

Latihan Modul 4

✓ `/* Latihan 4.1 */`

➤ Source Code

`//Home.js`

```
import React, { Component } from 'react';
import {
  View,
  Text,
} from 'react-native';

export default class Home extends Component {
  render(){
    return(
      <View>
        <Text>Home</Text>
      </View>
    );
  }
};
```

`//Profile.js`

```
import React, { Component } from 'react';
import {
  View,
  Text
} from 'react-native';

export default class Profile extends Component {
  render(){
    return(
      <View>
        <Text>Screen Profile</Text>
      </View>
    );
  }
};
```

//RootApp.js

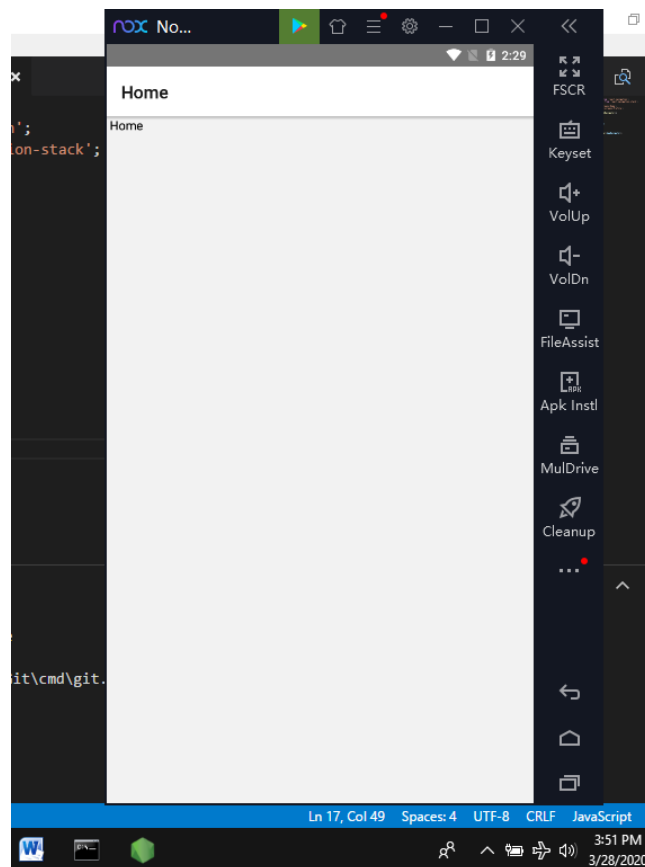
```
import React from 'react';
import { createAppContainer } from 'react-navigation';
import { createStackNavigator } from 'react-navigation-stack';

import HomeComponent from './screens/Home';
import ProfileComponent from './screens/Profile';

const AppNavigator = createStackNavigator({
  Home: {
    screen: HomeComponent
  },
  Profile: {
    screen: ProfileComponent
  }
});

export default createAppContainer(AppNavigator);
```

➤ Tampilan



✓ `/* Latihan 4.2 */`

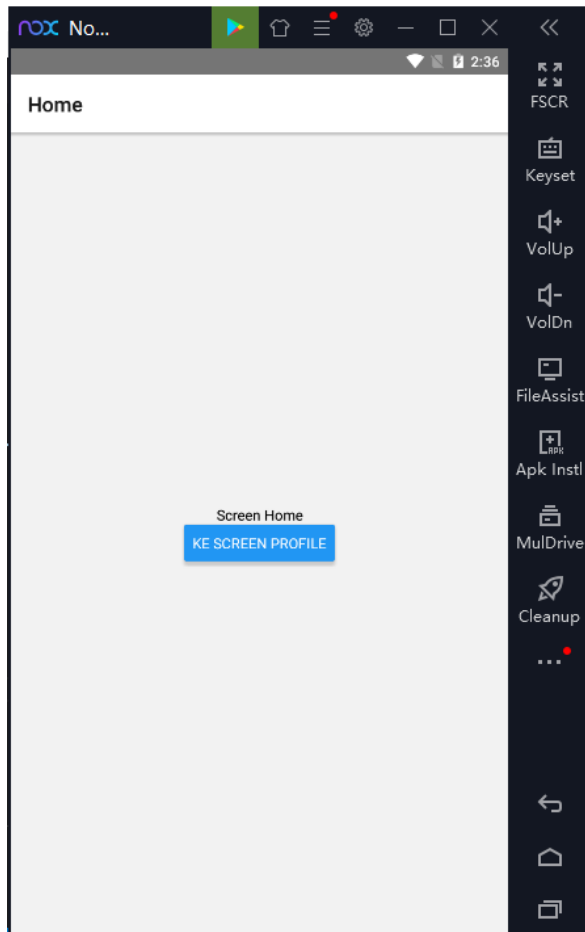
➤ Source Code

```
import React, { Component } from 'react';
import {
  View,
  Text,
  Button,
  StyleSheet
} from 'react-native';

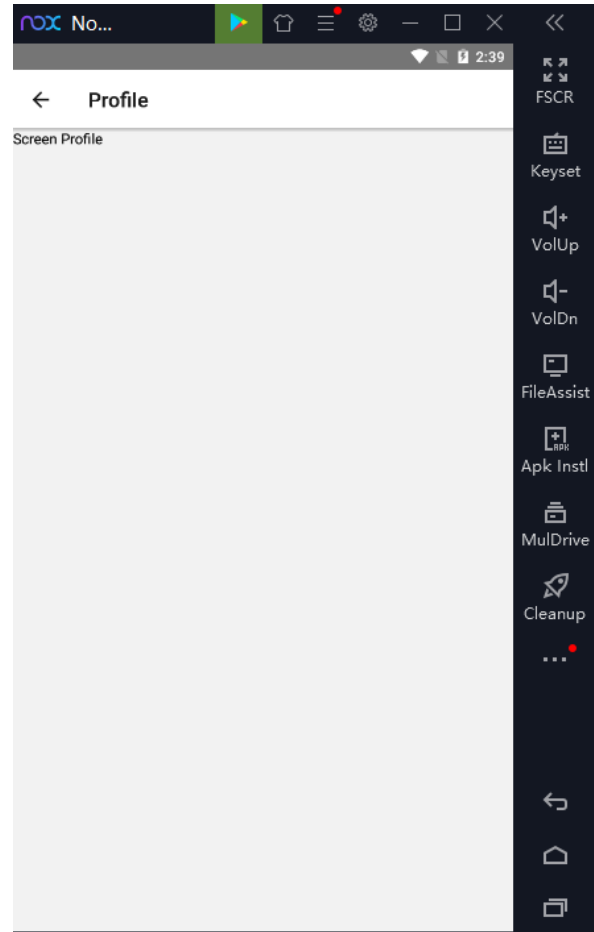
export default class Home extends Component {
  render(){
    return(
      <View style={styles.container}>
        <Text>Screen Home</Text>
        <Button title="Ke Screen Profile" onPress={()=>
this.props.navigation.navigate('Profile')}/>
      </View>
    );
  }
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    flexDirection: 'column',
    justifyContent: 'center',
    alignItems: 'center'
  }
});
```

➤ Tampilan
//Screen Home



//Screen Profile



Note : Pada Screen Home, Ketika Button ‘Ke Screen Profile’ ditekan maka akan beralih ke screen baru berdasarkan nama route yang dipanggil. Ketika kita klik header bar navigation yang berada di sudut kiri paling atas pada screen Profil, maka akan langsung beralih ke screen Home. Tetapi Methode Navigate akan mengecek Halaman yang dituju, apakah sudah ada di stack(Tumpukan Halaman) atau belum. Jikalau ada, maka akan dipanggil halaman yang sudah ada untuk ditampilkan.

✓ `/* Latihan 4.3 */`

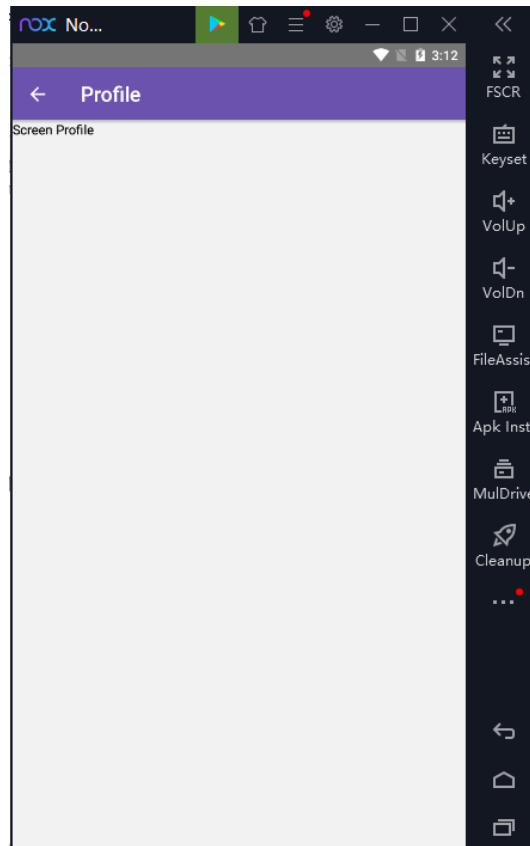
➤ Source Code

```
import React, { Component } from 'react';
import {
  View,
  Text
} from 'react-native';

export default class Profile extends Component {
  static navigationOptions = {
    title: 'Profile',
    headerStyle: {
      backgroundColor: '#6b52ae'
    },
    headerTintColor: 'fff',
    headerTitleStyle: {
      fontSize: 22
    }
  }

  render(){
    return(
      <View>
        <Text>Screen Profile</Text>
      </View>
    );
  }
};
```

➤ Tampilan



Note : Ketika Perubahan terhadap Code Profile.js selesai dan disave, maka Halaman Profile tidak terjadi perubahan yang berbeda sebelum perubahan. Maka dari itu harus dilakukan beberapa Konfigurasi, yaitu :

- Setelah Code pada Profile.js diberikan Perubahan, Maka Lakukan Save untuk Profile.js saja
 - Setelah itu, Berikan Perubahan Nama Route untuk halaman Profile pada RootApp.js
- Contoh :

```
Nico: { // Profile >> Nico
  screen: ProfileComponent
}
```

- Setelah itu, berikan juga perubahan terhadap nama Route untuk halaman Profile yang berfungsi untuk pemanggilan halaman Profile tersebut yang berada pada Home.js
- Contoh :

```
<Button title="Ke Screen Profile" onPress={()=>
this.props.navigation.navigate('Nico')}/> { /*Profile >> Nico*/ }
```

- Setelah itu Save All.
- Maka Jikalau Tampilan Emulator berada di halaman Profile, ketika dilakukan save all maka akan kembali ke halaman Home. Klik Button 'Ke Screen Profile' maka akan terjadi perubahan, dan begitu seterusnya

✓ `/* Latihan 4.4 */`

➤ Source Code

`//Home.js`

```
import React, { Component } from 'react';
import {
  View,
  Text,
  Button,
  StyleSheet
} from 'react-native';

export default class Home extends Component {
  render(){
    return(
      <View style={styles.container}>
        <Text>Screen Home</Text>
        <Button
          title="Ke Screen Profile"
          onPress={() => this.props.navigation.navigate('Nico', {nama: 'Jon
Doe'}})></>
      </View>
    )
  }
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    flexDirection: 'column',
    justifyContent: 'center',
    alignItems: 'center'
  }
});
```

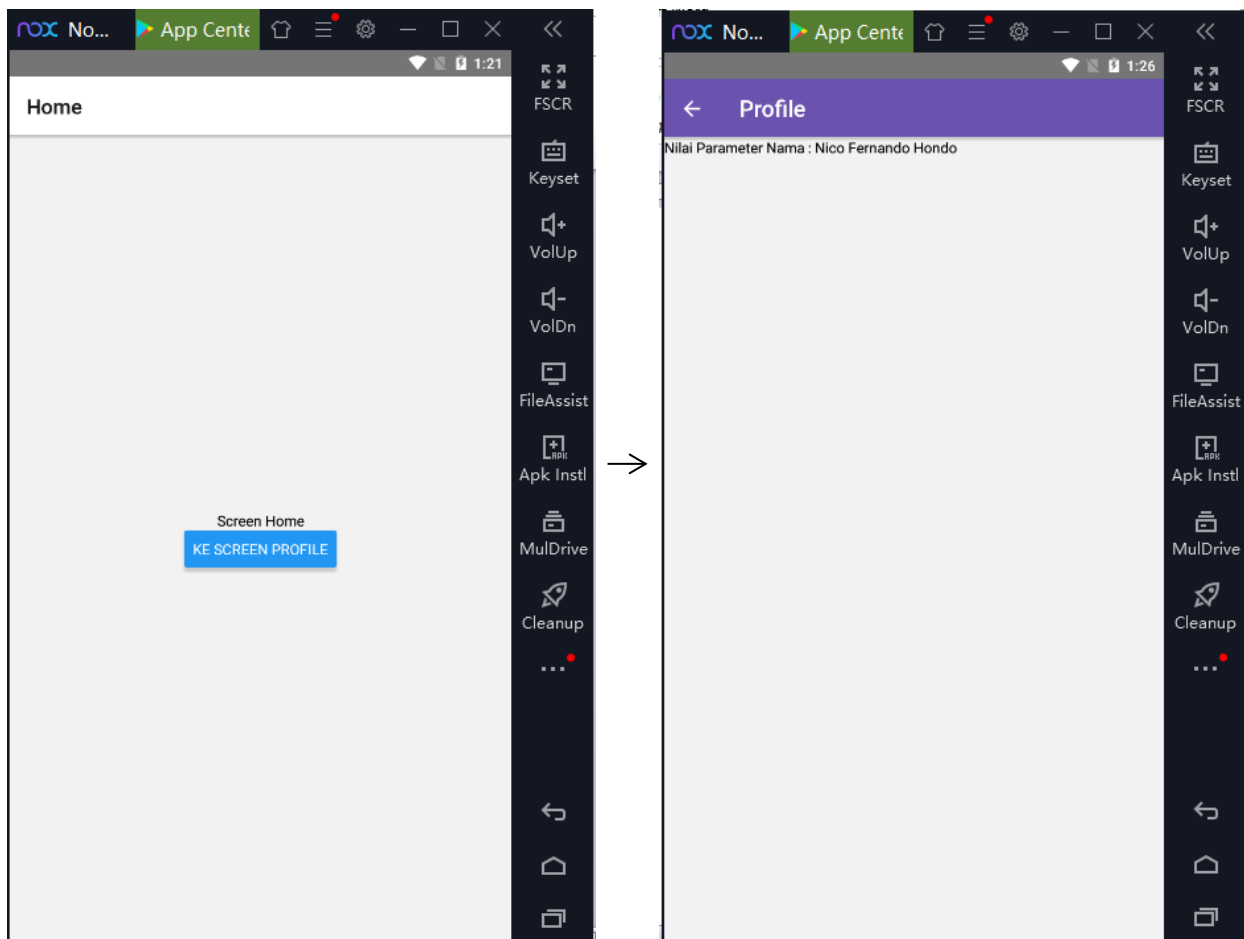
//Profile.js

```
import React, { Component } from 'react';
import {
  View,
  Text
} from 'react-native';

export default class Profile extends Component {
  static navigationOptions = {
    title: 'Profile',
    headerStyle: {
      backgroundColor: '#6b52ae'
    },
    headerTintColor: '#fff',
    headerTitleStyle: {
      fontSize: 22
    }
  };

  render(){
    return(
      <View>
        <Text>
          Nilai Parameter Nama:{this.props.navigation.getParam('nama',
'Ghost')}}
        </Text>
      </View>
    );
  }
};
```


➤ Tampilan



➤ Penambahan navigationOptions di header Profile.js

Pada Home.js saya tambahkan 1 parameter lagi yaitu 'namaPanggilan' yang dimana parameter tersebut akan digunakan untuk mengganti Title pada halaman Profile.js

//Home.js

```
return(  
  <View style={styles.container}>  
    <Text>Screen Home</Text>  
    <Button  
      title="Ke Screen Profile"  
      onPress={() => this.props.navigation.navigate('Akun', {nama: ' Nico  
Fernando Hondo', namaPanggilan: 'Nico'})}/>  
    </View>  
  )
```

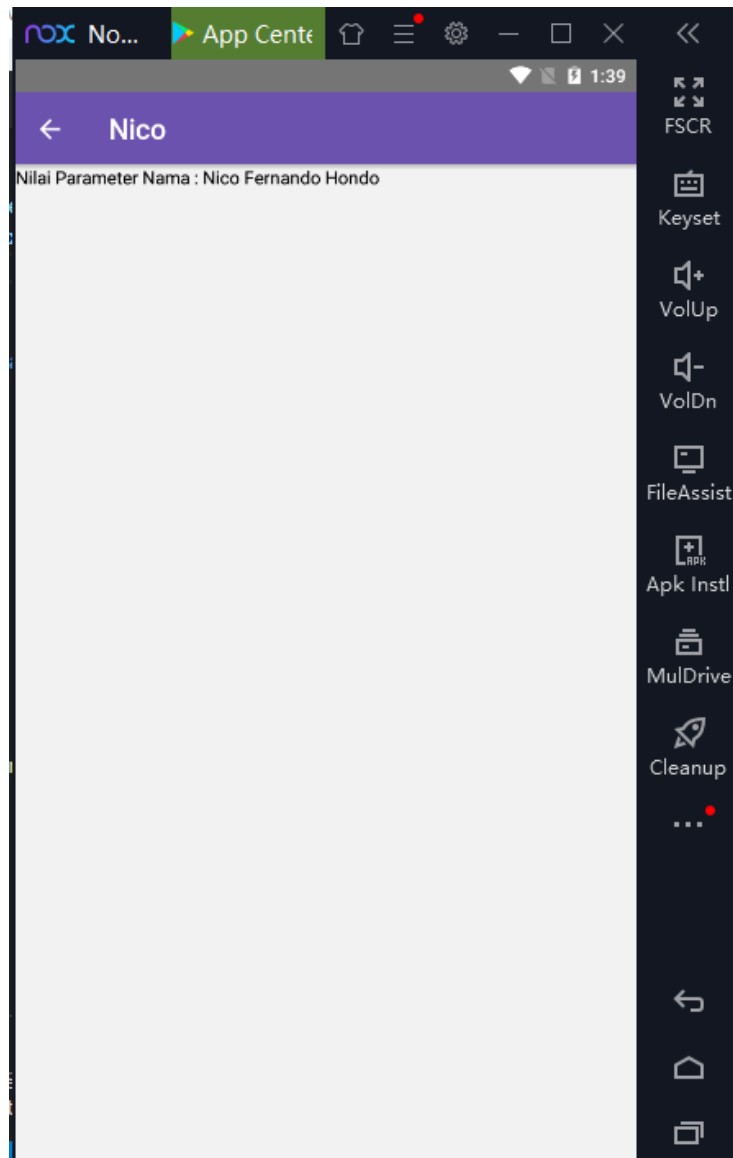
//Profile.js

```
import React, { Component } from 'react';
import {
  View,
  Text
} from 'react-native';

export default class Profile extends Component {
  static navigationOptions = ({navigation})=> {
    return {
      title: navigation.state.params.namaPanggilan,
      headerStyle: {
        backgroundColor: '#6b52ae'
      },
      headerTintColor: '#fff',
      headerTitleStyle: {
        fontSize: 22
      }
    }
  };

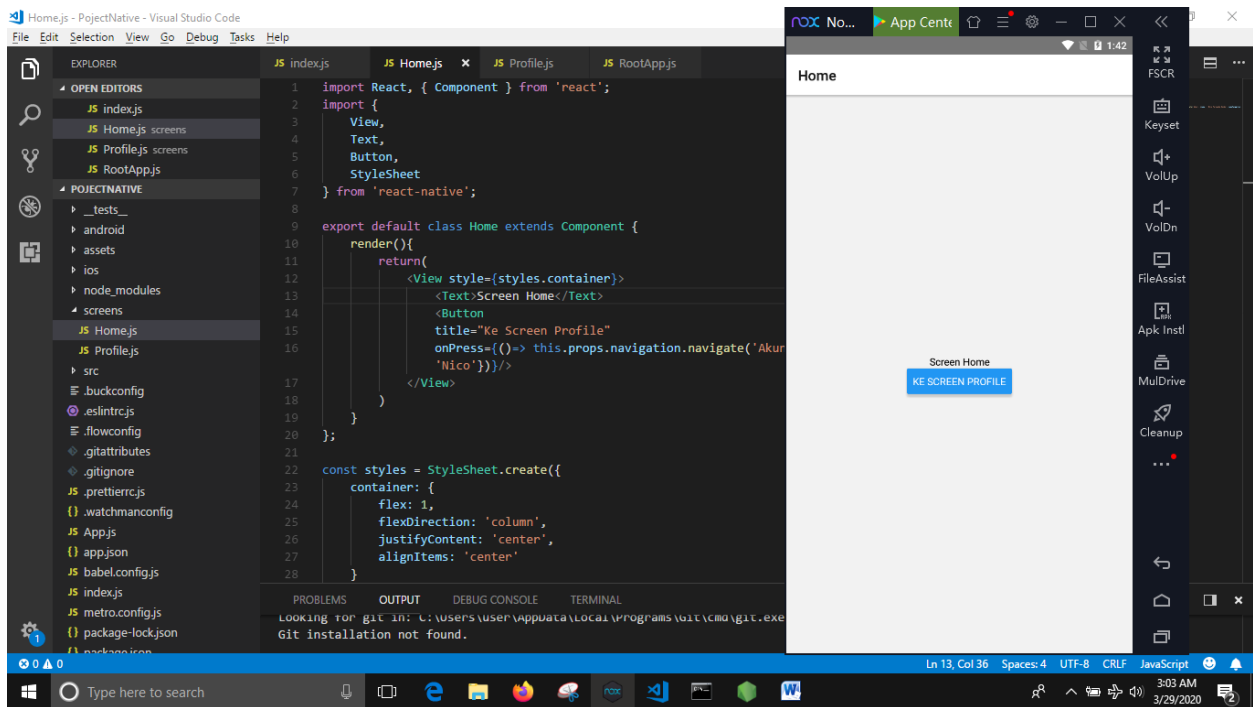
  render(){
    return(
      <View>
        <Text>
          Nilai Parameter Nama :{this.props.navigation.getParam('nama',
'Ghost')}
        </Text>
      </View>
    );
  }
};
```

➤ Tampilan



Tampilan Keseluruhan :

✓ Home.js



✓ Profile.js

