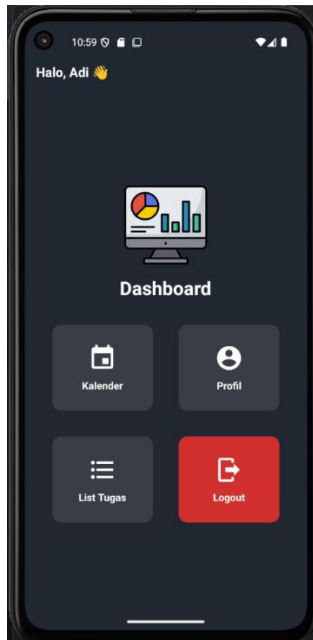
- Berikut merupakan tampilan untuk mereset kata sandi akun beserta code program nya



```
1  const ForgotPasswordScreen = ({navigation}) => {
2    const [email, setEmail] = useState("");
3
4    const handleReset = () => {
5      if (!email) {
6        Alert.alert('Error', 'Email wajib diisi');
7        return;
8      }
9
10     auth()
11       .sendPasswordResetEmail(email)
12       .then(() => {
13         Alert.alert('Berhasil', 'Cek email kamu untuk reset kata sandi');
14         navigation.goBack();
15       })
16       .catch(error => {
17         Alert.alert('Error', error.message);
18       });
19   };
```

TAMPILAN DASHBOARD

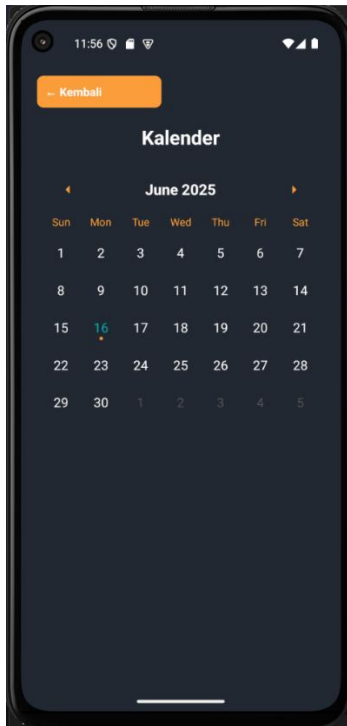
- Ini merupakan tampilan dashboard untuk mengakses beberapa screen seperti kalender, profil dan list tugas



```
1  const Dashboard = ({navigation}) => {
2    const [firstName, setFirstName] = useState("");
3
4    useEffect(() => {
5      const user = auth().currentUser;
6      if (user && user.displayName) {
7        const displayName = user.displayName.split(' ')[0];
8        setFirstName(displayName);
9      }
10   }, []);
11
12   const handleLogout = () => {
13     auth().signOut(); // biarkan navigasi dihandle dari App.js saat onAuthStateChanged
14   };
15 }
```

TAMPILAN KALENDER

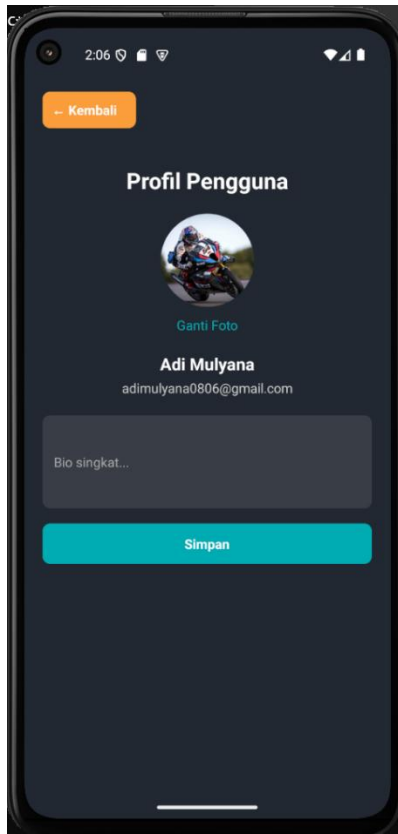
- Berikut merupakan tampilan dalam menu kalender untuk melihat tanggal bulan dan tahun, beserta kode program nya.



```
1  const CalendarScreen = ({navigation}) => {
2    const handleDayPress = day => {
3      alert(`Tanggal dipilih: ${day.dateString}`);
4    };
5
6    return (
7      <SafeAreaView style={styles.safeArea}>
8        <StatusBar barStyle="light-content" backgroundColor="#222831" />
9        <View style={styles.container}>
10         <TouchableOpacity
11           onPress={() => navigation.navigate('Dashboard')}
12           style={styles.backButton}>
13           <Text style={styles.backText}><- Kembali</Text>
14         </TouchableOpacity>
15
16         <Text style={styles.title}>Kalender</Text>
17
18         <Calendar
19           onDayPress={handleDayPress}
20           theme={{
21             backgroundColor: '#222831',
22             calendarBackground: '#222831',
23             textSectionTitleColor: '#FB9E3A',
24             selectedDayBackgroundColor: '#00ADB5',
25             selectedDayTextColor: '#ffffff',
26             todayTextColor: '#00ADB5',
27             dayTextColor: '#ffffff',
28             textDisabledColor: '#555',
29             arrowColor: '#FB9E3A',
30             monthTextColor: '#ffffff',
31             textMonthFontWeight: 'bold',
32             textDayFontSize: 16,
33             textMonthFontSize: 18,
34             textDayHeaderFontWeight: 'bold',
35           }}
36           markedDates={{
37             '2025-06-16': {marked: true, dotColor: '#FB9E3A'},
38           }}
39         />
40       </View>
41     </SafeAreaView>
42   );
43   };
```

TAMPILAN PROFIL

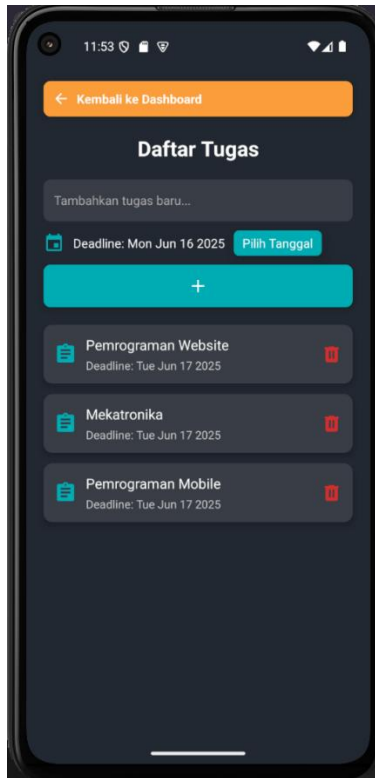
- Disini merupakan tampilan profil yang berisi gambar user, nama user, email user serta bio , beserta kode program nya



```
1  import {launchImageLibrary} from 'react-native-image-picker';
2  import auth from '@react-native-firebase/auth';
3  import firestore from '@react-native-firebase/firestore';
4
5  const ProfileScreen = ({navigation}) => {
6    const user = auth().currentUser;
7
8    const [photo, setPhoto] = useState(user?.photoURL || null);
9    const [bio, setBio] = useState("");
10   const name = user?.displayName || "";
11   const email = user?.email || "";
12
13   const handleChoosePhoto = async () => {
14     const result = await launchImageLibrary({mediaType: 'photo'});
15
16     if(!result.didCancel && result.assets && result.assets.length > 0) {
17       const selectedImage = result.assets[0];
18       const {uri} = selectedImage;
19
20       setPhoto(uri); // Menampilkan preview foto lokal
21       Alert.alert('Berhasil', 'Foto profil berhasil diperbarui (sementara).');
22     }
23   };
24
25   const handleSave = async () => {
26     try {
27       await firestore().collection('users').doc(user.uid).set(
28         {
29           bio: bio,
30         },
31         {merge: true},
32       );
33
34       Alert.alert('Tersimpan', 'Perubahan profil berhasil disimpan');
35     } catch (error) {
36       console.error('Gagal menyimpan bio:', error);
37       Alert.alert('Gagal', 'Terjadi kesalahan saat menyimpan bio');
38     }
39   };
40 }
```

TAMPILAN TASK LIST/TUGAS

- Berikut merupakan tampilan list tugas yang berfungsi untuk mencatat tugas tugas yang akan dikerjakan beserta tanggal deadline yang bisa disesuaikan sesuai kebutuhan beserta Kode Program nya



```
1  const TasksScreen = ({navigation}) => {
2    const [tasks, setTasks] = useState([]);
3    const [newTask, setNewTask] = useState("");
4    const [deadline, setDeadline] = useState(new Date());
5    const [openDatePicker, setOpenDatePicker] = useState(false);
6
7    const handleAddTask = () => {
8      if (newTask.trim() === "") {
9        Alert.alert('Error', 'Tugas tidak boleh kosong');
10       return;
11     }
12
13     const task = {
14       text: newTask,
15       deadline: deadline.toString(),
16     };
17
18     setTasks(prevTasks => [...prevTasks, task]);
19     setNewTask("");
20     setDeadline(new Date());
21   };
22
23   const handleDeleteTask = index => {
24     Alert.alert('Konfirmasi', 'Yakin ingin menghapus tugas ini?', [
25       {text: 'Batal'},
26       {
27         text: 'Hapus',
28         onPress: () => {
29           const updatedTasks = tasks.filter((_, i) => i !== index);
30           setTasks(updatedTasks);
31         },
32       },
33     ]);
34   };
35 }
```