

MANUAL SISTEM

"BudgetGenius" Aplikasi Manajemen Keuangan Berbasis Android



OLEH

Hielmi Sulaeman (NPM. 5200411469)
Anita Fira Waluyo, S.Si., M.Sc.

UNIVERSITAS TEKNOLOGI YOGYAKARTA
TAHUN 2023

DAFTAR ISI

DAFTAR ISI.....	i
DAFTAR GAMBAR.....	ii
1. TAMPILAN DAN KODE PROGRAM.....	1
1.1 Halaman Login Aplikasi.....	1
1.1.1. Tampilan Login Aplikasi BudgetGenius.....	1
1.1.2. Potongan Source Code Dashboard.....	1
1.2 Halaman Home.....	12
1.2.1. Tampilan Halaman Home.....	12
1.2.2. Potongan Source Code Membuat Halaman Home.....	12
1.3 Halaman Kategori Pemasukan dan Pengeluaran.....	15
1.3.1. Tampilan Halaman Kategori Pemasukan dan Pengeluaran.....	15
1.3.2. Potongan Source Code Kategori Pemasukan dan Pengeluaran.....	15
1.4 Halaman Transaksi Pemasukan atau Pengeluaran.....	18
1.4.1 Tampilan Halaman Transaksi Pemasukan atau Pengeluaran.....	18
1.4.2 Source Code Halaman Dashboard Admin.....	18
1.5 Halaman Kantong Tabungan.....	22
1.5.1 Tampilan Halaman Kantong Tabungan.....	22
1.5.2 Source Code Tampilan Kantong Tabungan.....	22
1.6 Halaman Laporan.....	26
1.6.1 Tampilan Halaman Laporan.....	26
1.6.2 Source Code Tampilan Halaman Laporan.....	26
1.7 Halaman Setting.....	29
1.7.1 Tampilan Halaman Setting.....	29
1.7.2 Source Code Halaman Setting.....	29
2. PENGGUNAAN PROGRAM.....	32
2.1 Langkah Membuat Akun Di BudgetGenius.....	32
2.2 Melakukan transaksi pemasukan.....	33
2.3 Menambah Kategori.....	34
2.4 Halaman Laporan.....	34

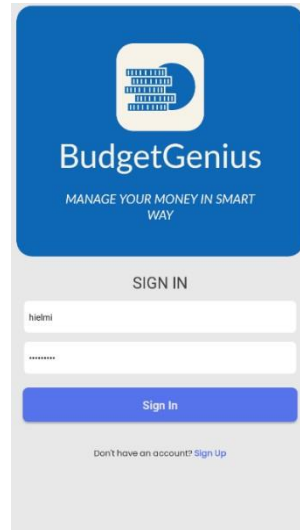
DAFTAR GAMBAR

Gambar 1 - Halaman Login	1
Gambar 2 – Halaman Home	12
Gambar 3 – Halaman Kategori Pemasukan dan Pengeluaran	15
Gambar 4 – Halaman Transaksi.....	18
Gambar 5 – Halaman Kantong Tabungan	22
Gambar 6 – Tampilan Halaman Laporan.....	26
Gambar 7 – Tampilan Halaman Setting.....	29
Gambar 8 - Halaman Home	33
Gambar 9 – Melakukan Transaksi pemasukan	33
Gambar 10 – Halaman Laporan	35

1. TAMPILAN DAN KODE PROGRAM

1.1 Halaman Login Aplikasi

1.1.1. Tampilan Login Aplikasi BudgetGenius



Gambar 1 - Halaman Login

1.1.2. Potongan Source Code Dashboard

```
src > screens > SignInScreens > ...
You, 6 seconds ago | 1 author (You)
1 import React, { useState, useContext } from 'react';
2 import { TextInput, StyleSheet, Text, View, TouchableOpacity, Image, useWindowDimensions, Alert } from 'react-native';
3 import { KeyboardAwareScrollView } from 'react-native-keyboard-aware-scroll-view'
4 import { theme } from '../assets/themes/themes'
5 import Logo from '../assets/images/Logo.jpeg'
6 import { Header } from '../components/Header'; You, 6 months ago • add Sign In Screen, Sign Up Screen, several compo...
7 import * as Keychain from 'react-native-keychain';
8 import { AuthContext } from '../context/AuthContext';
9 import { AxiosContext } from '../context/AxiosContext';
10
11 export const SignInScreen = ({ navigation }) => {
12   const authContext = useContext(AuthContext);
13   const { authAxios } = useContext(AxiosContext);
14   const [username, setUsername] = useState("");
15   const [password, setPassword] = useState("");
16   const { width } = useWindowDimensions();
17
18   const onLogin = async () => {
19     try {
20       const response = await authAxios.post('login', {
21         username: username,
22         password: password,
23       });
24
25       if (response.status == 200) {
26         const { accessToken } = response.data;
```

```

27     authContext.setAuthState({
28         token: accessToken,
29         authenticated: true
30     });
31     await Keychain.setGenericPassword('token', accessToken);
32 }
33 else {
34     Alert.alert('Login', response.data.msg)
35 }
36 }
37 catch (error) {
38     console.log(error);
39     Alert.alert('Error', 'Periksa internet anda atau inputan password atau email anda salah')
40 }
41 });
42
43 return (
44     <KeyboardAwareScrollView style={{ backgroundColor: theme.colors.background }}>
45         <View style={{ backgroundColor: theme.colors.background }}>
46             <View style={{ ...theme.centering, padding: theme.spacing.s }}>
47                 <Image source={Logo} style={{ width: width - 20, height: width - 80, borderRadius: 25 }} />
48             </View>
49             <Header title="SIGN IN" />
50             <View style={{ backgroundColor: theme.colors.background, borderRadius: 30 }}>
51                 <TextInput
52                     value={username}
53                     onChangeText={text => setUsername(text)}
54                     style={{ ...style.textInput, borderColor: '#e8e8e8', borderWidth: 2 }}
55                     placeholder="username or email"
56                     placeholderTextColor="#8687B9"
57                     underlineColorAndroid='transparent'
58                 />
59                 <TextInput
60                     value={password}
61                     onChangeText={text => setPassword(text)}
62                     style={{ ...style.textInput, }}
63                     placeholder="password"
64                     secureTextEntry={true}
65                     placeholderTextColor="#8687B9"
66                     underlineColorAndroid='transparent'
67                 />
68                 <TouchableOpacity
69                     style={{
70                         marginTop: 20,
71                         backgroundColor: theme.colors.primary,
72                         paddingVertical: 15,
73                         marginHorizontal: 20,
74                         justifyContent: 'center',
75                         alignItems: 'center',
76                         borderRadius: 9,
77                         elevation: 2,
78                     }} onPress={() => onLogin()}>
79

```

```

76     <Text style={{ color: theme.colors.background, fontSize: 18, fontWeight: 'bold' }}> Sign In </Text>
77 </TouchableOpacity>
78 <View style={{ padding: theme.spacing.x1 }}>
79     <Text style={{ ...theme.textVariants.body, fontSize: 13, textAlign: 'center', color: theme.colors.fontColor }}>
80         Don't have an account?{' '}
81         <Text style={{ color: theme.colors.primary, fontWeight: 'bold' }}
82             onPress={() => navigation.navigate('RegisterScreen')}>Sign Up</Text>
83     </Text>
84 </View>
85 </View>
86 </KeyboardAwareScrollView >;
87 }
88
89 const style = StyleSheet.create({
90     textInput: {
91         marginHorizontal: 20, backgroundColor: 'white', color: theme.colors.fontColorDark, marginTop: 10, borderRadius: 9,
92         padding: 12, paddingLeft: 10, borderColor: theme.colors.background,
93         borderWidth: 2 }
94 })

```


1.2 Halaman Home

1.2.1. Tampilan Halaman Home



Gambar 2 – Halaman Home

1.2.2. Potongan Source Code Membuat Halaman Home

```
1 import React, { useContext, useEffect, useState } from "react";
2 import { Text, View, ScrollView, TouchableOpacity } from "react-native";
3 import { theme } from "../assets/themes/themes";
4 import { AuthContext } from "../context/AuthContext";
5 import { AxiosContext } from "module "c:/Users/Administrator/Documents/react-native/MoneyBook_v2/src/components/HeaderProfile";
6 import { HeaderProfile } from "../components/HeaderProfile";
7 import { BalanceUser } from "../components/BalanceUser";
8 import { BalanceBudget } from "../components/BalanceBudget";
9 import HistoryComponent from "../components/HistoryComponent";
10 import { Spinner } from './Spinner';
11 import axios from "axios";
12
13 export const HomeScreen = ({ navigation, route }) => {
14   const authContext = useContext(AuthContext);
15   const authState = authContext.authState;
16   const userState = authContext.userState;
17   const saldoState = authContext.saldoState;
18   const historyState = authContext.historyState;
19   const { publicAxios, BASE_URL } = useContext(AxiosContext);
20   const [isLoading, setIsLoading] = useState(true);
21
22   useEffect(() => {
23     const fetchData = async () => {
24       try {
25         const accessToken = authState.token
26         if (!accessToken) {
```



```
src > screens > HomeScreen.js > HomeScreen
27     authContext.logout()
28   }
29   const getUser = publicAxios.get('user')
30   const getSaldo = publicAxios.get('user/saldo')
31   const getTransaction = publicAxios.get('user/transaction')
32   const getPocket = publicAxios.get('user/pocket');
33
34   Promise.all([getUser, getSaldo, getTransaction, getPocket])
35     .then(axios.spread(...responses) => {
36       const user = responses[0]
37       const saldo = responses[1]
38       const transaction = responses[2].data.sort(compareTheDate);
39       const pocket = responses[3].data
40       const { email, firstname, lastname, photoProfile, username } = user.data
41       authContext.setUserState({ email, firstname, lastname, photoProfile, username })
42       authContext?.setPocketState(pocket)
43       if (saldo.data === null) {
44         authContext.setSaldoStateInitialValue()
45       } else {
46         const { need, want, invest, saving } = saldo.data
47         authContext.setSaldoState({
48           saldoInvest: invest,
49           saldoNeed: need,
50           saldoSaving: saving,
51           saldoWant: want,
52           jumlah: [need, want, invest, saving].reduce((acc, value) => acc += value)
53         })
54     }
55     authContext.setHistoryState(transaction)
56   })
57   .catch((err) => {
58     setIsLoading(false)
59     authContext.logout()
60     console.log('err', err)
61   }).finally(() => setIsLoading(false))
62 } catch (error) {
63   authContext.logout()
64   console.log('error screen index ' + error);
65 }
66   fetchData()
67 }, [route]);
68 const compareTheDate = (a, b) => {
69   const dateA = new Date(a.createdAt);
70   const dateB = new Date(b.createdAt);
71   return dateB - dateA;
72 }
73 const handleDetailTransaction = (idTransaction,
74   amount,
75   date,
76   type,
77   note,
78   tb_category,) => {
79   navigation.navigate('DetailTransactionScreen', {
```

```

src > screens > HomeScreen.js > HomeScreen
81     data: {
82         idTransaction,
83         amount,
84         date,
85         type,
86         note,
87         tb_category,
88     }
89 }
90 }
91 if (isLoading) {
92     return <Spinner />
93 }
94 return (
95     <ScrollView style={{ padding: 10, paddingTop: 20 }}>
96         <HeaderProfile fullName={userState?.firstname + ' ' + userState?.lastname} imgUrl={userState?.photoProfile !== null ? userState?.photoProfile : null} />
97         <BalanceUser saldo={saldoState?.jumlah} />
98         <Text style={{ ...theme.textVariants.body, color: theme.colors.fontColor, fontWeight: 'bold', marginTop: theme.spacing.1 }}> Anggaran
99     <View style={{ backgroundColor: theme.colors.foreground, padding: theme.spacing.s, borderRadius: 15, marginTop: 5 }}>
100         <BalanceBudget
101             title={'Nominal Kebutuhan'}
102             saldo={saldoState?.saldoNeed}
103             nameIcon={'fast-food-outline'}
104             size={30}
105         />
106         <BalanceBudget

```

```

src > screens > HomeScreen.js > HomeScreen
107             title={'Nominal Keinginan'}
108             saldo={saldoState?.saldoWant}
109             nameIcon={'cart-outline'}
110             size={35}
111         />
112         <BalanceBudget
113             title={'Nominal Investasi'}
114             saldo={saldoState?.saldoInvest}
115             nameIcon={'trending-up-outline'}
116             size={35}
117         />
118         <BalanceBudget
119             title={'Nominal Tabungan'}
120             saldo={saldoState?.saldoSaving}
121             nameIcon={'wallet-outline'}
122             size={35}
123         />
124     </View>
125     <Text style={{ ...theme.textVariants.body, color: theme.colors.fontColor, fontWeight: 'bold', marginTop: theme.spacing.1 }}>
126     Transaksi terakhir</Text>
127     <View style={{ marginBottom: 100 }}>
128         {historyState && historyState.map((history, index) => {
129             if (index < 5) {
130                 return (
131                     <HistoryComponent
132                         key={history.idTransaction}

```

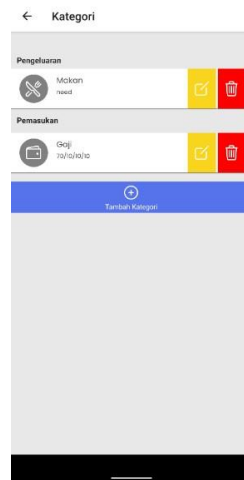
```

src > screens > HomeScreen.js > HomeScreen > historyState.map() callback
133         idTransaction={history.idTransaction}
134         amount={history.amount}
135         date={history.date}
136         type={history.type}
137         note={history.note}
138         tb_category={history.tb_category}
139         handler={handleDetailTransaction}
140     />
141     )
142 }
143 }}
144 </View>
145 </ScrollView>
146 )
147 }

```

1.3 Halaman Kategori Pemasukan dan Pengeluaran

1.3.1. Tampilan Halaman Kategori Pemasukan dan Pengeluaran



Gambar 3 – Halaman Kategori Pemasukan dan Pengeluaran

1.3.2. Potongan Source Code Kategori Pemasukan dan Pengeluaran

```
1 import React, { useEffect, useState, useContext } from 'react';
2 import { Text, View, StyleSheet, TouchableOpacity, Image, ScrollView, Alert, ToastAndroid } from 'react-native';
3 import { theme } from '../assets/themes/themes';
4 import { CategorySelection } from '../components/CategorySelection';
5 import { AxiosContext } from '../context/AxiosContext';
6 import { AuthContext } from '../context/AuthContext';
7 import Ionicons from 'react-native-vector-icons/Ionicons';
8 import { Spinner } from './Spinner';
9 import { DeleteCategory } from '../service/ApiServices';
10 import {
11   formatCurrency,
12 } from 'react-native-format-currency';
13 export const CategoryScreen = ({ navigation, route }) => {
14   const authContext = useContext(AuthContext);
15   const { publicAxios } = useContext(AxiosContext);
16   const [isLoading, setIsLoading] = useState(true);
17   const categoryState = authContext?.categoryState;
18   useEffect(() => {
19     const fetchCategory = async () => {
20       try {
21         const accessToken = authContext?.authState?.token;
22         if (accessToken) {
23           const response = await publicAxios.get('user/category');
24           if (response.status == 200) {
25             authContext.setCategoryState(response.data)
26           }
27         }
28       }
29     }
30     fetchCategory();
31   });
32   return (
33     <View style={styles.container}>
34       <CategorySelection />
35     </View>
36   );
37 }
```

```
27         } else {
28             throw Error(response.msg);
29         }
30     } else {
31         authContext.logout();
32     }
33 } catch (error) {
34     setIsLoading(false)
35     authContext.logout();
36     console.log('error screen index ' + error);
37 }
38 };
39 fetchCategory();
40 }, [route]);
41 const handlePress = ({ id, title, nameIcon, type, budget }) => {
42     navigation.navigate('Transaction', {
43         id,
44         nama: title,
45         icon: nameIcon,
46         budget,
47         type
48     });
49 };
50 const formateIdr = (amount) => {
51     const [valueFormattedWithSymbol, valueFormattedWithoutSymbol, symbol] =
52         formatCurrency({
53             amount: Number(amount),
54             code: 'IDR',
55         });
56     return valueFormattedWithSymbol
57 }
```

```
59 const handleDelete = async (id) => {
60     DeleteCategory(id, authContext, publicAxios)
61     .then((result) => {
62         ToastAndroid.showWithGravityAndOffset(
63             result,
64             ToastAndroid.SHORT,
65             ToastAndroid.BOTTOM,
66             25,
67             50
68         )
69     })
70     .catch((err) => {
71         console.log(err);
72         Alert.alert('Failed', 'Failed to Delete category')
73     })
74 }
75 const showConfirmDialog = (idCategory, categoryName) => {
76     Alert.alert(
77         'Delete Item',
78         'Anda yakin akan menghapus category ${categoryName}?',
79         [
80             {
81                 text: 'batal',
82                 onPress: () => console.log('Cancel Pressed'),
83                 style: 'cancel',
84             },
85             { text: 'OK', onPress: () => handleDelete(idCategory) },
86         ],
87         { cancelable: false },
88     );
89 };
```

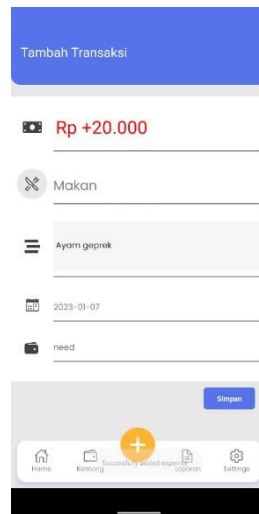
```

181 |   const handleUpdate = ({ id, title, nameIcon, type, budget }) => {
182 |     navigation.navigate('PostCategoryStack', { id, name: title, icon: nameIcon, budget, type });
183 |   }
184 |
185 |   const handlePocket = ({ idPocket, namePocket, targetAmount, amount }) => { navigation.navigate('Transaction', { idPocket, namePocket, targetAmount, amount }); }
186 |
187 |   const handleDefaultPocket = () => { navigation.navigate('Transaction', { idPocket: null, }); }
188 |
189 |   const handlePostCategory = () => { navigation.navigate('PostCategoryStack'); }
190 |
191 |   const expenseCategory = categoryState.filter((value) => value.budgeting !== '50/30/20' && value.budgeting !== '70/10/10/10').map((value) => value)
192 |   const incomeCategory = categoryState.filter((value) => value.budgeting === '50/30/20' || value.budgeting === '70/10/10/10').map((value) => value);
193 |
194 |   if (isLoading) {
195 |     return <Spinner />
196 |   } else {
197 |     if (route?.params?.pocket === false) {
198 |       return (...
199 |     );
200 |   } else {
201 |     return (
202 |       <ScrollView style={{ paddingTop: theme.spacing.1, }}>
203 |         <Text style={{ ...theme.textVariants.header, color: theme.colors.fontColor, fontSize: 15 }}> List Tabungan</Text>
204 |         <View style={style.containerPocket}>
205 |           {authContext.pocketState && authContext.pocketState.map((data, index) => (
206 |             <TouchableOpacity key={data.idPocket} style={{ flex: 1, flexDirection: 'row' }} onPress={() => handlePocket(data)}>
207 |               <View>
208 |                 <Text style={{ ...theme.textVariants.body, color: theme.colors.fontColor }}>{data.namePocket}</Text>
209 |                 <Text style={{ ...theme.textVariants.body, color: theme.colors.fontColor, fontSize: 12 }}> {formatIdr(data.amount)}</Text>
210 |               </View>
211 |             </TouchableOpacity>
212 |           ))}
213 |         </View>
214 |       </ScrollView>
215 |     );
216 |   }
217 | }
218 |
219 | const style = StyleSheet.create({
220 |   container: { backgroundColor: '#fff', flexDirection: 'row', alignItems: 'center', alignItems: 'center', paddingVertical: 7, borderBottomWidth: 1,
221 |     borderBottomColor: 'gray', },
222 |   wrapIcon: { marginHorizontal: 15, backgroundColor: 'red', padding: 7, borderRadius: 50, },
223 |   containerPocket: { backgroundColor: '#fff', flexDirection: 'row', alignContent: 'center', alignItems: 'center', borderBottomWidth: 1, borderBottomColor: 'gray',
224 |     minHeight: 50,
225 |     marginTop: 3,
226 |     padding: 10 },
227 |   wrapIconPocket: { marginHorizontal: 15, backgroundColor: 'gray', height: 50, width: 50, borderRadius: 50, alignContent: 'center', justifyContent: 'center',
228 |     alignItems: 'center'
229 |   },
230 | });

```

1.4 Halaman Transaksi Pemasukan atau Pengeluaran

1.4.1 Tampilan Halaman Transaksi Pemasukan atau Pengeluaran



Gambar 4 – Halaman Transaksi

1.4.2 Source Code Halaman Dashboard Admin

```
1 import React, { useState, useEffect, useContext } from 'react';
2 import {
3   ScrollView, Text, View, TextInput, StyleSheet, TouchableOpacity, ToastAndroid, Alert,
4 } from 'react-native';
5 import { theme } from '../assets/themes/themes';
6 import DateTimePickerModal from 'react-native-modal-datetime-picker';
7 import moment from 'moment/moment';
8 import Ionicons from 'react-native-vector-icons/Ionicons';
9 import IconFontAwesome from 'react-native-vector-icons/FontAwesome5';
10 import { AuthContext } from '../context/AuthContext';
11 import { AxiosContext } from '../context/AxiosContext';
12 import { PostTransaction } from '../service/Post';
13 import CurrencyInput from 'react-native-currency-input';
14
15 export const PostScreen = ({ navigation, route }) => {
16   const authContext = useContext(AuthContext);
17   const { publicAxios } = useContext(AxiosContext);
18   const [amount, setAmount] = useState(0);
19   const [category, setCategory] = useState('Pilih Kategori');
20   const [note, setNote] = useState('');
21   const [date, setDate] = useState(moment().format('YYYY-MM-DD'));
22   const [isDatePickerVisible, setDatePickerVisibility] = useState(false);
23   const [isDisable, setIsDisable] = useState(false);
24   const [budgetShow, setBudgetShow] = useState("Otomatis sesuai kategori");
25   const saldoState = authContext.saldoState
26   const pocketState = authContext.pocketState
27
28   useEffect(() => {
29     try {
30       if (route?.params?.id !== undefined) {
31         setCategory({ id: route.params.id, name: route.params.nama, icon: route.params.icon, budgeting: route.params.budget, type: route.params.type });
32       }
33     }
34   });
35 }
```

```

32     setBudgetShow(route.params.budget)
33   } else if (route?.params?.idPocket !== undefined) {
34     if (route?.params?.idPocket === null) {
35       setBudgetShow('Saldo Utama Tabungan')
36     } else {
37       setBudgetShow(`kantong ${route.params.namePocket}`)
38     }
39   } else {
40     setAmount(0)
41     setCategory('Pilih Kategori')
42     setNote('')
43     setDate(moment({}).format('YYYY-MM-DD'))
44     setBudgetShow("Otomatis sesuai kategori")
45   }
46   } catch (err) {
47     console.log(err)
48   }
49 }, [route]);
50
51 const showDatePicker = () => { setDatePickerVisibility(true); };
52 const hideDatePicker = () => { setDatePickerVisibility(false); };
53 const handleConfirm = (date) => {
54   setDate(moment(date).format('YYYY-MM-DD'));
55   hideDatePicker();
56 };
57
58 const handleSave = async () => {
59   setIsDisable(!isDisable)
60   if (category.budgeting === 'saving') {
61     if (route.params.idPocket === undefined) {
62       Alert.alert('Failed', 'Silahkan pilih kantong tabungan');
63     }
64     if (route.params.idPocket === null) {
65       const pocket = pocketState.reduce((acc, cur) => acc + cur.amount, 0)

```

```

64     if (amount <= (saldoState?.saldoSaving - pocket))
65       PostTransaction(amount, category, note, date, publicAxios, authContext).then((result) => {
66         ToastAndroid.showWithGravityAndOffset(result, ToastAndroid.SHORT, ToastAndroid.BOTTOM, 25, 50)
67         setAmount(0)
68         setCategory('Pilih Kategori')
69         setNote('')
70         setDate(moment({}).format('YYYY-MM-DD'))
71         publicAxios.get('user/transaction').then((result) => {
72           const transaction = result.data.sort((a, b) => {
73             const dateA = new Date(a.createdAt);
74             const dateB = new Date(b.createdAt);
75             return dateB - dateA;
76           })
77           authContext.setHistoryState(transaction);
78         })
79       })
80     }).catch((err) => {
81       console.log(err);
82       Alert.alert('Failed', 'Something Error');
83     }).finally(() => setIsDisable(false))
84   } else {
85     Alert.alert('', 'Saldo utama tabungan tidak mencukupi');
86     return
87   }
88 } else {
89   const pocket = pocketState.filter((value) => value.idPocket === route.params.idPocket)[0]
90   if (amount > pocket.amount) {
91     Alert.alert('', 'Saldo kantong tabungan tidak mencukupi');
92     return
93   }
94   const response = await publicAxios.put('user/pocket/${route.params.idPocket}', {
95     namePocket: pocket.namePocket,

```



```

96         targetAmount: pocket.targetAmount,
97         amount: amount
98     })
99     if (response.status !== 200) {
100         throw 'Gagal menambah data';
101     }
102     pocket.amount = amount
103     const updatePocket = pocketState.filter((value) => value.idPocket !== route.params.idPocket)
104     authContext?.setPocketState([pocket, ...updatePocket,])
105     PostTransaction(amount, category, note, date, publicAxios, authContext).then((result) => {
106         ToastAndroid.showWithGravityAndOffset(result, ToastAndroid.SHORT, ToastAndroid.BOTTOM, 25, 50 )
107         setAmount(0)
108         setCategory('Pilih Kategori')
109         setNote('')
110         setDate(moment({}).format('YYYY-MM-DD'))
111         publicAxios.get('user/transaction').then((result) => {
112             const transaction = result.data.sort((a, b) => {
113                 const dateA = new Date(a.createdAt);
114                 const dateB = new Date(b.createdAt);
115                 return dateB - dateA;
116             })
117             authContext.setHistoryState(transaction);
118         })
119     }).catch((err) => {
120         console.log(err);
121         Alert.alert('Failed', 'Something Error');
122     }).finally(() => setIsDisable(false))
123 }
124 } else {
125     PostTransaction(amount, category, note, date, publicAxios, authContext).then((result) => {
126         ToastAndroid.showWithGravityAndOffset(result, ToastAndroid.SHORT, ToastAndroid.BOTTOM, 25, 50 )

```

```

127         setAmount(0)
128         setCategory('Pilih Kategori')
129         setNote('')
130         setDate(moment({}).format('YYYY-MM-DD'))
131         publicAxios.get('user/transaction').then((result) => {
132             const transaction = result.data.sort((a, b) => {
133                 const dateA = new Date(a.createdAt);
134                 const dateB = new Date(b.createdAt);
135                 return dateB - dateA;
136             })
137             authContext.setHistoryState(transaction);
138         })
139     }).catch((err) => {
140         console.log(err);
141         Alert.alert('Failed', 'Something Error');
142     }).finally(() => setIsDisable(false))
143 }
144 }
145 }
146 const handleCategory = (bool) => { navigation.navigate('CategoryStack', { pocket: bool });};
147 const budget = { 'need': 'Kebutuhan', 'want': 'Keinginan', 'invest': 'Investasi', 'saving': 'Tabungan', '70/10/10/10': '70/10/10/10', '50/30/20': '50/30/20' }
148 return (
149     <ScrollView style={{ backgroundColor: theme.colors.background }}>
150         <View style={{ backgroundColor: theme.colors.primary, padding: theme.spacing.m, paddingVertical: theme.spacing.xl, borderBottomRightRadius: 15,
151             borderBottomLeftRadius: 15}}>
152             <Text style={{ ...theme.textVariants.header, color: theme.colors.foreground, marginTop: 20, }}> Tambah Transaksi </Text>
153         </View>
154         <View style={{ backgroundColor: 'white', marginTop: 20, paddingBottom: 20, }}>
155             <View style={style.containerItem}>
156                 <View style={style.containerIcon}><IconFontAwesome name="money-bill" size={25} color={theme.colors.fontColor} light /></View>
157                 <View style={{ ...style.containerInput, flexDirection: 'row', }} >
158                     <CurrencyInput value={amount} onChangeValue={setAmount} prefix="Rp " delimiter="." separator="," precision={0} minValu={0} showPositiveSign

```



```

160         style={{ ...style.textInput, color: category?.type === 'income' ? 'green' : 'red', }} />
161     </View>
162 </View>
163     { /* Category select */ }
164     <View style={style.containerItem}>
165         <View style={{ ...style.containerIcon, backgroundColor: theme.colors.background, }}>
166             <Ionicons name={category.icon === undefined ? "help-outline" : category.icon} size={25} color={theme.colors.fontColor} />
167         </View>
168         <View style={style.containerInput}>
169             (property) TextInput: ViewStyle | TextStyle | ImageStyle
170             <TouchableOpacity style={{ ...style.textInput, paddingBottom: 0 }} onPress={() => handleCategory(false)} >
171                 <Text style={{ ...theme.textVariants.body, fontSize: 25, color: 'gray', }}> {category.name === undefined ? category : category.name}</Text>
172             </TouchableOpacity>
173         </View>
174     </View>
175     { /* Keterangan Input */ }
176     <View style={style.containerItem}>
177         <View style={style.containerIcon}> <IonFontAwesome name='stream' size={25} color={theme.colors.fontColor} /> </View>
178         <View style={{ ...style.containerInput }}>
179             <TextInput value={note} onChangeText={(text) => setNote(text)} style={{ ...theme.textVariants.body, ...style.text, padding: 0, paddingLeft: 4,
180                 height: 80, textAlignVertical: 'center', backgroundColor: '#F6F6F6' }}
181                 underlineColorAndroid="transparent"
182                 placeholder="Catatan"
183                 placeholderTextColor="gray"
184                 multiline={true}/>
185         </View>
186     </View>
187     { /* Dates input */ }
188     <View style={style.containerItem}>
189         <View style={style.containerIcon}> <Ionicons name='calendar-outline' size={25} color={theme.colors.fontColor} /></View>
190         <View style={style.containerInput}>
191             <TouchableOpacity style={{ ...style.textInput, paddingBottom: 0 }} onPress={showDatePicker} >
192                 <Text style={{ ...theme.textVariants.body, color: 'gray', }}> {date}</Text>

```

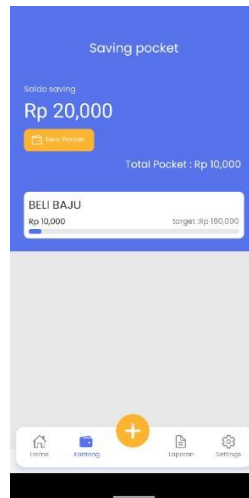
```

192             <DatePickerModal isVisible={isDatePickerVisible} mode="date" onConfirm={handleConfirm} onCancel={hideDatePicker}/>
193         </TouchableOpacity>
194     </View>
195 </View>
196     { /* Pocket select */ }
197     <View style={style.containerItem}>
198         <View style={style.containerIcon}> <Ionicons name='wallet' size={25} color={theme.colors.fontColor} /></View>
199         <TouchableOpacity style={style.containerInput} disabled={category.budgeting === 'saving' ? false : true} onPress={() => handleCategory(true)}>
200             <Text style={{ ...theme.textVariants.body, color: 'gray' }}> {budgetShow === 'saving' ? 'Pilih Tabungan : ' : budget[budgetShow]} </Text>
201         </TouchableOpacity>
202     </View>
203 </View>
204 <View style={{ flex: 1, width: '20%', alignSelf: 'flex-end', marginHorizontal: 20, marginVertical: 10, }} >
205     <TouchableOpacity style={{ backgroundColor: theme.colors.primary, flex: 4, alignItems: 'center', alignContent: 'center', padding: 10, borderRadius: 5, }}
206         disabled={false}
207         onPress={() => handleSave()} >
208         <Text style={{ fontWeight: 'bold', color: '#fff' }}> Simpan </Text>
209     </TouchableOpacity>
210 </View>
211 </ScrollView>
212 );
213 };
214
215 const style = StyleSheet.create({
216     textInput: { paddingVertical: 10, fontSize: 25, ...theme.textVariants.header, fontSize: 35, },
217     containerItem: { flexDirection: 'row', ...theme.centering, },
218     containerIcon: { height: 50, width: 50, borderRadius: 50, alignItems: 'center', justifyContent: 'center', marginRight: 10, },
219     containerInput: { flex: 1, paddingVertical: 10, borderBottomWidth: 1, borderBottomColor: theme.colors.fontColor, },
220     text: { color: theme.colors.fontColorDark, },
221 });

```

1.5 Halaman Kantong Tabungan

1.5.1 Tampilan Halaman Kantong Tabungan



Gambar 5 – Halaman Kantong Tabungan

1.5.2 Source Code Tampilan Kantong Tabungan

```
1 import { useContext, useEffect, useState } from 'react';
2 import { Text, TouchableOpacity, View, ScrollView, TextInput, StyleSheet, Alert } from 'react-native';
3 import { theme } from '../assets/themes/themes';
4 import { FormatCurrency } from 'react-native-format-currency';
5 import { AuthContext } from '../context/AuthContext';
6 import { AxiosContext } from '../context/AxiosContext';
7 import Ionicons from 'react-native-vector-icons/Ionicons';
8
9 export const PocketsScreen = ({ navigation, route }) => {
10   const authContext = useContext(AuthContext);
11   const { publicAxios } = useContext(AxiosContext);
12   const formatIdr = (amount) => {
13     const [valueFormattedWithSymbol, valueFormattedWithoutSymbol, symbol] =
14       formatCurrency({ amount: Number(amount), code: 'IDR', });
15     return valueFormattedWithSymbol;
16   }
17   const pocketState = authContext.pocketState;
18   const [isShowDetail, setIsShowDetail] = useState({ isShow: false, id: null,});
19   const [newAmount, setNewAmount] = useState(0);
20   const [isDisable, setIsDisable] = useState(false);
21   useEffect(() => {
22     const fetctAllPocket = async () => {
23       try {
24         const response = await publicAxios.get('user/pocket');
25         if (response.status !== 200) {
26           throw response.data.message;
27         }
28         authContext?.setPocketState(response.data)
29       } catch (err) {
30         console.log(err);
31       }
32     };
33   });
34 }
```

```

33     fetchAllPocket();
34   }, []);
35
36   const countProgress = (amount, target) => `${(amount / target) * 100}%`
37
38   const handlePress = (id) => {
39     setNewAmount(0);
40     setIsShowDetail({ isShow: !isShowDetail.isShow, id: id, })
41   }
42
43   const handlePostPocket = () => {
44     navigation.navigate('PostPocketStack');
45   }
46
47   const handleDeletePocket = async (id) => {
48     try {
49       setIsDisable(true)
50       const response = await publicAxios.delete(`user/pocket/${id}`)
51       if (response.status !== 200) {
52         throw response.data.message
53       }
54       const responseB = await publicAxios.get('user/pocket');
55       if (responseB.status !== 200) {
56         throw responseB.data.message;
57       }
58
59       authContext?.setPocketState([...pocketState.filter((value) => value.idPocket !== id)])
60     } catch (err) {
61       console.log(err)
62     } finally {
63       setIsDisable(false)
64       setIsShowDetail({ isShow: !isShowDetail.isShow, id: id, })
65     }
66   }
67
68   const handleAddAmount = async (id) => {
69     try {
70       setIsDisable(true)
71       const pocket = pocketState.filter((value) => value.idPocket === id)[0]
72       const response = await publicAxios.put(`user/pocket/${id}`, { namePocket: pocket.namePocket, targetAmount: pocket.targetAmount,
73         amount: (parseInt(pocket.amount) + parseInt(newAmount))
74       })
75       if (response.status !== 200) {
76         throw 'Gagal menambah data';
77       }
78       pocket.amount = parseInt(pocket.amount) + parseInt(newAmount)
79       const updatePocket = pocketState.filter((value) => value.idPocket !== id)
80       authContext?.setPocketState([pocket, ...updatePocket,])
81     } catch (err) {
82       Alert.alert('Error', 'Gagal menambah jumlah kantong')
83       console.log('err', err.message);
84     } finally {
85       setIsDisable(false)
86       setIsShowDetail({ isShow: !isShowDetail.isShow, id: id, })
87     }
88   }
89
90   return (
91     <ScrollView style={{ flex: 1, backgroundColor: theme.colors.background, marginBottom: theme.spacing.x1 }} >
92       <View style={{ flex: 1, backgroundColor: theme.colors.primary, padding: theme.spacing.m, paddingHorizontal: theme.spacing.l, height: 420,}} >
93         <View style={{marginTop: theme.spacing.x1,justifyContent: 'center', alignContent: 'center',alignItems: 'center', }} >
94           <Text style={{...theme.textVariants.body,textAlign: 'center',fontSize: 25,color: 'white', }}> Saving pocket</Text>
95         </View>
96       </View>

```

```

97 <Text style={{ ...theme.textVariants.body,color: theme.colors.background, }}> Saldo saving</Text>
98 <Text style={{ ...theme.textVariants.header, color: theme.colors.Foreground, fontSize: 40, }}> {formatIdr(authContext.saldoState.saldoSaving)}</Text>
99 <TouchableOpacity disabled={isDisable} onPress={handlePostPocket} style={{ padding: 8, paddingHorizontal: 5, marginVertical: 10, backgroundColor: theme.color
100 <Icons name='wallet-outline' size={20} color={'#fff'} />
101 <Text style={{ ...theme.textVariants.body, color: '#fff', fontSize: 13, marginHorizontal: 5 }}>New Pocket</Text>
102 </TouchableOpacity>
103 <View style={{ alignItems: 'flex-end' }}>
104 <Text style={{ ...theme.textVariants.body,fontSize: 20,color: '#fff' }}>
105 Total Pocket : {formatIdr(pocketState.reduce(acc, cur) => acc + cur.amount, 0)} </Text>
106 </View>
107 </View>
108 </View>
109 <View style={{ padding: theme.spacing.s, paddingHorizontal: theme.spacing.l}}>
110 <View style={{ marginTop: -110 }}> {pocketState && pocketState.map((data, index) => (
111 <TouchableOpacity disabled={isDisable}onPress={() => handlePress(data.idPocket)} key={index}style={{ width: '100%', backgroundColor: '#fff',
112 borderRadius: 8, padding: theme.spacing.s,marginBottom: 5 }} >
113 <Text style={{...theme.textVariants.body,color: theme.colors.fontColorDark,fontSize: 23}}>{data.namePocket}</Text>
114 <View style={{ flexDirection: 'row' }}>
115 <View style={{ flex: 1 }}>
116 <Text style={{...theme.textVariants.body,color: theme.colors.fontColorDark, }}>{formatIdr(data.amount)} </Text>
117 </View>
118 <View style={{ flex: 1 }}>
119 <Text style={{...theme.textVariants.body,color: 'gray',textAlign: 'right',fontSize: 15,}} >target :{formatIdr(data.targetAmount)}</Text>
120 </View>
121 </View>
122 <View style={{ backgroundColor: theme.colors.background,height: 8,borderRadius: 8,}}>
123 <View style={{backgroundColor: theme.colors.primary,height: 8, width: countProgress(data.amount, data.targetAmount),borderRadius: 8,}} ></View>
124 </View>
125 {isShowDetail.isShow && isShowDetail.id === data.idPocket ? (
126 <View style={{ flex: 1, flexDirection: 'row' }}>
127 <View style={styles.textInputWrap}>
128 <Text style={{color: 'gray',fontSize: 15,width: '100%'}}>Rp.</Text>
129 <TextInput value={newAmount} onChangeText={(text) => setNewAmount(text)} keyboardType="numeric"style={styles.textInput}/>
130 <TouchableOpacity disabled={isDisable} style={{backgroundColor: theme.colors.primary, alignContent: 'center',alignItems: 'center',
131 justifyContent: 'center',
132 width: '20%',
133 borderRadius: 50,
134 borderWidth: 0.4,
135 borderColor: theme.colors.primary
136 }}
137 onPress={() => handleAddAmount(data.idPocket)}>
138 <Text style={{...theme.textVariants.body,color: '#fff',fontSize: 12,textAlign: 'center',}}>Tambah</Text>
139 </TouchableOpacity>
140 </View>
141 <View style={{marginTop: 10,flexDirection: 'row',justifyContent: 'flex-end',alignContent: 'center',alignItems: 'center',width: 60,
142 marginRight: 10 }}>
143 <TouchableOpacity disabled={isDisable} style={styles.btnList} onPress={() => { handleDeletePocket(data.idPocket) }}>
144 <Icons name='trash-outline' size={30} color={theme.colors.danger} />
145 </TouchableOpacity>
146 </View>
147 </View> : ''
148 </TouchableOpacity>
149 </View>
150 </View>
151 </ScrollView>
152 </View>
153 </View>
154 </View>
155 </View>
156 </View>
157 const styles = StyleSheet.create({
158 btnList: { padding: 7, height: 50, },
159 textInput: {padding: 0.5,fontSize: 15,width: '70%',color: theme.colors.fontColorDark},
160 textInputWrap: { flex: 1, marginTop: 15,borderWidth: 0.3,borderRadius: 50,flexDirection: 'row', height: 35,justifyContent: 'center',width: '100%',paddingLeft: 10,}

```

```

129 |         <TextInput value={newAmount} onChangeText={({text} => setNewAmount(text)} keyboardType="numeric" style={styles.textInput}/>
130 |         <TouchableOpacity disabled={isDisable} style={{backgroundColor: theme.colors.primary, alignContent: 'center', alignItems: 'center',
131 |             justifyContent: 'center',
132 |             width: '20%',
133 |             borderRadius: 50,
134 |             borderWidth: 0.4,
135 |             borderColor: theme.colors.primary
136 |         }}
137 |         onPress={() => handleAddAmount(data.idPocket)}>
138 |             <Text style={{...theme.textVariants.body,color: '#fff',fontSize: 12,textAlign: 'center'}}>Tambah</Text>
139 |         </TouchableOpacity>
140 |     </View>
141 |     <View style={{marginTop: 10,flexDirection: 'row',justifyContent: 'flex-end',alignContent: 'center',alignItems: 'center',width: 60,
142 |         marginRight: 10 }}>
143 |         <TouchableOpacity disabled={isDisable} style={styles.btnList} onPress={() => { handleDeletePocket(data.idPocket) }}>
144 |             <Ionicons name={'trash-outline'} size={30} color={theme.colors.danger} />
145 |         </TouchableOpacity>
146 |     </View>
147 |     </View> : ''
148 | </TouchableOpacity>
149 |     ))
150 | </View>
151 | </View>
152 | </ScrollView>
153 |
154 | );
155 |
156 | const styles = StyleSheet.create({
157 |     btnList: { padding: 7, height: 50, },
158 |     textInput: {padding: 0.5,fontSize: 15,width: '70%',color: theme.colors.fontColorDark},
159 |     textInputWrap: { flex: 1, marginTop: 15,borderWidth: 0.3,borderRadius: 50,flexDirection: 'row', height: 35,justifyContent: 'center',width: '100%',paddingLeft: 10,}
160 | })

```

1.6 Halaman Laporan

1.6.1 Tampilan Halaman Laporan



Gambar 6 – Tampilan Halaman Laporan

1.6.2 Source Code Tampilan Halaman Laporan

```
1 import React, { useState, useEffect, useContext } from 'react';
2 import { Text, View, TouchableOpacity, StyleSheet, ScrollView } from 'react-native';
3 import { theme } from '../assets/themes/themes';
4 import { AuthContext } from '../context/AuthContext';
5 import Modal from 'react-native-modal';
6 import HistoryComponent from '../components/HistoryComponent';
7 import { VictoryPie } from 'victory-native';
8 import { FormatCurrency } from 'react-native-format-currency';
9 import { month, years } from '../other/date';
10
11 export const HistoryScreen = ({ navigation }) => {
12   const authContext = useContext(AuthContext);
13   const historyState = authContext.historyState;
14   const [isModalVisible, setModalVisible] = useState(false);
15   const [modalTitle, setModalTitle] = useState(null);
16   const [historyByDate, setHistoryByDate] = useState(historyState);
17   const [monthPick, setMonthPick] = useState(new Date().getMonth());
18   const [yearPick, setYearPick] = useState(new Date().getFullYear());
19   useEffect(() => {
20     const newHistory = historyState.filter((data) => {
21       const monthHistory = new Date(data.date).getMonth();
22       const yearHistory = new Date(data.date).getFullYear();
23       return monthHistory === monthPick && yearHistory === yearPick;
24     });
25     setHistoryByDate(newHistory);
26   }, [historyState]);
27   const data = [
28     {
29       x: 'Pemasukan',
30       y: historyByDate
31         .filter((history) => history.type === 'income')
32         .reduce((acc, value) => acc + value.amount, 0),
33     },
34   ],
```

```

33     },
34     {
35       x: 'Pengeluaran',
36       y: historyByDate
37         .filter((history) => history.type === 'expense')
38         .reduce((acc, value) => acc + value.amount, 0),
39     },
40 ];
41 const formatIdr = (amount) => {
42   const [valueFormattedWithSymbol, valueFormattedWithoutSymbol, symbol] = formatCurrency({ amount: Number(amount), code: 'IDR' });
43   return valueFormattedWithSymbol;
44 };
45 const toggleModal = (title) => { setModalVisible(!isModalVisible); setModalTitle(title); };
46 const handleSelectMonth = (index) => { setMonthPick(index); };
47 const handleSelectYear = (index) => { setYearPick(years[index]); };
48
49 const handleConfirm = () => {
50   const newHistory = historyState.filter((data) => {
51     const monthHistory = new Date(data.date).getMonth();
52     const yearHistory = new Date(data.date).getFullYear();
53     return monthHistory === monthPick && yearHistory === yearPick;
54   });
55   setHistoryByDate(newHistory);
56   setModalVisible(!isModalVisible);
57 };
58
59 const handleDetailTransaction = (idTransaction, amount, date, type, note, tb_category) => {
60   navigation.navigate('DetailTransactionScreen', { data: { idTransaction, amount, date, type, note, tb_category } });
61 };
62
63 return (
64   <ScrollView style={{ padding: theme.spacing.s }}>
65     <View style={{ flexDirection: 'row', alignItems: 'center', }} >
66       <TouchableOpacity onPress={() => toggleModal('Bulan')} style={styles.btn} > <Text style={{ color: '#fff' }}>Bulan</Text> </TouchableOpacity>
67       <TouchableOpacity onPress={() => toggleModal('Tahun')} style={styles.btn} > <Text style={{ color: '#fff' }}>Tahun</Text> </TouchableOpacity>
68     <Modal isVisible={isModalVisible}>
69       <View style={{ backgroundColor: '#fff', padding: theme.spacing.s, }} >
70         <Text style={{ textAlign: 'center', marginBottom: 20 }}> {modalTitle} </Text>
71         <View style={{ flexDirection: 'row', flexWrap: 'wrap', justifyContent: 'center' }} >
72           {modalTitle === 'Bulan'
73             ? month.map((data, index) => (
74               <TouchableOpacity
75                 key={index}
76                 onPress={() => handleSelectMonth(index)}
77                 style={{ padding: 10, backgroundColor: month[monthPick] === data ? 'gray' : '#d4d3cf', justifyContent: 'center', margin: 3, width: 90 }} >
78                 <Text>{data}</Text>
79               </TouchableOpacity>
80             ))
81             : years.map((data, index) => (
82               <TouchableOpacity key={index} onPress={() => handleSelectYear(index)}
83                 style={{ padding: 10, backgroundColor: yearPick === data ? 'gray' : '#b0b0ae', justifyContent: 'center', margin: 3, }} >
84                 <Text>{data}</Text>
85               </TouchableOpacity>
86             ))
87           )
88         </View>
89       </View>
90       <View style={{ flexDirection: 'row', justifyContent: 'flex-end', marginTop: 15, }} >
91         <TouchableOpacity onPress={() => setModalVisible(!isModalVisible)} style={{ padding: 10, justifyContent: 'center', }} >
92           <Text style={{ color: theme.colors.fontColorDark, }} > Cancel </Text>
93         </TouchableOpacity>
94         <TouchableOpacity onPress={handleConfirm} style={{ marginLeft: 10, padding: 10, justifyContent: 'center' }} >
95           <Text style={{ color: theme.colors.fontColorDark, }} > Ok </Text>
96         </TouchableOpacity>
97       </View>
98     </Modal>
99   </ScrollView>
100 );

```

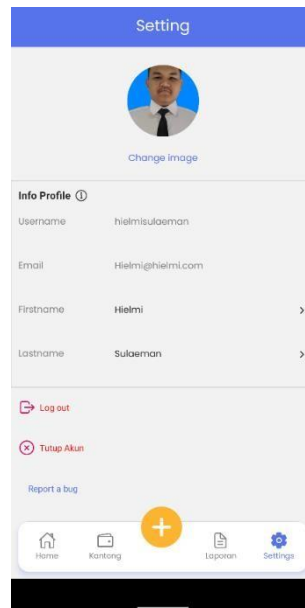
```

127     </View>
128   </View>
129 </Modal>
130 </View>
131 <View style={{ backgroundColor: '#fff', borderRadius: 15, padding: theme.spacing.s, }} >
132   <Text style={{ ...theme.textVariants.body, color: theme.colors.fontColorDark, }} > {'Periode : ' + month[monthPick] + ' ' + yearPick} </Text>
133   {historyByDate.length >= 1 ? (
134     <VictoryPie colorScale={['#82CD47', '#DC3535']} data={data} height={300} />
135   ) : (
136     <VictoryPie colorScale={['#82CD47', '#DC3535']} data={[{ x: 1, y: 1, label: 'Pengeluaran' }, { x: 2, y: 1, label: 'Pemasukan' }, ]} height={300} />
137   )
138   <Text style={{ color: theme.colors.fontColorDark }}> Total Pemasukan : {formatIdr(data[0].y)} </Text>
139   <Text style={{ color: theme.colors.fontColorDark }}> Total Pengeluaran : {formatIdr(data[1].y)} </Text>
140 </View>
141 <Text style={{ ...theme.textVariants.body, color: theme.colors.fontColorDark, marginTop: 30, }}>Riwayat Transaction</Text>
142 <View style={{ Flex: 1, paddingHorizontal: 10 }}>
143   <View style={{ marginBottom: 100 }}>
144     {historyByDate.length >= 1 ? (
145       historyByDate.map((history, index) => {
146         if (index < 10) {
147           return (
148             <HistoryComponent key={index} idTransaction={history.idTransaction} amount={history.amount} date={history.date} type={history.type}
149               note={history.note}
150               tb_category={history.tb_category}
151               handler={handleDetailTransaction} /> );
152         }
153       })
154     ) : (
155       <View style={{ backgroundColor: theme.colors.foreground, padding: theme.spacing.s, borderRadius: 15, marginTop: 5, }} >
156         <Text style={{ color: theme.colors.fontColorDark }}>
157           {' '}
158         </Text>
159         Tidak ada history
160       </View>
161     )
162   </View>
163 </View>
164 </ScrollView>
165 );
166 };
167
168 const styles = StyleSheet.create({
169   btn: { paddingVertical: 10, paddingHorizontal: 20, backgroundColor: theme.colors.secondary, justifyContent: 'center', margin: 5, borderRadius: 7, },
170 });

```


1.7 Halaman Setting

1.7.1 Tampilan Halaman Setting



Gambar 7 – Tampilan Halaman Setting

1.7.2 Source Code Halaman Setting

```
1 import React, { useState, useEffect, useContext } from 'react';
2 import { Text, View, TouchableOpacity, Image, StyleSheet, ScrollView, Button, Alert, Platform, Dimensions, Linking } from 'react-native';
3 import { AuthContext } from '../context/AuthContext';
4 import { AxiosContext } from '../context/AxiosContext';
5 import { launchImageLibrary } from 'react-native-image-picker';
6 import { theme } from '../assets/themes/themes';
7 import Icon from 'react-native-vector-icons/Ionicons';
8 import { Spinner } from './Spinner';
9
10 export const SettingsScreen = ({ navigation, route }) => {
11   const authContext = useContext(AuthContext);
12   const { publicAxios, BASE_URL } = useContext(AxiosContext);
13   const [imgUrl, setImgUrl] = useState(`${BASE_URL}images/${authContext.userState.photoProfile}`);
14   const [isLoading, setIsLoading] = useState(false);
15
16   const selectImage = async () => {
17     try {
18       setIsLoading(true)
19       const result = await launchImageLibrary({
20         mediaType: 'photo',
21       })
22     }
23     if (result.didCancel === true) {
24       return;
25     }
26     const newPhoto = result.assets.map((value) => value)[0]
27
28     // update photoProfile
29     const formData = new FormData();
30     formData.append('photoProfile', { name: newPhoto.fileName, type: newPhoto.type,
31       uri: Platform.OS === 'android' ? newPhoto.uri : newPhoto.uri.replace("file://", "")
32     })
33   }
34 }
```

```

33     const config = { method: 'PATCH', headers: {
34         'Accept': 'application/json',
35         'Content-Type': 'multipart/form-data',
36         'x-access-token': authContext?.authState?.token
37     },
38     body: formData,
39 };
40 fetch(`${BASE_URL}user`, config)
41     .then((checkStatusAndGetJSONResponse) => {
42         setImgUrl(newPhoto.uri)
43         publicAxios.get('user').then(user => { const { email, firstname, lastname, photoProfile, username } = user.data
44             authContext.setUserState({ email, firstname, lastname, photoProfile, username })
45         }).catch(err => console.log(err))
46     }).catch(err => {
47         setIsLoading(false)
48         Alert.alert('Failed', 'Gagal untuk upload foto')
49         console.log(err)
50     })
51     .finally((result) => { setIsLoading(false)})
52 } catch (err) {
53     Alert.alert('Failed', err)
54     console.log(err)
55 }
56 }
57
58 const handleUbahProfile = (title, value) => { navigation.navigate('EditUserScreen', { title, value })}
59 const logOut = () => { authContext.logout()}
60
61 const handleDeleteUser = async () => {
62     try {
63         const response = await publicAxios.delete('user')
64         if (response.status !== 200) {
65             throw 'Failed to remove user'
66         }
67         logOut()
68     } catch (err) { console.log(err);}
69 }
70
71
72 const showConfirmDialog = () => {
73     Alert.alert(
74         'Tutup akun',
75         'Semua data anda akan dihapus, Anda yakin akan menutup akun?',
76         [
77             { text: 'batal', onPress: () => console.log('Cancel Pressed'), style: 'cancel' },
78             { text: 'OK', onPress: () => handleDeleteUser() },
79         ],
80         { cancelable: false },
81     );
82 };
83
84 return (
85     <ScrollView style={{ flex: 1, marginTop: 25 }}>
86         <View style={{ alignItems: 'center' }}>
87             <View style={styles.wrapLogo}>
88                 {isLoading ? <Spinner /> : (
89                     <Image style={styles.tinyLogo} source={{ uri: imgUrl !== null ? imgUrl : 'https://www.w3schools.com/howto/img_avatar.png' }} />
90                 )}
91             </View>
92             <TouchableOpacity style={{ padding: 20 }} onPress={selectImage}>
93                 <Text style={{ color: theme.colors.primary, ...theme.textVariants.body, fontSize: 15 }}> Change image</Text>
94             </TouchableOpacity>
95         </View>
96         <View style={styles.infoProfile}>

```

```

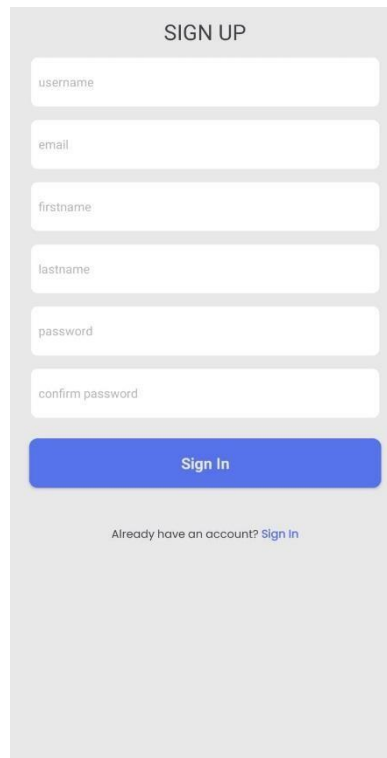
97 |         <View style={{flexDirection: 'row'}}>
98 |             <Text style={styles.fontSecondary}>Info Profile</Text>
99 |             <Icon name="information-circle-outline" size={20} color={theme.colors.fontColorDark} />
100 |         </View>
101 |         <View style={styles.infoProfileContent}>
102 |             <Text style={styles.fontList}>Username </Text>
103 |             <TouchableOpacity style={styles.btnChangeUser} disabled={true}> <Text style={{ ...theme.textVariants.body, color: 'gray',
104 |                 fontSize: 15,}}>{authContext.userState.username}</Text>
105 |             </TouchableOpacity>
106 |         </View>
107 |         <View style={styles.infoProfileContent}>
108 |             <Text style={styles.fontList}>Email </Text>
109 |             <TouchableOpacity style={styles.btnChangeUser} disabled={true}> <Text style={{ ...theme.textVariants.body, color: 'gray',
110 |                 fontSize: 15 }}> {authContext.userState.email} </Text>
111 |             </TouchableOpacity>
112 |         </View>
113 |         <View style={styles.infoProfileContent}>
114 |             <Text style={styles.fontList}>Firstname </Text>
115 |             <TouchableOpacity style={styles.btnChangeUser} onPress={() => handleUbahProfile('Firstname', authContext.userState.firstname)}>
116 |                 <Text style={{ ...theme.textVariants.body, color: theme.colors.fontColorDark, fontSize: 15, }}>
117 |                     {authContext.userState.firstname}
118 |                 </Text>
119 |                 <View style={{ flex: 1, justifyContent: 'center', alignItems: 'flex-end' }}>
120 |                     <Icon name="chevron-forward-outline" size={15} color={theme.colors.fontColorDark} />
121 |                 </View>
122 |             </TouchableOpacity>
123 |         </View>
124 |         <View style={styles.infoProfileContent}>
125 |             <Text style={styles.fontList}>Lastname </Text>
126 |             <TouchableOpacity
127 |                 style={styles.btnChangeUser}
128 |                 onPress={() => handleUbahProfile('Lastname', authContext.userState.lastname)}>
129 |                 <Text style={{ ...theme.textVariants.body, color: theme.colors.fontColorDark, fontSize: 15, }}> {authContext.userState.lastname}
130 |                 </Text>
131 |                 <View style={{ flex: 1, justifyContent: 'center', alignItems: 'flex-end' }}>
132 |                     <Icon name="chevron-forward-outline" size={15} color={theme.colors.fontColorDark} />
133 |                 </View>
134 |             </TouchableOpacity>
135 |         </View>
136 |     </View>
137 |
138 |     <TouchableOpacity style={{ flexDirection: 'row', paddingHorizontal: 10, width: 100, marginBottom: 10}} onPress={() => logOut()}>
139 |         <View style={{ justifyContent: 'center', marginRight: 5 }}>
140 |             <Icon name="log-out-outline" size={25} color={theme.colors.failure} />
141 |         </View>
142 |         <Text style={{ color: 'red', paddingVertical: 10, }}> Log out </Text>
143 |     </TouchableOpacity>
144 |     <TouchableOpacity
145 |         style={{ flexDirection: 'row', paddingHorizontal: 10, width: 110, }}
146 |         onPress={showConfirmDialog}>
147 |         <View style={{ justifyContent: 'center', marginRight: 5 }}>
148 |             <Icon name="close-circle-outline" size={25} color={theme.colors.failure} />
149 |         </View>
150 |         <Text style={{ color: 'red', paddingVertical: 10, }}> Tutup Akun </Text>
151 |     </TouchableOpacity>
152 |     <TouchableOpacity style={{ marginTop: 20, marginLeft: 15, }} onPress={() => Linking.openURL('mailto:hielmisulaeman10@gmail.com?subject=Bug Report MoneyBook')}>
153 |         <Text style={{ color: theme.colors.primary, }}>Report a bug</Text>
154 |     </TouchableOpacity>
155 | </ScrollView>
156 | );
157 | }
158 |
159 | const styles = StyleSheet.create({
160 |     tinyLogo: { width: '100%', height: '100%', shadowColor: 'gray', borderRadius: 50 },
161 |     wrapLogo: { width: 100, height: 100, borderRadius: 100, backgroundColor: 'gray' },
162 |     infoProfile: { borderWidth: 0.3, borderColor: 'gray', padding: 10, marginBottom: 10 },
163 |     fontSecondary: { color: theme.colors.fontColorDark, ...theme.textVariants.body, fontWeight: 'bold', fontSize: 17, marginRight: 5 },
164 |     infoProfileContent: { flex: 1, flexDirection: 'row', padding: 10, margin: 10, },
165 |     fontList: { ...theme.textVariants.body, color: 'gray', fontSize: 15, flex: 1, paddingVertical: 5 },
166 |     btnChangeUser: { flex: 2, paddingVertical: 5, flexDirection: 'row', },
167 | });

```

2. PENGGUNAAN PROGRAM

BudgetGenius adalah sebuah aplikasi mobile berbasis Android yang dibangun dengan framework React Native dengan menggunakan sistem REST API dan menggunakan database mysql. Aplikasi ini dibuat untuk memberikan kemudahan dalam mencatat pemasukan dan pengeluaran uang serta terdapat fitur budgeting seperti 50/30/20 dan 70/10/10/10.

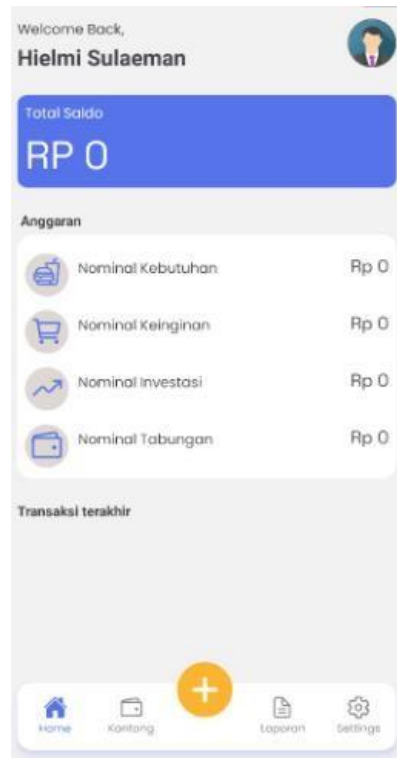
2.1 Langkah Membuat Akun Di BudgetGenius



The image shows a mobile application screen for signing up. At the top, the text 'SIGN UP' is centered. Below it are six white input fields with light gray borders, labeled 'username', 'email', 'firstname', 'lastname', 'password', and 'confirm password'. A blue button with the text 'Sign In' is positioned below the input fields. At the bottom of the screen, there is a link that reads 'Already have an account? Sign In'.

Gambar 1 – Membuat Akun di BudgetGenius

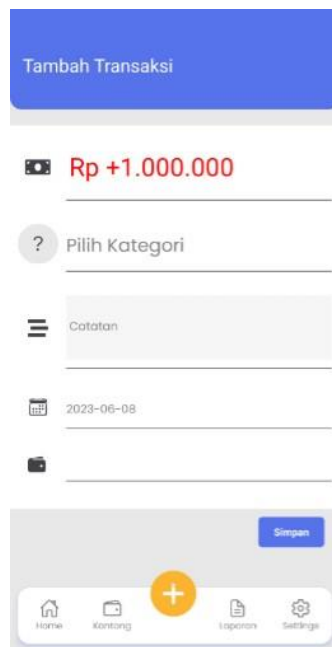
Halaman Sign Up ini digunakan untuk menambahkan akun pada aplikasi. Dimana data User yang diminta meliputi Username, Email, Firstname, Lastname, Password dan confirm password. Setelah data disimpan akun dapat digunakan pada aplikasi BudgetGenius. Halaman ini dapat di akses pada menu login dan pilih Sign Up. Setelah membuat akun setelah itu dapat melakukan transaksi pemasukan ataupun pengeluaran. Jika sudah melakukan Login, akan diarahkan ke halaman Home, seperti gambar .



Gambar 8 - Halaman Home

2.2 Melakukan transaksi pemasukan

Untuk melakukan transaksi, pilih tombol transaksi (tombol tengah).



Gambar 9 – Melakukan Transaksi pemasukan

Halaman ini digunakan untuk melakukan pencatatan pemasukan, dibutuhkan data jumlah uang pemasukannya, kategorinya, catatan, tanggalnya dan budgeting yang digunakan (otomatis oleh sistem).

2.3 Menambah Kategori

Menu menambahkan kategori akan muncul Ketika di halaman transaksi memilih kategori.



Gambar 2 – Menambah Kategori

Halaman ini bertujuan untuk menambahkan kategori dalam transaksi, kategori ini dibuat sesuai dari kelompoknya, kelompok ini terbagi beberapa ada Pemasukan 50/30/20, Pemasukan 70/10/1010 dan pengeluaran ada Tabungan, Kebutuhan, Keinginan, dan Investasi. Kelompok kategori ini disesuaikan dengan kategori yang akan dibuat. Data yang diperlukan pada halaman ini adalah Nama kategori, Icon Kategori dan kelompok kategori.

2.4 Halaman Laporan

Setelah Melakukan transaksi maka transaksi tersebut akan muncul pada halaman laporan, seperti gambar



Gambar 10 – Halaman Laporan

Pada halaman ini menampilkan Riwayat dari pemasukan dan pengeluaran, juga ditampilkan berupa grafik yang akan menunjukkan perbandingan pengeluaran dan pemasukan dalam satu bulan. Pada grafik yang di munculkan juga dapat diatur sesuai bulan dan tahun yang diinginkan.