LAMPIRAN

Lampiran – 1 Splashscreen.js

```
import React, {Component} from 're
act';
import {
  View,
  Text,
  ImageBackground,
  PermissionsAndroid,
  BackHandler,
  Alert,
} from 'react-native';

import {StackActions} from '@react-
navigation/native';
import Geolocation from 'react-
native-geolocation-service';
import {notifikasi} from '../compone
nt/Notifikasi';

var bg = require('../img/Splash/bg.pn
g');

class SplashScreen extends Compone
nt {
  constructor(props) {
    super(props);
    this.state = {};
    notifikasi.configure();
    notifikasi.buatChannel('17', functi
on (created) {
      console.log(`createChannel retur
ned = '${created}'`);
    });
  }
  componentDidMount() {
    this.getLocation();
  }

  navToHome(latitude, longitude) {
    this.getPosisi(latitude, longitude).t
hen(data => {
      // console.log(data);
      // setPosisi(
      //   data.results[1].address_compo
nents[2].short_name +
      //     ', ' +
      //     data.results[1].address_comp
onents[3].short_name +
      //     ', ' +
      //     data.results[1].address_comp
onents[4].short_name,
      // );
      setTimeout(() => {
        this.props.navigation.dispatch(
          StackActions.replace('Home',
{
            latitude: latitude,
            longitude: longitude,
            posisi:
              data.results[1].address_com
ponents[2].short_name +
              ', ' +
              data.results[1].address_com
ponents[3].short_name +
              ', ' +
              data.results[1].address_com
ponents[4].short_name,
          }),
        );
      }, 500);
    });
  }

  setNotifShalat(shalat, shalatArr, tgl,
 bln, thn) {
    let id_shalat =
      tgl.toString() + bln.toString() + th
n.toString() + shalatArr.id;
    id_shalat = parseInt(id_shalat);
    let wkt_shalat = shalatArr.wkt;
    let dateNow = new Date();
    if (wkt_shalat >= dateNow) {
      console.log(shalat + ' = ' + id_sha
lat + ' = ' + wkt_shalat);
      // console.log(wkt_shalat);
```

```
      notifikasi.kirimNotifikasiJadwal(
        id_shalat,
        '17',
        'Shalat ' + shalat,
        'Sudah masuk waktu Shalat ' + s
halat + '!',
        wkt_shalat,
      );
    }
  }

  getPosisi(lat, long) {
    let urlposisi =
      'https://maps.googleapis.com/ma
ps/api/geocode/json?address=' +
      lat +
      ',' +
      long +
      'dankey=AIzaSyA1MgLuZuyqR
_OGY3ob3M52N46TDBRI_9k';
    console.log(urlposisi);
    return fetch(urlposisi)
      .then(response => response.json()
)
      .then(responseJson => {
        return responseJson;
      })
      .catch(error => {
        Alert.alert('Perhatian!', 'Gagal L
oad Posisi!', [
          {text: 'Keluar', onPress: () =>
BackHandler.exitApp()},
        ]);
      });
  }

  getJadwalShalatDay(dt) {
    let d = parseInt(dt.date.gregorian.d
ay);
    let m = parseInt(dt.date.gregorian.
month.number);
    let y = parseInt(dt.date.gregorian.y
ear);
    let shalatDhuhr = dt.timings.Dhuh
r;
    if (dt.timings.Dhuhr.includes(' ')) {
      shalatDhuhr = dt.timings.Dhuhr.s
plit(' ')[0];
    }
    let dateDhuhr = new Date(m + '/' +
 d + '/' + y + ' ' + shalatDhuhr);
    // let dateDhuhr = new Date(new
Date().getTime()+5000);
    let shalatAsr = dt.timings.Asr;
    if (dt.timings.Asr.includes(' ')) {
      shalatAsr = dt.timings.Asr.split(' '
)[0];
    }
    let dateAsr = new Date(m + '/' + d
+ '/' + y + ' ' + shalatAsr);
    let shalatMaghrib = dt.timings.Ma
ghrib;
    if (dt.timings.Maghrib.includes(' ')
) {
      shalatMaghrib = dt.timings.Magh
rib.split(' ')[0];
    }
    let dateMaghrib = new Date(m + '/
' + d + '/' + y + ' ' + shalatMaghrib);
    let shalatIsha = dt.timings.Isha;
    if (dt.timings.Isha.includes(' ')) {
      shalatIsha = dt.timings.Isha.split('
 ')[0];
    }
    let dateIsha = new Date(m + '/' + d
 + '/' + y + ' ' + shalatIsha);
    let shalatFajr = dt.timings.Fajr;
    if (dt.timings.Fajr.includes(' ')) {
      shalatFajr = dt.timings.Fajr.split('
 ')[0];
    }
    let dateFajr = new Date(m + '/' + d
+ '/' + y + ' ' + shalatFajr);
    let jadwal = {};
    jadwal['Dhuzur'] = {};
    jadwal['Dhuzur']['id'] = '1';
    jadwal['Dhuzur']['wkt'] = dateDhu
hr;
    jadwal['Ashar'] = {};
    jadwal['Ashar']['id'] = '2';
    jadwal['Ashar']['wkt'] = dateAsr;
    jadwal['Magrib'] = {};
```

```
    jadwal['Magrib']['id'] = '3';
    jadwal['Magrib']['wkt'] = dateMag
hrib;
    jadwal['Isya'] = {};
    jadwal['Isya']['id'] = '4';
    jadwal['Isya']['wkt'] = dateIsha;
    jadwal['Subuh'] = {};
    jadwal['Subuh']['id'] = '5';
    jadwal['Subuh']['wkt'] = dateFajr;
    return jadwal;
  }

  getApiJadwalShalat(lat, long, mont
hx, year) {
    let month = monthx;
    if (month > 12) {
      month = 1;
    }
    let urlapijadwal =
      'https://api.aladhan.com/v1/calen
dar?latitude=' +
      lat +
      'danlongitude=' +
      long +
      'danmethod=2danmonth=' +
      month +
      'danyear=' +
      year;
    // console.log(urlapijadwal);
    return fetch(urlapijadwal)
      .then(response => response.json()
)
      .then(responseJson => {
        if (responseJson.code == 200) {
          if (responseJson.status == 'OK'
) {
            return responseJson;
          } else {
            Alert.alert('Perhatian!', 'Gagal
 Load Jadwal Shalat 2!', [
              {text: 'Keluar', onPress: () =
> BackHandler.exitApp()},
            ]);
          }
        }
      })
      .catch(error => {
        Alert.alert('Perhatian!', 'Gagal L
oad Jadwal Shalat!', [
          {text: 'Keluar', onPress: () =>
BackHandler.exitApp()},
        ]);
      });
  }

  getLocation = async () => {
    await PermissionsAndroid.request
(
      PermissionsAndroid.PERMISSI
ONS.ACCESS_FINE_LOCATION,
    );

    if (
      await PermissionsAndroid.check(
        PermissionsAndroid.PERMISSI
ONS.ACCESS_FINE_LOCATION,
      )
    ) {
      Geolocation.getCurrentPosition(
        position => {
          const {latitude, longitude} = p
osition.coords;
          console.log(position);
          console.log(latitude, longitude)
;

          this.getApiJadwalShalat(
            latitude,
            longitude,
            new Date().getMonth() + 1,
            new Date().getFullYear(),
          ).then(data => {
            let today = new Date().getDat
e();
            var notifTgl = 5;
            data.data.forEach(dtx => {
              let tgl = parseInt(dtx.date.gr
egorian.day);
              let bln = parseInt(dtx.date.gr
egorian.month.number);
              let thn = parseInt(dtx.date.gr
egorian.year);
```

```
          if (tgl >= today) {
            if (notifTgl > 0) {
              let jadwalShalat = this.get
JadwalShalatDay(dtx);
              console.log(
                tgl.toString() +
                  '/' +
                  bln.toString() +
                  '/' +
                  thn.toString(),
              );
              Object.keys(jadwalShalat
).forEach(shalat => {
                this.setNotifShalat(
                  shalat,
                  jadwalShalat[shalat],
                  tgl,
                  bln,
                  thn,
                );
              });
              notifTgl--;
            }
          }
        });

        if (notifTgl > 0) {
          console.log('notifTgl:');
          console.log(notifTgl);
          let m = new Date().getMont
h() + 2;
          let y = new Date().getFullYe
ar();
          if (m > 12) {
            m = 1;
            y = y + 1;
          }

          this.getApiJadwalShalat(lati
tude, longitude, m, y).then(data2 =>
{
            let notifTgl2 = notifTgl;
            data2.data.forEach(dtx2 =>
 {
              let tgl2 = parseInt(dtx2.da
te.gregorian.day);
              let bln2 = parseInt(dtx2.d
ate.gregorian.month.number);
              let thn2 = parseInt(dtx2.d
ate.gregorian.year);
              if (tgl2 <= notifTgl2) {
                let jadwalShalat2 = this.
getJadwalShalatDay(dtx2);
                // console.log(
                //   tgl2.toString() +
                //     '/' +
                //     bln2.toString() +
                //     '/' +
                //     thn2.toString(),
                // );
                Object.keys(jadwalShala
t2).forEach(shalat2 => {
                  this.setNotifShalat(
                    shalat2,
                    jadwalShalat2[shalat2
],
                    tgl2,
                    bln2,
                    thn2,
                  );
                });

                notifTgl--;
              }
            });
            console.log('SELESAI = '
+ notifTgl.toString());
            if (notifTgl == 0) {
              this.navToHome(latitude,
longitude);
            }
          });
        } else if (notifTgl == 0) {
          this.navToHome(latitude, lo
ngitude);
        } else {
          Alert.alert('Perhatian!', 'Gag
al Set Notifikasi!', [
            {text: 'Keluar', onPress: ()
=> BackHandler.exitApp()},
          ]);
        }
```

```
        });
      },
      error => {
        console.log(error.code, error.message);
        Alert.alert('Perhatian!', 'Gagal Load Lokasi!', [
          {text: 'Keluar', onPress: () => BackHandler.exitApp()},
        ]);
      },
      {enableHighAccuracy: true, timeout: 15000, maximumAge: 10000},
    );
  } else {
    Alert.alert('Perhatian!', 'Izin dibutuhkan untuk menjalankan aplikasi!', [
      {text: 'Keluar', onPress: () => BackHandler.exitApp()},
    ]);
  }
 };

 render() {
  return (
    <ImageBackground source={bg} style={{height: '100%', width: '100%'}} />
  );
 }
}

export default SplashScreen;
```

Lampiran – 2 HomeScreen.js

```
import React, {useEffect, useState} f
rom 'react';
import {
  View,
  StyleSheet,
  Text,
  Image,
  TouchableOpacity,
  ImageBackground,
  SafeAreaView,
  FlatList,
  PermissionsAndroid,
  ActivityIndicator,
  BackHandler,
  Alert,
} from 'react-native';
import {createBottomTabNavigator}
 from '@react-navigation/bottom-
tabs';
import base64 from 'react-native-
base64';
import Geolocation from 'react-
native-geolocation-service';

var user = require('../img/Home/user.
png');
var bg_masjid = require('../img/Hom
e/bg_masjid.png');
var location = require('../img/Home/l
ocation.png');
var tab_exit = require('../img/Home/t
ab_exit.png');
var tab_alquran = require('../img/Ho
me/tab_alquran.png');
var tab_arah_kiblat = require('../img/
Home/tab_arah_kiblat.png');
var tab_jadwal_shalat = require('../im
g/Home/tab_jadwal_shalat.png');
var tab_masjid = require('../img/Hom
e/tab_masjid.png');

var items = [];

function HomeTab(props) {
  const [menu, setMenu] = useState(it
ems);
  const [isLoading, setIsLoading] = u
seState(true);
  const [dt, setDt] = useState(
    ('0' + new Date().getHours()).slice
(-2) +
      '.' +
      ('0' + new Date().getMinutes()).sl
ice(-2),
  );

  useEffect(() => {
    // console.log(props.route.params.
posisi);
    fetch('https://panduan-shalat.it-
97.com/api/getbab')
      .then(response => response.json()
)
      .then(responseJson => {
        if (responseJson.status) {
          let newMenu = [];
          newMenu = [...menu];
          newMenu = [];
          var xx = 0;
          responseJson.data.forEach(dtx
=> {
            newMenu[xx] = {};
            newMenu[xx]['id'] = xx + 1;
            newMenu[xx]['title'] = dtx.ju
dulbab;
            newMenu[xx]['src'] = {
              uri: dtx.iconbab,
            };
            newMenu[xx]['idbab'] = base
64.encode(dtx.idbab);
            xx++;
          });
          setMenu(newMenu);
        }
```

```
      setIsLoading(false);
    });

  let secTimer = setInterval(() => {
    setDt(
      ('0' + new Date().getHours()).sli
ce(-2) +
        ':' +
        ('0' + new Date().getMinutes())
.slice(-2),
    );
  }, 1000);
  return () => clearInterval(secTime
r);
 }, []);

 function renderMenu() {
  if (isLoading) {
    return (
      <View style={styles.container}
>
        <ActivityIndicator size="large
" />
      </View>
    );
  } else {
    if (menu.length > 0) {
      return (
        <SafeAreaView style={styles.
container}>
          <FlatList
            data={menu}
            renderItem={({item}) => (
              <TouchableOpacity
                style={{
                  flex: 1,
                  flexDirection: 'column',
                  margin: 10,
                  backgroundColor: 'white
',
                  justifyContent: 'center',
                  alignItems: 'center',
                  height: 100,
                  borderRadius: 20,
                }}
                onPress={() => {
                  // alert(item.title);
                  props.navigation.navigat
e('Bab', {
                    idbab: item.idbab,
                    title: item.title,
                  });
                }}>
                <Image style={styles.ima
geThumbnail} source={item.src} />
                <Text style={{fontSize: 1
0, textAlign: 'center'}}>
                  {item.title}
                </Text>
              </TouchableOpacity>
            )}
            //Setting the number of colu
mn
            numColumns={3}
            keyExtractor={(item, index)
 => index}
          />
        </SafeAreaView>
      );
    } else {
      return (
        <View style={styles.container
}>
          <Text>Belum ada data.</Tex
t>
        </View>
      );
    }
  }
 }

 return (
  <View
    style={[
      styles.containerx,
      {
        flexDirection: 'column',
        backgroundColor: '#9bd7df',
      },
    ]}>
    <ImageBackground
      source={bg_masjid}
```

```
      style={{flex: 1, backgroundCol
or: 'lightblue', padding: 10}}>
      <View
       style={[
        styles.containerx,
        {
         flexDirection: 'row',
        },
       ]}>
       <View style={{flex: 2}}>
        <Text style={{color: 'white',
fontWeight: 'bold'}}>APLIKASI</T
ext>
        <Text style={{color: 'white',
fontWeight: 'bold', fontSize: 9}}>
         RISALAH TUNTUNAN S
HALAT SECARA LENGKAP
        </Text>
       </View>
       <View style={{flex: 1, alignIte
ms: 'flex-end'}}>
        <TouchableOpacity
         style={{}}
         onPress={() => props.navig
ation.navigate('Profil')}>
          <Image source={user} />
        </TouchableOpacity>
       </View>
      </View>
      <Text style={{color: 'white', fo
ntWeight: 'bold', fontSize: 80}}>
       {dt}
      </Text>
      <Text
       style={{
        color: 'white',
        fontWeight: 'bold',
        position: 'relative',
        top: -10,
        left: 5,
       }}>
       <Image source={location} /> {
props.route.params.posisi}
      </Text>
     </ImageBackground>
     <View style={{flex: 2}}>
      <Text
       style={{
        color: 'white',
        fontWeight: 'bold',
        fontSize: 20,
        padding: 10,
       }}>
       MENU
      </Text>
      {renderMenu()}
     </View>
    </View>
  );
}

const Tab = createBottomTabNaviga
tor();

export default function HomeScreen(
{navigation, route}) {
 const [lat, setLat] = useState(route.p
arams.latitude);
 const [long, setLong] = useState(ro
ute.params.longitude);
 const [posisi, setPosisi] = useState(r
oute.params.posisi);

 // useEffect(() => {
 //   getPosisi().then(data => {
 //     // console.log(data);
 //     setPosisi(
 //       data.results[1].address_compo
nents[2].short_name +
 //         ', ' +
 //         data.results[1].address_comp
onents[3].short_name +
 //         ', ' +
 //         data.results[1].address_comp
onents[4].short_name,
 //     );
 //   });
 //   return () => {};
 // }, []);

 return (
  <Tab.Navigator
```

```
      initialRouteName="Tab1"
      screenOptions={({route}) => ({
        headerShown: false,
        tabBarShowLabel: false,
        tabBarStyle: {
          height: 70,
          //   paddingHorizontal: 5,
          //   paddingTop: 0,
          backgroundColor: '#3bbdd4',
          //   position: 'absolute',
          //   borderTopWidth: 0,
        },
      })}>
      <Tab.Screen
        name="Tab1"
        component={HomeTab}
        initialParams={{posisi: posisi}}
        options={{
          tabBarIcon: ({focused}) => {
            return (
              <TouchableOpacity
                onPress={() => {
                  Alert.alert(
                    'Tutup Aplikasi!',
                    'Yakin ingin menutup A
plikasi?',
                    [
                      // The "Yes" button
                      {
                        text: 'Ya',
                        onPress: () => {
                          BackHandler.exitAp
p();
                        },
                      },
                      // The "No" button
                      // Does nothing but dis
miss the dialog when tapped
                      {
                        text: 'Tidak',
                      },
                    ],
                  );
                }}>
                <Image resizeMode={'cont
ain'} source={tab_exit} />
              </TouchableOpacity>
            );
          },
        }}
        // children={() => <HomeTab p
ropName='propValue' />}
      />
      <Tab.Screen
        name="Tab2"
        component={HomeTab}
        options={{
          tabBarIcon: ({focused}) => {
            return (
              <TouchableOpacity
                onPress={() => {
                  navigation.navigate('Sura
hPendek');
                }}>
                <Image resizeMode={'cont
ain'} source={tab_alquran} />
              </TouchableOpacity>
            );
          },
        }}
        listeners={{
          tabPress: e => {
            e.preventDefault();
            // console.log('tess')
          },
        }}
      />
      <Tab.Screen
        name="Tab3"
        component={HomeTab}
        options={{
          tabBarIcon: ({focused}) => {
            return (
              <TouchableOpacity
                onPress={() => {
                  navigation.navigate('Kom
pas', {posisi: posisi});
                }}>
                <Image resizeMode={'cont
ain'} source={tab_arah_kiblat} />
              </TouchableOpacity>
            );
```

```
        },
      }}
      listeners={{
        tabPress: e => {
          e.preventDefault();
          // console.log('tess')
        },
      }}
    />
    <Tab.Screen
      name="Tab4"
      component={HomeTab}
      options={{
        tabBarIcon: ({focused}) => {
          return (
            <TouchableOpacity
              onPress={() => {
                navigation.navigate('Jadw
alShalat', {
                  lat: lat,
                  long: long,
                  posisi: posisi,
                });
              }}>
              <Image resizeMode={'cont
ain'} source={tab_jadwal_shalat} />
            </TouchableOpacity>
          );
        },
      }}
      listeners={{
        tabPress: e => {
          e.preventDefault();
          // console.log('tess')
        },
      }}
    />
    <Tab.Screen
      name="Tab5"
      component={HomeTab}
      options={{
        tabBarIcon: ({focused}) => {
          return (
            <TouchableOpacity
              onPress={() => {
                navigation.navigate('Mesj
idTerdekat', {
                  lat: lat,
                  long: long,
                });
              }}>
              <Image resizeMode={'cont
ain'} source={tab_masjid} />
            </TouchableOpacity>
          );
        },
      }}
      listeners={{
        tabPress: e => {
          e.preventDefault();
          // console.log('tess')
        },
      }}
    />
  </Tab.Navigator>
);
}

const styles = StyleSheet.create({
  containerx: {
    flex: 1,
    padding: 0,
  },
  container: {
    flex: 1,
    justifyContent: 'center',
    marginTop: 16,
  },
  imageThumbnail: {
    // height: 100,
    width: 50,
    height: 50,
  },
});
```

Lampiran – 3 BabScreen.js

```
import React, {useEffect, useState} f
rom 'react';
import {
  View,
  Text,
  TouchableOpacity,
  SafeAreaView,
  FlatList,
  ActivityIndicator,
  StyleSheet,
  Image,
} from 'react-native';
import base64 from 'react-native-
base64';

var items = [];

export default function PengantarUm
umScreen({navigation, route}) {
  // console.log(route);
  const [menu, setMenu] = useState(it
ems);
  const [isLoading, setIsLoading] = u
seState(true);
  useEffect(() => {
    navigation.setOptions({
      title: route.params.title,
    });
    fetch(
      'https://panduan-shalat.it-
97.com/api/getsubbab/' + route.para
ms.idbab,
    )
      .then(response => response.json()
)
      .then(responseJson => {
        if (responseJson.status) {
          let newMenu = [];
          newMenu = [...menu];
          newMenu = [];
          var xx = 0;
          responseJson.data.forEach(dtx
=> {
            newMenu[xx] = {};
            newMenu[xx]['id'] = xx + 1;
            newMenu[xx]['title'] = dtx.ju
dulsubbab;
            newMenu[xx]['idbab'] = base
64.encode(dtx.idbab);
            newMenu[xx]['idsubbab'] = b
ase64.encode(dtx.idsubbab);
            newMenu[xx]['src'] = {
              uri: dtx.gambarsubbab,
            };
            // console.log(newMenu);
            xx++;
          });
          setMenu(newMenu);
        }
        setIsLoading(false);
      });
  }, []);

  if (isLoading) {
    return (
      <View style={styles.container}>
        <ActivityIndicator size="large"
/>
      </View>
    );
  } else {
    if (menu.length > 0) {
      return (
        <SafeAreaView style={{backgr
oundColor: '#9bd7df', flex: 1}}>
          <FlatList
            data={menu}
            renderItem={({item}) => (
              <TouchableOpacity
                style={{
                  flex: 1,
                  flexDirection: 'column',
                  margin: 10,
```

```
            backgroundColor: '#3bbd
d4',
            justifyContent: 'center',
            alignItems: 'center',
            height: 100,
            borderRadius: 20,
           }}
           onPress={() => {
            navigation.navigate('Mate
ri', {
              idbab: item.idbab,
              idsubbab: item.idsubbab,
              title: item.title,
            });
           }}>
           <View style={{flex: 1, flex
Direction: 'row'}}>
            <View
              style={{
                flex: 1,
                backgroundColor: 'whit
e',
                borderRadius: 20,
                justifyContent: 'center',
                alignItems: 'center',
              }}>
              <Image style={styles.im
ageThumbnail} source={item.src} />
            </View>
            <View style={{flex: 2, ju
stifyContent: 'center'}}>
              <Text
                style={{
                  fontSize: 16,
                  textAlign: 'center',
                  color: 'white',
                  fontWeight: 'bold',
                }}>
                  {item.title}
                </Text>
              </View>
            </View>
          </TouchableOpacity>
         )}
         numColumns={1}
         keyExtractor={(item, index)
=> index}
        />
      </SafeAreaView>
    );
   } else {
    return (
      <View style={styles.container}
>
       <Text>Belum ada data.</Text
>
      </View>
    );
   }
  }
}
const styles = StyleSheet.create({
 container: {
  flex: 1,
  flexDirection: 'column',
  justifyContent: 'center',
  alignItems: 'center',
  backgroundColor: '#9bd7df',
 },
 imageThumbnail: {
  width: '100%',
  height: '100%',
  borderRadius: 20,
 },
});
```

Lampiran – 4 MateriScreen.js

```
import React, {Component} from 're
act';
import {
  ActivityIndicator,
  ScrollView,
  StyleSheet,
  Text,
  View,
} from 'react-native';
import Accordian from '../component
/Accordian';

export default class HukumislamScre
en extends Component {
  constructor({navigation, route}) {
    super();
    this.state = {
      isLoading: true,
      idbab: route.params.idbab,
      idsubbab: route.params.idsubbab,
      menu: [],
    };
    navigation.setOptions({
      title: route.params.title,
    });
  }

  componentDidMount() {
    fetch(
      'https://panduan-shalat.it-
97.com/api/getmateri/' +
        this.state.idbab +
        '/' +
        this.state.idsubbab,
    )
      .then(response => response.json()
)
      .then(responseJson => {
        if (responseJson.status) {
          let newMenu = [];
          newMenu = [...this.state.menu]
;
          newMenu = [];
          var xx = 0;
          responseJson.data.forEach(dtx
=> {
            newMenu[xx] = {};
            newMenu[xx]['id'] = xx + 1;
            newMenu[xx]['title'] = dtx.ju
dulmateri;
            newMenu[xx]['data'] = {};
            newMenu[xx]['data']['html'] =
 dtx.isimateri;
            if (typeof dtx.video !== 'unde
fined' dandan dtx.video !== null) {
              newMenu[xx]['video'] = dtx.
video;
            } else {
              newMenu[xx]['video'] = '';
            }
            console.log(dtx.video);
            xx++;
          });
          this.setState({menu: newMenu
, isLoading: false});
        } else {
          this.setState({isLoading: false}
);
        }
      });
  }

  render() {
    if (this.state.isLoading) {
      return (
        <View style={[styles.container,
styles.horizontal]}>
          <ActivityIndicator size="large
" />
        </View>
      );
    } else {
      if (this.state.menu.length > 0) {
        return (
```

```
        <ScrollView style={styles.con
tainer}>
          {this.renderAccordians()}
        </ScrollView>
      );
    } else {
      return (
        <View style={styles.container
1}>
          <Text>Belum ada data.</Tex
t>
        </View>
      );
    }
  }
 }

 renderAccordians = () => {
  const items = [];
  this.state.menu.map(item => {
   // for (item of this.state.menu) {
   items.push(
    <Accordian
     key={item.id}
     register={this.register.bind(thi
s)}
     title={item.title}
     data={item.data}
     video={item.video}
     expanded={item.exp}
     id={item.id}
    />,
   );
   // }
  });
  return items;
 };

 register(aa, expp) {
  let menu = [...this.state.menu];
  let ind = aa - 1;
  var n = 0;
  this.state.menu.map(item => {
   // for (item of this.state.menu) {
   let mn_x = {...menu[n]};
   mn_x.exp = false;
   menu[n] = mn_x;
   n++;
   // }
  });
  let mn = {...menu[ind]};
  mn.exp = expp;
  menu[ind] = mn;
  this.setState({menu: menu});
 }
}

const styles = StyleSheet.create({
 container: {
  flex: 1,
  backgroundColor: '#9bd7df',
 },
 container1: {
  flex: 1,
  backgroundColor: '#9bd7df',
  justifyContent: 'center',
  alignItems: 'center',
 },
 horizontal: {
  flexDirection: 'row',
  justifyContent: 'space-around',
  padding: 10,
 },
});
```

Lampiran – 5 SurahPendekScreen.js

```
import React, {Component} from 're
act';
import {
  ActivityIndicator,
  Button,
  LogBox,
  ScrollView,
  StyleSheet,
  Text,
  View,
} from 'react-native';
import SoundPlayer from 'react-
native-sound-player';
import Accordian2 from '../compone
nt/Accordian2';
LogBox.ignoreLogs([
  'Non-
serializable values were found in the
navigation state',
]);

const customData = require('../json/j
us-30.json');

export default class SurahPendekScr
een extends Component {
  constructor({navigation, route}) {
    super();
    this.state = {
      isLoading: true,
      menu: [],
      ref: null,
      boxTerbuka: 0,
    };
  }

  componentDidMount() {
    console.log(this.props.route.param
s);
    this.props.route.params.setmnright
('');
    this.setState({menu: customData, i
sLoading: false});
    // this.props.navigation.setOptions
({
    //   headerRight: () => (
    //     <Button
    //       onPress={() => {
    //         try {
    //           SoundPlayer.setVolume(1
);
    //         } catch (error) {}
    //       }}
    //       title="Mute"
    //     />
    //   ),
    // });
  }

  render() {
    if (this.state.isLoading) {
      return (
        <View style={[styles.container,
styles.horizontal]}>
          <ActivityIndicator size="large
" />
        </View>
      );
    } else {
      if (this.state.menu.length > 0) {
        return (
          <ScrollView
            ref={ref => {
              // this.setState({ref: ref});
            }}>
            <View style={styles.containe
r}>{this.renderAccordians()}</View
>
          </ScrollView>
        );
      } else {
        return (
```

```
        <View style={styles.container
1}>
          <Text>Belum ada data.</Tex
t>
        </View>
      );
    }
  }
 }

 renderAccordians = () => {
  const items = [];

  // this.setState({boxTerbuka: 0});
  this.state.menu.map(item => {
   // for (item of this.state.menu) {
   items.push(
    <Accordian2
     key={item.id}
     setmnright={this.props.route.p
arams.setmnright.bind(this)}
     register={this.register.bind(thi
s)}
     setBoxTerbuka={this.setBoxT
erbuka.bind(this)}
     expanded={item.exp}
     id={item.id}
     surah={item}
    />,
   );
   // }
  });
  return items;
 };

 setBoxTerbuka(boxTerbuka) {
  // this.setState({boxTerbuka: this.s
tate.boxTerbuka + boxTerbuka});
  // console.log('boxTerbuka');
  // console.log(this.state.boxTerbuk
a);
 }

 register(aa, expp, datascroll) {
  let menu = [...this.state.menu];
  let ind = aa - 1;
  var n = 0;
  this.state.menu.map(item => {
   // for (item of this.state.menu) {
   let mn_x = {...menu[n]};
   mn_x.exp = false;
   menu[n] = mn_x;
   n++;
   // }
  });
  let mn = {...menu[ind]};
  mn.exp = expp;
  menu[ind] = mn;
  this.setState({menu: menu});

  // this.state.ref.scrollTo({
  //   x: 0,
  //   y: datascroll,
  //   animated: true,
  // });
 }
}

const styles = StyleSheet.create({
 container: {
  flex: 1,
  backgroundColor: '#9bd7df',
 },
 container1: {
  flex: 1,
  backgroundColor: '#9bd7df',
  justifyContent: 'center',
  alignItems: 'center',
 },
 horizontal: {
  flexDirection: 'row',
  justifyContent: 'space-around',
  padding: 10,
 },
});
```

Lampiran – 6 KompasScreen.js

```
import React, {useState, useEffect, u
seRef} from 'react';
import {
  Animated,
  Image,
  ImageBackground,
  LayoutAnimation,
  PermissionsAndroid,
  StyleSheet,
  Text,
  UIManager,
  View,
} from 'react-native';
import CompassHeading from 'react-
native-compass-heading';
import Geolocation from 'react-
native-geolocation-service';

var location = require('../img/Home/l
ocation.png');
var calibration = require('../img/Kom
pas/compass-calibration.gif');

export default function KompasScree
n({navigation, route}) {
  const [compassHeading, setCompas
sHeading] = useState(0);
  const [qiblad, setQiblad] = useState
(0);
  const [isLoading, setIsLoading] = u
seState(true);
  const [posisi, setPosisi] = useState(r
oute.params.posisi);

  useEffect(() => {
    getLocation();
    const degree_update_rate = 3;
    CompassHeading.start(degree_upd
ate_rate, degree => {
      console.log(degree);
      if (degree.accuracy == 1) {
        setCompassHeading(degree.hea
ding);
        setIsLoading(false);
      }
    });

    return () => {
      CompassHeading.stop();
      setCompassHeading(0);
      setQiblad(0);
      setIsLoading(true);
      setPosisi('-');
    };
  }, []);

  function calculate(latitude, longitud
e) {
    const PI = Math.PI;
    let latk = (21.4225 * PI) / 180.0;
    let longk = (39.8264 * PI) / 180.0;
    let phi = (latitude * PI) / 180.0;
    let lambda = (longitude * PI) / 180
.0;
    let qiblad =
      (180.0 / PI) *
      Math.atan2(
        Math.sin(longk - lambda),
        Math.cos(phi) * Math.tan(latk) -
          Math.sin(phi) * Math.cos(long
k - lambda),
      );
    setQiblad(qiblad);
    // this.setState({qiblad});
  }

  async function getLocation() {
    await PermissionsAndroid.request
(
      PermissionsAndroid.PERMISSI
ONS.ACCESS_FINE_LOCATION,
    );
```

```
    if (
      await PermissionsAndroid.check(
        PermissionsAndroid.PERMISSIONS.ACCESS_FINE_LOCATION,
      )
    ) {
      Geolocation.getCurrentPosition(
        position => {
          const {latitude, longitude} = position.coords;
          console.log(latitude, longitude);
          calculate(latitude, longitude);
        },
        error => {
          console.log(error.code, error.message);
          Alert.alert('Perhatian!', 'Gagal Load Lokasi!', [
            {text: 'Keluar', onPress: () => BackHandler.exitApp()},
          ]);
        },
        {enableHighAccuracy: true, timeout: 15000, maximumAge: 10000},
      );
    } else {
      Alert.alert('Perhatian!', 'Izin dibutuhkan untuk menjalankan aplikasi!', [
        {text: 'Keluar', onPress: () => BackHandler.exitApp()},
      ]);
    }
  }

  function renderKompas() {
    if (isLoading) {
      return (
        <View
          style={{
            backgroundColor: '#3bbdd4',
            width: 330,
            height: 330,
            borderRadius: 15,
            justifyContent: 'center',
            alignItems: 'center',
            overflow: 'hidden',
          }}>
          <Image
            source={{
              uri: 'https://calibratecompass.com/images/figure-8-compass-calibration.gif',
            }}
            style={{width: '100%', height: '100%'}}
          />
        </View>
      );
    } else {
      return (
        <View
          style={{
            backgroundColor: '#3bbdd4',
            width: 330,
            height: 330,
            borderRadius: 15,
            justifyContent: 'center',
            alignItems: 'center',
          }}>
          <View style={{width: 250, height: 250}}>
            <ImageBackground
              source={require('../img/Kompas/compass3.png')}
              style={[
                styles.image,
                {transform: [{rotate: `${360 - compassHeading}deg`}]},
              ]}>
              <View
                style={{
                  flex: 1,
                  alignItems: 'center',
                  justifyContent: 'center',
                  transform: [{rotate: `${qiblad}deg`}],
                }}>
                <Image
                  source={require('../img/Kompas/kakbah3.png')}
```

```
              style={{
                flex: 0.7,
                position: 'relative',
                top: -20,
              }}
             />
            </View>
          </ImageBackground>
        </View>
        <Text
          style={{
            color: 'white',
            fontWeight: 'bold',
            marginTop: 10,
          }}>
          <Image source={location} />
{posisi}
        </Text>
      </View>
```

```
      );
    }
  }

  return <View style={styles.containe
r}>{renderKompas()}</View>;
}

const styles = StyleSheet.create({
  image: {width: '100%', flex: 1, align
Self: 'center'},
  container: {
    backgroundColor: '#9bd7df',
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
  },
});
```

Lampiran – 7 JadwalShalatScreen.js

```
import React, {useState, useEffect} f
rom 'react';
import {
  View,
  Text,
  StyleSheet,
  TouchableOpacity,
  Image,
  ActivityIndicator,
} from 'react-native';

var datajadwal = {
  waktu: '00.00',
  shalat: 'SHALAT',
  sisawaktu: '-',
  jadwal: {
    subuh: {
      shalat: 'Subuh',
      waktu: '00.00',
      bg: '#3bbdd4',
    },
    dhuzur: {
      shalat: 'Dhuzur',
      waktu: '00.00',
      bg: '#3bbdd4',
    },
    ashar: {
      shalat: 'Ashar',
      waktu: '00.00',
      bg: '#3bbdd4',
    },
    magrib: {
      shalat: 'Magrib',
      waktu: '00.00',
      bg: '#3bbdd4',
    },
    isya: {
      shalat: 'Isya',
      waktu: '00.00',
      bg: '#3bbdd4',
    },
  },
};
var alarm = require('../img/Jadwal/al
arm.png');
var secTimer = false;

function RenderJadwal(datashalat) {
  return (
    <View
      style={{
        flex: 1,
        flexDirection: 'row',
        backgroundColor: datashalat.bg,
        borderRadius: 15,
        margin: 8,
      }}>
      <View
        style={{
          flex: 4,
          justifyContent: 'center',
          alignItems: 'flex-start',
          paddingLeft: 20,
        }}>
        <Text style={{color: 'white', fo
ntWeight: '600', fontSize: 22}}>
          {datashalat.shalat}
        </Text>
      </View>
      <View style={{flex: 2, justifyCo
ntent: 'center'}}>
        <Text
          style={{
            fontSize: 22,
            textAlign: 'right',
            color: 'white',
            fontWeight: '600',
            paddingRight: 8,
          }}>
          {datashalat.waktu}
        </Text>
      </View>
      <View
        style={{
```

```
        flex: 1,
        justifyContent: 'center',
        alignItems: 'flex-end',
        paddingRight: 20,
       }}>
       <Image source={alarm} />
     </View>
   </View>
  );
}

export default function JadwalShalat
Screen({navigation, route}) {
  const [jadwalShalat, setJadwalShala
t] = useState(datajadwal);
  const [isLoading, setIsLoading] = u
seState(true);
  const [posisi, setPosisi] = useState(r
oute.params.posisi);

  useEffect(() => {
   getApiJadwalShalat().then(data =>
 {
     secTimer = setInterval(() => {
       // console.log('sss');
       updateJadwal(data);
       // let jjj={...jadwalShalat};
       // jjj.waktu='sss';
       // setJadwalShalat(jjj);
     }, 1000);
   });

   return () => {
    if (secTimer !== false) {
      clearInterval(secTimer);
      secTimer = false;
      setIsLoading(true);
      setPosisi('-');
      setJadwalShalat({});
     }
   };
  }, []);

  function updateJadwal(responseJso
n) {
   let today = new Date().getDate();
   let month = new Date().getMonth(
) + 1;
   let year = new Date().getFullYear(
);
   // let lastday = new Date(year, mo
nth, 0).getDate();
   var yyy = {...jadwalShalat};
   var datahariini = false;
   responseJson.data.forEach(dtx =>
{
     let tgl = parseInt(dtx.date.gregori
an.day);
     if (today == tgl) {
       datahariini = dtx;
     }
   });
   // console.log(lastday)
   if (datahariini !== false) {
     var dt = datahariini;
     let dateNow = new Date();
     // let dateNow = new Date('01/11
/2022 16:34');
     let shalatDhuhr = dt.timings.Dhu
hr;
     if (dt.timings.Dhuhr.includes(' '))
{
       shalatDhuhr = dt.timings.Dhuhr
.split(' ')[0];
     }
     let dateDhuhr = new Date(
       month + '/' + today + '/' + year +
 ' ' + shalatDhuhr,
     );
     yyy.jadwal.dhuzur = {
       shalat: 'Dhuzur',
       waktu: shalatDhuhr,
       bg: '#3bbdd4',
     };
     let shalatAsr = dt.timings.Asr;
     if (dt.timings.Asr.includes(' ')) {
       shalatAsr = dt.timings.Asr.split(
' ')[0];
     }
     let dateAsr = new Date(
       month + '/' + today + '/' + year +
 ' ' + shalatAsr,
```

```
    );
    yyy.jadwal.ashar = {
      shalat: 'Ashar',
      waktu: shalatAsr,
      bg: '#3bbdd4',
    };
    let shalatMaghrib = dt.timings.M
aghrib;
    if (dt.timings.Maghrib.includes(' '
)) {
      shalatMaghrib = dt.timings.Mag
hrib.split(' ')[0];
    }
    let dateMaghrib = new Date(
      month + '/' + today + '/' + year +
 ' ' + shalatMaghrib,
    );
    yyy.jadwal.magrib = {
      shalat: 'Maghrib',
      waktu: shalatMaghrib,
      bg: '#3bbdd4',
    };
    let shalatIsha = dt.timings.Isha;
    if (dt.timings.Fajr.includes(' ')) {
      shalatIsha = dt.timings.Isha.split
(' ')[0];
    }
    let dateIsha = new Date(
      month + '/' + today + '/' + year +
 ' ' + shalatIsha,
    );
    yyy.jadwal.isya = {
      shalat: 'Isya',
      waktu: shalatIsha,
      bg: '#3bbdd4',
    };

    let shalatsubuh = dt.timings.Fajr;
    if (dt.timings.Fajr.includes(' ')) {
      shalatsubuh = dt.timings.Fajr.sp
lit(' ')[0];
    }
    let datesubuh = new Date(
      month + '/' + today + '/' + year +
 ' ' + shalatsubuh,
    );
    yyy.jadwal.subuh = {
      shalat: 'Subuh',
      waktu: shalatsubuh,
      bg: '#3bbdd4',
    };
    yyy.waktu =
      ('0' + new Date().getHours()).sli
ce(-2) +
      '.' +
      ('0' + new Date().getMinutes()).
slice(-2);

    let selisihWaktu = 0;
    if (datesubuh >= dateNow) {
      yyy.jadwal.subuh.bg = '#f45f27'
;
      yyy.shalat = yyy.jadwal.subuh.s
halat.toLocaleUpperCase();
      selisihWaktu = datesubuh - date
Now;
    } else if (dateDhuhr >= dateNow)
 {
      yyy.jadwal.dhuzur.bg = '#f45f2
7';
      yyy.shalat = yyy.jadwal.dhuzur.
shalat.toLocaleUpperCase();
      selisihWaktu = dateDhuhr - date
Now;
    } else if (dateAsr >= dateNow) {
      yyy.jadwal.ashar.bg = '#f45f27';
      yyy.shalat = yyy.jadwal.ashar.s
halat.toLocaleUpperCase();
      selisihWaktu = dateAsr - dateN
ow;
    } else if (dateMaghrib >= dateNo
w) {
      yyy.jadwal.magrib.bg = '#f45f2
7';
      yyy.shalat = yyy.jadwal.magrib.
shalat.toLocaleUpperCase();
      selisihWaktu = dateMaghrib - d
ateNow;
    } else if (dateIsha >= dateNow) {
      yyy.jadwal.isya.bg = '#f45f27';
      yyy.shalat = yyy.jadwal.isya.sha
lat.toLocaleUpperCase();
```

```
      selisihWaktu = dateIsha - dateNow;
    } else {
      yyy.shalat = yyy.jadwal.subuh.shalat.toLocaleUpperCase();
    }
    if (selisihWaktu > 0) {
      let selisih = new Date();
      selisih.setTime(selisihWaktu);
      let selisihJam = selisih.getHours() - 8;
      let selisihMenit = selisih.getMinutes();
      yyy.sisawaktu = selisihJam + ' Jam ' + selisihMenit + ' Menit lagi';
      if (selisihJam <= 0) {
        yyy.sisawaktu = selisihMenit + ' Menit lagi';
        if (selisihMenit <= 0) {
          yyy.sisawaktu = 'Masuk Waktu Shalat';
        }
      }
    } else {
      yyy.sisawaktu = '-';
    }
    setJadwalShalat(yyy);
    setIsLoading(false);
  }
 }

 function getApiJadwalShalat() {
  let month = new Date().getMonth() + 1;
  let year = new Date().getFullYear();
  return fetch(
    'https://api.aladhan.com/v1/calendar?latitude=' +
      route.params.lat +
      'danlongitude=' +
      route.params.long +
      'danmethod=2danmonth=' +
      month +
      'danyear=' +
      year,
  )
    .then(response => response.json())
    .then(responseJson => {
      if (responseJson.code == 200) {
        if (responseJson.status == 'OK') {
          return responseJson;
        }
      }
    })
    .catch(error => {
      alert('Gagal!!!');
    });
 }

 if (isLoading) {
  return (
    <View style={styles.container}>
      <ActivityIndicator size="large" />
    </View>
  );
 } else {
  return (
    <View style={styles.container}>
      <View style={{flex: 1, padding: 10}}>
        <View
          style={{
            backgroundColor: '#3bbdd4',
            borderRadius: 15,
            flex: 1,
            flexDirection: 'column',
          }}>
          <View style={{flex: 2, flexDirection: 'row'}}>
            <View
              style={{
                flex: 0.8,
                justifyContent: 'flex-end',
                alignItems: 'flex-end',
              }}>
              <Text
```

```
              style={{color: 'white', fon
tWeight: 'bold', fontSize: 30}}>
                {jadwalShalat.shalat}
              </Text>
              <Text style={{color: 'whit
e', fontSize: 12}}>
                {jadwalShalat.sisawaktu}
              </Text>
            </View>
            <View
              style={{
                flex: 1,
                justifyContent: 'flex-end',
                alignItems: 'flex-end',
                paddingRight: 20,
              }}>
              <Text
                style={{color: 'white', fon
tWeight: 'bold', fontSize: 50}}>
                {jadwalShalat.waktu}
              </Text>
            </View>
          </View>
          <View
            style={{
              flex: 1.3,
              justifyContent: 'center',
              paddingHorizontal: 20,
            }}>
            <Text style={{color: 'white'
}}>
              {posisi}
            </Text>
          </View>
        </View>
      </View>
        <View
          style={{
            flex: 2,
            flexDirection: 'column',
            padding: 2,
            paddingBottom: 10,
          }}>
          {RenderJadwal(jadwalShalat.j
adwal.subuh)}
          {RenderJadwal(jadwalShalat.j
adwal.dhuzur)}
          {RenderJadwal(jadwalShalat.j
adwal.ashar)}
          {RenderJadwal(jadwalShalat.j
adwal.magrib)}
          {RenderJadwal(jadwalShalat.j
adwal.isya)}
        </View>
      </View>
    );
  }
}
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#9bd7df',
    flexDirection: 'column',
    justifyContent: 'center',
  },
  horizontal: {
    flexDirection: 'row',
    justifyContent: 'space-around',
    padding: 10,
  },
});
```

Lampiran – 8 MesjidTerdekatScreen.js

```
import React, {Component} from 're
act';
import {
  Text,
  View,
  StyleSheet,
  Image,
  PermissionsAndroid,
  ActivityIndicator,
  Alert,
  BackHandler,
} from 'react-native';
import MapView, {Marker, PROVI
DER_GOOGLE} from 'react-native-
maps';
import Geolocation from 'react-
native-geolocation-service';

var markermasjid = require('../img/M
aps/marker-masjid.png');

export class MesjidTerdekatScreen e
xtends Component {
  constructor({navigation, route}) {
    super();
    this.state = {
      isLoading: true,
      latitude: route.params.lat,
      longitude: route.params.long,
      mesjid: {},
    };
    // console.log(route.params.lat);
  }

  componentDidMount() {
    this.getLocation();
  }

  getLocation = async () => {
    await PermissionsAndroid.request
(
      PermissionsAndroid.PERMISSI
ONS.ACCESS_FINE_LOCATION,
    );

    if (
      await PermissionsAndroid.check(
        PermissionsAndroid.PERMISSI
ONS.ACCESS_FINE_LOCATION,
      )
    ) {
      Geolocation.getCurrentPosition(
        position => {
          const {latitude, longitude} = p
osition.coords;
          console.log(latitude, longitude)
;
          let urlmesjid =
            'https://maps.googleapis.com/
maps/api/place/nearbysearch/json?lo
cation=' +
            latitude +
            ',' +
            longitude +
            'danradius=1500dantype=mos
quedankeyword=dankey=AIzaSyA1
MgLuZuyqR_OGY3ob3M52N46TD
BRI_9k';
          console.log(urlmesjid);
          fetch(urlmesjid)
            .then(response => response.js
on())
            .then(responseJson => {
              // console.log(responseJson)
;
              if (responseJson.status == 'O
K') {
                this.setState({
                  latitude: latitude,
                  longitude: longitude,
                  isLoading: false,
                  mesjid: responseJson,
                });
```

```
            } else {
              Alert.alert(
                'Perhatian!',
                'Gagal Deteksi Mesjid, St
atus : ' + responseJson.status + '!',
                [{text: 'Keluar', onPress: (
) => BackHandler.exitApp()}],
              );
            }
          });
      },
      error => {
        console.log(error.code, error.m
essage);
        Alert.alert('Perhatian!', 'Gagal
Load Lokasi!', [
          {text: 'Keluar', onPress: () =>
BackHandler.exitApp()},
        ]);
      },
      {enableHighAccuracy: true, tim
eout: 15000, maximumAge: 10000},
    );
  } else {
    Alert.alert('Perhatian!', 'Izin dibut
uhkan untuk menjalankan aplikasi!',
[
      {text: 'Keluar', onPress: () => B
ackHandler.exitApp()},
    ]);
  }
 };

 renderMarkers() {
  return this.state.mesjid.results.map
(item => {
    return (
      <Marker
        key={item.place_id}
        coordinate={{
          latitude: item.geometry.locati
on.lat,
          longitude: item.geometry.loca
tion.lng,
        }}>
        <Text style={{fontWeight: 'bo
ld'}}>{item.name}</Text>
        <Image
          style={{height: 40, width: 27,
 resizeMode: 'stretch'}}
          source={markermasjid}
        />
      </Marker>
    );
  });
 }

 renderMaps() {
  if (this.state.isLoading) {
    return <ActivityIndicator size="l
arge" />;
  } else {
    return (
      <MapView
        style={styles.map}
        // mapType={'satellite'}
        provider={PROVIDER_GOO
GLE}
        initialRegion={{
          latitude: this.state.latitude,
          longitude: this.state.longitude
,
          latitudeDelta: 0.006,
          longitudeDelta: 0.006,
        }}
        customMapStyle={styles.hide
}>
        {this.renderMarkers()}
      </MapView>
    );
  }
 }

 render() {
  return <View style={styles.wrappe
r}>{this.renderMaps()}</View>;
 }
}

const styles = StyleSheet.create({
 wrapper: {
```

```
    ...StyleSheet.absoluteFillObject,
    justifyContent: 'center',
  },
  map: {
    ...StyleSheet.absoluteFillObject,
  },
  hide: [
    {
      featureType: 'poi.place_of_worsh
ip',
      // elementType: 'labels.icon',
      stylers: [{visibility: 'off'}],
    },
  ],
});

export default MesjidTerdekatScreen
;
```

Lampiran – 9 ProfilScreen.js

```
import React from 'react';
import {View, Text, ImageBackgrou
nd, Image} from 'react-native';

var bg_profil = require('../img/Profil/
bg_profil.png');
var profil = require('../img/Profil/user
2.png');

export default function ProfilScreen(
) {
  return (
    <ImageBackground
      source={bg_profil}
      style={{
        flex: 1,
        flexDirection: 'column',
        alignItems: 'center',
        justifyContent: 'center',
      }}>
      <View style={{flex: 1}}>
        <Image source={profil} style={
{marginTop: 20}} />
      </View>
      <View
        style={{
          flex: 3,
          width: '100%',
          paddingHorizontal: 20,
          paddingBottom: 20,
        }}>
        <View
          style={{
            backgroundColor: 'white',
            width: '100%',
            flex: 1,
            position: 'relative',
            top: -20,
            borderRadius: 10,
            justifyContent: 'flex-start',
            padding: 20,
          }}>
            <Text style={{fontSize: 20, f
ontWeight: '600', color:'#00BCD4', t
extAlign:'center', padding: 1}}>
              Judul Skripsi
            </Text>
            <Text style={{fontSize: 20, fo
ntWeight: '800', color:'#00BCD4', te
xtAlign:'center', padding:10, padding
Bottom: 20}}>Pengembangan Aplik
asi Risalah Tuntunan Shalat secara L
engkap</Text>
            <Text style={{fontSize: 16, fo
ntWeight: '600', color:'#00BCD4'}}>
              Nama : Kurniawan AS
            </Text>
            <Text style={{fontSize: 16, fo
ntWeight: '600', color:'#00BCD4'}}>
NIM : 217280064</Text>
            <Text style={{fontSize: 16, fo
ntWeight: '600', color:'#00BCD4',pa
ddingBottom: 20}}>Kls : Teknik Inf
ormatika B</Text>
            <Text style={{fontSize: 16, fo
ntWeight: '600', color:'#00BCD4'}}>
Pembimbing I : Hj. A. Irmayani, S.T.
, M.T</Text>
            <Text style={{fontSize: 16, fo
ntWeight: '600', color:'#00BCD4',pa
ddingBottom: 20}}>Pembimbing II:
Wahyuddin, S.Kom., M.Kom</Text
>
            <Text style={{fontSize: 16, fo
ntWeight: '600', color:'#00BCD4'}}>
Penguji I : Ade Hastuty, S.T., S.Kom
., M.T</Text>
            <Text style={{fontSize: 16, fo
ntWeight: '600', color:'#00BCD4',pa
ddingBottom:40}}>Penguji II: Muh.
Zainal, S.T., M.T</Text>
            <Text style={{fontSize: 16, fo
ntWeight: '800', color:'#00BCD4', te
xtAlign:'center', padding:1}}>PROG
```

```
RAM STUDI TEKNIK INFORMAT
IKA</Text>
        <Text style={{fontSize: 16, fo
ntWeight: '800', color:'#00BCD4', te
xtAlign:'center', padding:1}}>FAKU
LTAS TEKNIK</Text>
        <Text style={{fontSize: 16, fo
ntWeight: '800', color:'#00BCD4', te
xtAlign:'center', padding:1}}>UNIV
ERSITAS MUHAMMADIYAH PA
REPARE</Text>
        <Text style={{fontSize: 16, fo
ntWeight: '800', color:'#00BCD4', te
xtAlign:'center', padding:1}}>2022</
Text>
      </View>
      <Text style={{fontWeight: 'bol
d', fontSize: 18, color: 'white'}}>
        Sumber Data :
      </Text>
      <Text style={{fontSize: 14, col
or: 'white'}}>
        Buku Risalah Tuntunan Shalat
Secara Lengkap
      </Text>
      <Text style={{fontSize: 14, col
or: 'white'}}>Penulis Moh. Rifa'i</T
ext>
      <Text style={{fontSize: 14, col
or: 'white'}}>
        PT. KARYA TOHA PUTRA
Semarang 2020
      </Text>
    </View>
  </ImageBackground>
 );
}
```

Lampiran – 10 navigation.js

```
// In Navigation.js in a new project

import * as React from 'react';
import {NavigationContainer} from '
@react-navigation/native';
import {createNativeStackNavigator
} from '@react-navigation/native-
stack';
import HomeScreen from '../screens/
HomeScreen';
import ProfilScreen from '../screens/
ProfilScreen';
import BabScreen from '../screens/Ba
bScreen';
import SplashScreen from '../screens/
SplashScreen';
import MateriScreen from '../screens/
MateriScreen';
import KompasScreen from '../screen
s/KompasScreen';
import MesjidTerdekatScreen from '.
./screens/MesjidTerdekatScreen';
import JadwalShalatScreen from '../s
creens/JadwalShalatScreen';
import SurahPendekScreen from '../s
creens/SurahPendekScreen';
import {Button} from 'react-native-
vector-icons/MaterialIcons';
import SoundPlayer from 'react-
native-sound-player';
import {Text, TouchableOpacity} fro
m 'react-native';

const Stack = createNativeStackNavi
gator();

function Navigation() {
  const [mnright, setmnright1] = Reac
t.useState('');
  function setaksi(isi) {
    if (isi == 'Pause') {
      // SoundPlayer.setVolume(0);
      SoundPlayer.pause();
      setmnright1('Resume');
    } else if (isi == 'Resume') {
      // SoundPlayer.setVolume(1);
      SoundPlayer.resume();
      setmnright1('Pause');
    }
  }
  function setmnright(isi) {
    if (isi == 'hide') {
      setmnright1('');
    } else {
      setmnright1(isi);
    }
  }
  return (
    <NavigationContainer>
      <Stack.Navigator initialRouteNa
me="Splash">
        <Stack.Screen
          name="Splash"
          options={{headerShown: false
}}
          component={SplashScreen}
        />
        <Stack.Screen
          name="Home"
          options={{headerShown: false
}}
          component={HomeScreen}
        />
        <Stack.Screen
          name="Profil"
          options={{
            title: 'PROFIL',
            headerStyle: {
              backgroundColor: '#3bbdd4'
,
            },
            headerTintColor: '#fff',
          }}
          component={ProfilScreen}
```

```
      />
      <Stack.Screen
        name="Bab"
        options={{
          title: 'BAB',
          headerStyle: {
            backgroundColor: '#3bbdd4'
,
          },
          headerTintColor: '#fff',
        }}
        component={BabScreen}
      />
      <Stack.Screen
        name="Materi"
        options={{
          title: 'Materi',
          headerStyle: {
            backgroundColor: '#3bbdd4'
,
          },
          headerTintColor: '#fff',
        }}
        component={MateriScreen}
      />
      <Stack.Screen
        name="SurahPendek"
        options={{
          title: 'Surah - Surah Pendek',
          headerStyle: {
            backgroundColor: '#3bbdd4'
,
          },
          headerTintColor: '#fff',
          headerRight: () => (
            <TouchableOpacity onPress
={() => setaksi(mnright)}>
              <Text>{mnright}</Text>
            </TouchableOpacity>
          ),
        }}
        component={SurahPendekScr
een}
        initialParams={{setmnright: se
tmnright}}
      />
      <Stack.Screen
        name="Kompas"
        options={{
          title: 'Arah Kiblat',
          headerStyle: {
            backgroundColor: '#3bbdd4'
,
          },
          headerTintColor: '#fff',
        }}
        component={KompasScreen}
      />
      <Stack.Screen
        name="MesjidTerdekat"
        options={{
          title: 'Mesjid Terdekat',
          headerStyle: {
            backgroundColor: '#3bbdd4'
,
          },
          headerTintColor: '#fff',
        }}
        component={MesjidTerdekatS
creen}
      />
      <Stack.Screen
        name="JadwalShalat"
        options={{
          title: 'Jadwal Shalat',
          headerStyle: {
            backgroundColor: '#3bbdd4'
,
          },
          headerTintColor: '#fff',
        }}
        component={JadwalShalatScr
een}
      />
    </Stack.Navigator>
  </NavigationContainer>
 );
}

export default Navigation;
```

Lampiran - 11 login.blade.php

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">
  <title>Login</title>
  <meta content="" name="description">
  <meta content="" name="keywords">
  @include('layouts.header')
</head>
<body>
    <main>
      <div class="container">
        <section class="section register min-vh-100 d-flex flex-column align-items-center justify-content-center py-4">
          <div class="container">
            <div class="row justify-content-center">
              <div class="col-lg-4 col-md-6 d-flex flex-column align-items-center justify-content-center">
                <div class="d-flex justify-content-center py-4">
                  <a href="{{asset('/')}}" class="logo d-flex align-items-center w-auto">
                    <img src="{{asset('/template/NiceAdmin/assets/img/logo.png')}}" alt="">
                    <span class="d-none d-lg-block">{{env('APP_NAME')}}</span>
                  </a>
                </div><!-- End Logo -->
                <div class="card mb-3">
                  <div class="card-body">
                    <div class="pt-4 pb-2">
                      <p class="text-center small">{{Lang::get('bahasa.Masukkan username dan password kamu untuk login!')}}</p>
                    </div>
                    <form class="row g-3 needs-validation" novalidate method="POST" action="{{ route('login') }}">
                      @csrf
                      <div class="col-12">
                        <label for="email" class="form-label">Username</label>
                        <div class="input-group has-validation">
                          <span class="input-group-text" id="inputGroupPrepend">@</span>
                          <input type="text" name="email" value="{{ old('email') }}" required autocomplete="email" autofocus class="form-control" id="email" required>
                          @error('email')
                            <div class="invalid-feedback" style="display: block">{{Lang::get('bahasa.Silahkan masukkan username.')}}</div>
                          @enderror
                        </div>
                      </div>
                      <div class="col-12">
                        <label for="password" class="form-label">Password</label>
```

```
                    <input
type="password" name="password"
required autocomplete="current-
password" class="form-control"
id="password" required>
                    @error('password')
                      <div
class="invalid-feedback"
style="display:
block">{{Lang::get('bahasa.Silahkan
masukkan password!')}}</div>
                    @enderror
                    {{-- <small><a
href="{{ route('password.request')
}}">{{Lang::get('bahasa.Lupa
Password!')}}</a></small> --}}
                  </div>
                  {{-- <div class="col-
12">
                    <div class="form-
check">
                      <input
class="form-check-input"
type="checkbox"  name="remember"
id="remember" value="true" {{
old('remember') ? 'checked' : '' }}>
                      <label class="form-
check-label"
for="remember">{{Lang::get('bahas
a.Ingat saya!')}}</label>
                    </div>
                  </div> --}}
                  <div class="col-12">
                    <button class="btn
btn-primary w-100"
type="submit">Login</button>
                  </div>
                  <div class="col-12">
                    {{-- <p class="small
mb-0">{{Lang::get('bahasa.Belum
punya akun?')}} <a
href="{{route('register')}}">{{Lang:
:get('bahasa.Buat akun
baru')}}</a></p> --}}
                  </div>
                </form>
              </div>
            </div>
          </div>
        </div>
      </section>
    </div>
  </main><!-- End #main -->

  <a href="#" class="back-to-top d-
flex align-items-center justify-
content-center"><i class="bi bi-
arrow-up-short"></i></a>
  @include('layouts.footer')
  <script>
    @if($errors->any())
      swal({
          title: "{{$errors-
>first()}}",
          text: "error",
          icon: "error",
          dangerMode: true,
        })
      var errjs="
      @if(count($errors->all())>1)

errjs='<p><b>{{Lang::get('bahasa.K
esalahan lainnya')}} : </b></p><ul>'
        @foreach($errors->all() as
$err)
          @if($loop->iteration>1)

errjs+='<li>{{$err}}</li>'
          @endif
        @endforeach
        errjs+='</ul>'
      @endif
      $('.swal-text').html(errjs)
    @endif
  </script>
  @yield('script')

</body>
</html>
```

Lampiran - 12 bab_index.blade.php

```
@extends('layouts.main')
@section('judul_halaman', 'BAB')
@section('konten')
  <div class="pagetitle">
    <h1>BAB</h1>
    <nav>
      <ol class="breadcrumb">
        <li class="breadcrumb-
item"><a href="#">Home</a></li>
        <li class="breadcrumb-
item">BAB</li>
      </ol>
    </nav>
  </div><!-- End Page Title -->

  <section class="section">
    <div class="row">
      <div class="col-lg-12">

        @if(session()->has('message'))
            <div class="alert alert-
success alert-
dismissible fade show" role="alert">
                <i class="bi bi-check-
circle me-1"></i>
                {{ session()-
>get('message') }}
                <button type="button" clas
s="btn-close" data-bs-
dismiss="alert" aria-
label="Close"></button>
            </div>
        @endif

        <div class="card">
          <div class="card-body">
            <h5 class="card-title">
                <div class="row">
                    <div class="col-sm-
6">BAB</div>
                    <div class="col-sm-
6" style="text-align: right">
                        <a class="btn btn-
sm btn-
success" href="{{route('bab.tambah')
}}"><i class="bi bi-plus-
square"></i> Tambah</a>
                    </div>
                </div>
            </h5>

            <div style="font-
size: 14px;">
                <!-
- Table with stripped rows -->
                <table class="table datat
able">
                    <thead>
                      <tr>
                        <th scope="col">#
</th>
                        <th scope="col">Ju
dul BAB</th>
                        <th scope="col">Ic
on BAB</th>
                        <th scope="col">A
ksi</th>
                      </tr>
                    </thead>
                    <tbody>
                        @foreach ($bab as
$item)
                            <tr>
                                <td>{{$loop-
>iteration}}.</td>
                                <td>{{$item-
>judulbab}}</td>
                                <td><img src=
"{{$item-
>iconbab}}" alt="Icon BAB" style="
max-width: 100px"></td>
                                <td align="rig
ht">
```

```
                                <a class="bt
n btn-sm btn-
warning" href="{{route('bab.edit',Cr
ypt::encrypt($item-
>idbab))}}"><i class="bi bi-pencil-
square"></i></a>
                                <button clas
s="btn btn-sm btn-
danger" onclick="hapus('{{Crypt::en
crypt($item-
>idbab)}}')"><i class="bi bi-
trash"></i></button>
                            </td>
                        </tr>
                    @endforeach
                </tbody>
                </table>
                <!-
- End Table with stripped rows -->
            </div>

        </div>
      </div>

    </div>
  </div>
 </section>

 <div class="modal fade" id="modal
loader" data-bs-
backdrop="static" data-bs-
keyboard="false" tabindex="-1">
    <div class="modal-dialog modal-
dialog-centered">
        <div class="modal-content">
            <div class="modal-
body" style="text-
align: center;padding-top: 23px;">
                <!-
- Growing Color spinnersr -->
                <div class="spinner-
grow text-primary" role="status">
                    <span class="visually-
hidden">Loading...</span>
                </div>
                <div class="spinner-
grow text-secondary" role="status">
                    <span class="visually-
hidden">Loading...</span>
                </div>
                <div class="spinner-
grow text-success" role="status">
                    <span class="visually-
hidden">Loading...</span>
                </div>
                <div class="spinner-
grow text-danger" role="status">
                    <span class="visually-
hidden">Loading...</span>
                </div>
                <div class="spinner-
grow text-warning" role="status">
                    <span class="visually-
hidden">Loading...</span>
                </div>
                <div class="spinner-
grow text-info" role="status">
                    <span class="visually-
hidden">Loading...</span>
                </div>
                <div class="spinner-
grow text-light" role="status">
                    <span class="visually-
hidden">Loading...</span>
                </div>
                <div class="spinner-
grow text-dark" role="status">
                    <span class="visually-
hidden">Loading...</span>
                </div><!-
- End Growing Color spinners -->
            </div>
        </div>
    </div>
  </div>

@endsection

@section('script')
    <script>
        function hapus(idx) {
```

```
        let psn="Data yang terhapus t
idak dapat dikembalikan!";
        swal({
            title: "Hapus Data!",
            text: psn,
            icon: "error",
            buttons: true,
        }).then((willDelete) => {
            if (willDelete) {
                prosesHapus(idx)
            }
        });
    }

    var myModal = new bootstrap.
Modal(document.getElementById('m
odalloader'));
    function prosesHapus(idx) {
        myModal.show();
        $.ajax({
            type: "POST",
            url: "{{route('bab.hapus')}
}",
            data: {
                _token:'{{ csrf_token() }
}',
                idx:idx,
            },
            dataType: "json",
            async: true,
            cache: false,
            success: function(res) {
                if (res['status']==1) {
                    swal({
                        title: "Terhapus!",
                        text: 'Data berhasil
terhapus!',
                        icon: "success",
                    }).then((willDelete) =
> {
                        location.reload();
                    });
                }else{
                    swal({
                        title: "Gagal!",
                        text: 'Data gagal ter
hapus!',
                        icon: "error",
                    }).then((willDelete) =
> {
                        location.reload();
                    });
                }
            },
            error: function (XHR, statu
s, response) {
                swal({
                    title: 'Koneksi Bermas
alah',
                    text: "Coba Lagi?!",
                    icon: 'error',
                    buttons: true,
                }).then((willDelete) =>
{
                    if (willDelete) {
                        prosesHapus(idx)
                    }else{
                        myModal.hide();
                    }
                });
            }
        })
    }
</script>
@endsection
```

Lampiran - 13 subbab_index.blade.php

```
@extends('layouts.main')
@section('judul_halaman', 'SUB BAB')
@section('konten')

  <div class="pagetitle">
    <h1>SUB BAB</h1>
    <nav>
      <ol class="breadcrumb">
        <li class="breadcrumb-item"><a href="#">Home</a></li>
        <li class="breadcrumb-item">SUB BAB</li>
      </ol>
    </nav>
  </div><!-- End Page Title -->

  <section class="section">
    <div class="row">
      <div class="col-lg-12">

        @if(session()->has('message'))
          <div class="alert alert-success alert-dismissible fade show" role="alert">
            <i class="bi bi-check-circle me-1"></i>
            {{ session()->get('message') }}
            <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
          </div>
        @endif

        <div class="card">
          <div class="card-body">
            <h5 class="card-title">
              <div class="row">
                <div class="col-sm-6">SUB BAB</div>
                <div class="col-sm-6" style="text-align: right">
                  <a class="btn btn-sm btn-success" href="{{route('subbab.tambah')}}"><i class="bi bi-plus-square"></i> Tambah</a>
                </div>
              </div>
            </h5>

            <div style="font-size: 14px;">
              <!-- Table with stripped rows -->
              <table class="table datatable">
                <thead>
                  <tr>
                    <th scope="col">#</th>
                    <th scope="col">Judul Sub BAB</th>
                    <th scope="col">BAB</th>
                    <th scope="col">Gambar Sub BAB</th>
                    <th scope="col">Aksi</th>
                  </tr>
                </thead>
                <tbody>
                  @foreach ($subbab as $item)
                    <tr>
                      <td>{{$loop->iteration}}.</td>
                      <td>{{$item->judulsubbab}}</td>
                      <td>{{$item->judulbab}}</td>
                      <td><img src="{{$item-
```

```
>gambarsubbab}}" alt="Gambar Sub
 BAB" style="max-
width: 100px"></td>
                        <td align="rig
ht">
                          <a class="bt
n btn-sm btn-
warning" href="{{route('subbab.edit'
,Crypt::encrypt($item-
>idsubbab))}}"><i class="bi bi-
pencil-square"></i></a>
                          <button clas
s="btn btn-sm btn-
danger" onclick="hapus('{{Crypt::en
crypt($item-
>idsubbab)}}')"><i class="bi bi-
trash"></i></button>
                        </td>
                      </tr>
                    @endforeach
                  </tbody>
                </table>
                <!-
- End Table with stripped rows -->
          </div>

        </div>
      </div>

    </div>
  </div>
 </section>

 <div class="modal fade" id="modal
loader" data-bs-
backdrop="static" data-bs-
keyboard="false" tabindex="-1">
    <div class="modal-dialog modal-
dialog-centered">
        <div class="modal-content">
          <div class="modal-
body" style="text-
align: center;padding-top: 23px;">
            <!-
- Growing Color spinnersr -->
            <div class="spinner-
grow text-primary" role="status">
              <span class="visually-
hidden">Loading...</span>
            </div>
            <div class="spinner-
grow text-secondary" role="status">
              <span class="visually-
hidden">Loading...</span>
            </div>
            <div class="spinner-
grow text-success" role="status">
              <span class="visually-
hidden">Loading...</span>
            </div>
            <div class="spinner-
grow text-danger" role="status">
              <span class="visually-
hidden">Loading...</span>
            </div>
            <div class="spinner-
grow text-warning" role="status">
              <span class="visually-
hidden">Loading...</span>
            </div>
            <div class="spinner-
grow text-info" role="status">
              <span class="visually-
hidden">Loading...</span>
            </div>
            <div class="spinner-
grow text-light" role="status">
              <span class="visually-
hidden">Loading...</span>
            </div>
            <div class="spinner-
grow text-dark" role="status">
              <span class="visually-
hidden">Loading...</span>
            </div><!-
- End Growing Color spinners -->
          </div>
        </div>
    </div>
 </div>
```

```
@endsection

@section('script')
   <script>
      function hapus(idx) {
         let psn="Data yang terhapus t
idak dapat dikembalikan!";
         swal({
            title: "Hapus Data!",
            text: psn,
            icon: "error",
            buttons: true,
         }).then((willDelete) => {
            if (willDelete) {
               prosesHapus(idx)
            }
         });
      }

      var myModal = new bootstrap.
Modal(document.getElementById('m
odalloader'));
      function prosesHapus(idx) {
         myModal.show();
         $.ajax({
            type: "POST",
            url: "{{route('subbab.hapus
')}}",
            data: {
               _token:'{{ csrf_token() }
}',
               idx:idx,
            },
            dataType: "json",
            async: true,
            cache: false,
            success: function(res) {
               if (res['status']==1) {
                  swal({
                     title: "Terhapus!",
                     text: 'Data berhasil
terhapus!',
                     icon: "success",
                  }).then((willDelete) =
> {
                     location.reload();
                  });
               }else{
                  swal({
                     title: "Gagal!",
                     text: 'Data gagal ter
hapus!',
                     icon: "error",
                  }).then((willDelete) =
> {
                     location.reload();
                  });
               }
            },
            error: function (XHR, statu
s, response) {
               swal({
                  title: 'Koneksi Bermas
alah',
                  text: "Coba Lagi?!",
                  icon: 'error',
                  buttons: true,
               }).then((willDelete) =>
{
                  if (willDelete) {
                     prosesHapus(idx)
                  }else{
                     myModal.hide();
                  }
               });
            }
         })
      }
   </script>
@endsection
```

Lampiran - 14 materi_index.blade.php

```
@extends('layouts.main')
@section('judul_halaman', 'MATERI
')
@section('konten')

  <div class="pagetitle">
    <h1>MATERI</h1>
    <nav>
      <ol class="breadcrumb">
        <li class="breadcrumb-
item"><a href="#">Home</a></li>
        <li class="breadcrumb-
item">MATERI</li>
      </ol>
    </nav>
  </div><!-- End Page Title -->

  <section class="section">
    <div class="row">
      <div class="col-lg-12">

        @if(session()->has('message'))
            <div class="alert alert-
success alert-
dismissible fade show" role="alert">
                <i class="bi bi-check-
circle me-1"></i>
                {{ session()-
>get('message') }}
                <button type="button" clas
s="btn-close" data-bs-
dismiss="alert" aria-
label="Close"></button>
            </div>
        @endif

        <div class="card">
          <div class="card-body">
            <h5 class="card-title">
              <div class="row">
                <div class="col-sm-
6">MATERI</div>
                <div class="col-sm-
6" style="text-align: right">
                  <a class="btn btn-
sm btn-
success" href="{{route('materi.tamba
h')}}"><i class="bi bi-plus-
square"></i> Tambah</a>
                </div>
              </div>
            </h5>

            <div style="font-
size: 14px;">
                <!-
- Table with stripped rows -->
                <table class="table datat
able">
                  <thead>
                    <tr>
                      <th scope="col">#
</th>
                      <th scope="col">B
AB</th>
                      <th scope="col">S
ub BAB</th>
                      <th scope="col">Ju
dul Materi</th>
                      <th scope="col">Is
i Materi</th>
                      <th scope="col">A
ksi</th>
                    </tr>
                  </thead>
                  <tbody>
                    @foreach ($materi
as $item)
                      <tr>
                        <td>{{$loop-
>iteration}}.</td>
                        <td>{{$item-
>judulbab}}</td>
                        <td>{{$item-
>judulsubbab}}</td>
```

```
                        <td>{{$item->judulmateri}}</td>
                        <td>{{$item->isimateri}}</td>
                        <td align="right">
                            <a class="btn btn-sm btn-warning" href="{{route('materi.edit', Crypt::encrypt($item->idmateri))}}"><i class="bi bi-pencil-square"></i></a>
                            <button class="btn btn-sm btn-danger" onclick="hapus('{{Crypt::encrypt($item->idmateri)}}')"><i class="bi bi-trash"></i></button>
                        </td>
                    </tr>
                @endforeach
            </tbody>
        </table>
        <!-- End Table with stripped rows -->
        </div>

      </div>
    </div>

  </div>
 </section>

 <div class="modal fade" id="modalloader" data-bs-backdrop="static" data-bs-keyboard="false" tabindex="-1">
    <div class="modal-dialog modal-dialog-centered">
      <div class="modal-content">
        <div class="modal-body" style="text-align: center;padding-top: 23px;">
          <!-- Growing Color spinnersr -->
          <div class="spinner-grow text-primary" role="status">
            <span class="visually-hidden">Loading...</span>
          </div>
          <div class="spinner-grow text-secondary" role="status">
            <span class="visually-hidden">Loading...</span>
          </div>
          <div class="spinner-grow text-success" role="status">
            <span class="visually-hidden">Loading...</span>
          </div>
          <div class="spinner-grow text-danger" role="status">
            <span class="visually-hidden">Loading...</span>
          </div>
          <div class="spinner-grow text-warning" role="status">
            <span class="visually-hidden">Loading...</span>
          </div>
          <div class="spinner-grow text-info" role="status">
            <span class="visually-hidden">Loading...</span>
          </div>
          <div class="spinner-grow text-light" role="status">
            <span class="visually-hidden">Loading...</span>
          </div>
          <div class="spinner-grow text-dark" role="status">
            <span class="visually-hidden">Loading...</span>
          </div><!-- End Growing Color spinners -->
        </div>
      </div>
    </div>
  </div>
```

```
@endsection

@section('script')
   <script>
      function hapus(idx) {
         let psn="Data yang terhapus t
idak dapat dikembalikan!";
         swal({
            title: "Hapus Data!",
            text: psn,
            icon: "error",
            buttons: true,
         }).then((willDelete) => {
            if (willDelete) {
               prosesHapus(idx)
            }
         });
      }

      var myModal = new bootstrap.
Modal(document.getElementById('m
odalloader'));
      function prosesHapus(idx) {
         myModal.show();
         $.ajax({
            type: "POST",
            url: "{{route('materi.hapus'
)}}",
            data: {
               _token:'{{ csrf_token() }
}',
               idx:idx,
            },
            dataType: "json",
            async: true,
            cache: false,
            success: function(res) {
               if (res['status']==1) {
                  swal({
                     title: "Terhapus!",
                     text: 'Data berhasil
terhapus!',
                     icon: "success",
                  }).then((willDelete) =
> {
                     location.reload();
                  });
               }else{
                  swal({
                     title: "Gagal!",
                     text: 'Data gagal ter
hapus!',
                     icon: "error",
                  }).then((willDelete) =
> {
                     location.reload();
                  });
               }
            },
            error: function (XHR, statu
s, response) {
               swal({
                  title: 'Koneksi Bermas
alah',
                  text: "Coba Lagi?!",
                  icon: 'error',
                  buttons: true,
               }).then((willDelete) =>
{
                  if (willDelete) {
                     prosesHapus(idx)
                  }else{
                     myModal.hide();
                  }
               });
            }
         })
      }
   </script>
@endsection
```

Lampiran - 15 profil_index.blade.php

```
@extends('layouts.main')
@section('judul_halaman', Lang::get
('bahasa.Profil'))
@section('konten')

  <div class="pagetitle">
    <h1>{{Lang::get('bahasa.Profil')}
}</h1>
    <nav>
      <ol class="breadcrumb">
        <li class="breadcrumb-
item"><a href="index.html">{{Lang
::get('bahasa.Pengaturan')}}</a></li
>
        <li class="breadcrumb-
item">{{Lang::get('bahasa.Profil')}}
</li>
      </ol>
    </nav>
  </div><!-- End Page Title -->

  <section class="section">
    <div class="row">
      <div class="col-lg-12">

        @if(session()->has('message'))
            <div class="alert alert-
success alert-
dismissible fade show" role="alert">
                <i class="bi bi-check-
circle me-1"></i>
                {{ session()-
>get('message') }}
                <button type="button" clas
s="btn-close" data-bs-
dismiss="alert" aria-
label="Close"></button>
            </div>
        @endif

        <div class="card">
          <div class="card-body">
            <h5 class="card-title">
              {{Lang::get('bahasa.Profil'
)}}
            </h5>

            <form action="{{route('profil
.proses')}}" method="post" enctype=
"multipart/form-data">
                @csrf
                <div class="row mb-3">
                  <label for="name" class=
"col-sm-2 col-form-
label">{{Lang::get('bahasa.Nama')}
}</label>
                  <div class="col-sm-10">
                    <input type="text" class
="form-
control" name="name" required valu
e="{{auth()->user()->name}}">
                  </div>
                </div>

                <div class="row mb-3">
                  <label for="foto" class="c
ol-sm-2 col-form-
label">{{Lang::get('bahasa.Foto')}}
</label>
                  <div class="col-sm-10">
                    <input type="file" class
="form-
control" name="foto" onchange="rea
lchangeimg(this)" accept="image/*"
>
                    <img src="{{auth()-
>user()-
>foto}}" alt="Logo Rek" style="max
-width: 100px;margin-
top: 5px;margin-left: 1px;">
                  </div>
                </div>

                <div class="row mb-3">
```

```
                <label for="email" class
="col-sm-2 col-form-
label">Email</label>
                <div class="col-sm-
10">
                  <input type="email" cl
ass="form-
control" name="email" required valu
e="{{auth()->user()->email}}">
                </div>
              </div>

              <div class="row mb-3">
                <label class="col-sm-
2 col-form-label"></label>
                <div class="col-sm-10">
                  <button type="submit" c
lass="btn btn-
primary">{{Lang::get('bahasa.Simpa
n')}}</button>
                </div>
              </div>

            </form>

          </div>
        </div>

      </div>
    </div>
  </section>

@endsection

@section('modal')
@endsection

@section('script')
  <script>
    function realchangeimg(input) {
      if (input.files dandan input.fil
es[0]) {
        var reader = new FileRead
er();

        reader.onload = function(e)
 {
          $(input).parent().find('i
mg').attr('src', e.target.result);
        }
        reader.readAsDataURL(in
put.files[0]);
      }else{
        $(input).parent().find('img'
).attr('src', '{{asset('asset/img/noimag
e.jpg')}}');
      }
    }
  </script>
@endsection
```

Lampiran - 16 password_index.blade.php

```
@extends('layouts.main')
@section('judul_halaman', 'Password
')
@section('konten')

  <div class="pagetitle">
    <h1>Password</h1>
    <nav>
      <ol class="breadcrumb">
        <li class="breadcrumb-
item"><a href="index.html">{{Lang
::get('bahasa.Pengaturan')}}</a></li
>
        <li class="breadcrumb-
item">Password</li>
      </ol>
    </nav>
  </div><!-- End Page Title -->

  <section class="section">
    <div class="row">
      <div class="col-lg-12">

        @if(session()->has('message'))
            <div class="alert alert-
success alert-
dismissible fade show" role="alert">
                <i class="bi bi-check-
circle me-1"></i>
                {{ session()-
>get('message') }}
                <button type="button" clas
s="btn-close" data-bs-
dismiss="alert" aria-
label="Close"></button>
            </div>
        @endif

        <div class="card">
          <div class="card-body">
            <h5 class="card-title">
              Password
            </h5>
            <form action="{{route('pass
word.proses')}}" method="post" enct
ype="multipart/form-data">
                @csrf
                <div class="row mb-3">
                  <label for="passwordlam
a" class="col-sm-2 col-form-
label">{{Lang::get('bahasa.Password
 Lama')}}</label>
                  <div class="col-sm-10">
                    <input type="password"
class="form-
control" id="passwordlama" name="
passwordlama" required value="">
                  </div>
                </div>

                <div class="row mb-3">
                  <label for="password" cla
ss="col-sm-2 col-form-
label">{{Lang::get('bahasa.Password
 Baru')}}</label>
                  <div class="col-sm-10">
                    <input type="password"
class="form-
control" id="password" name="pass
word" required value="">
                  </div>
                </div>

                <div class="row mb-3">
                  <label for="password_co
nfirmation" class="col-sm-2 col-
form-
label">{{Lang::get('bahasa.Konfirm
asi Password Baru')}}</label>
                  <div class="col-sm-10">
                    <input type="password"
class="form-
control" id="password_confirmation
```

```
" name="password_confirmation" re
quired value="">
                </div>
              </div>

              <div class="row mb-3">
                <label class="col-sm-
2 col-form-label"></label>
                <div class="col-sm-10">
                  <button type="submit" c
lass="btn btn-
primary">{{Lang::get('bahasa.Simpa
n')}}</button>
                </div>
              </div>

            </form>
                                    </div>
                                  </div>

                                </div>
                              </div>
                            </section>

@endsection

@section('modal')
@endsection

@section('script')
    <script>
    </script>
@endsection
```