**Pengujian Aplikasi Manager Berbasis Android**



Nama : Sahda Salsabil

NIM : 2000018089

Kelas : B

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INDUSTRI**

**UNIVERSITAS AHMAD DAHLAN**

1. **Diagram Path**
2. Listing Program

ContainerButtonProgress(

widget: ProgressButtonCustom(

colorIdle: Variables.primaryColor,

colorSuccess: Variables.colorGreen,

idleText: 'Pinjam',

idleIcon: Icon(Icons.send\_outlined, color: Colors.white),

failText: 'Pinjam Success'

failIcon: Icon(Icons.send\_outlined, color: Colors.white),

successText: 'Pinjam Success',

successIcon:

Icon(Icons.send\_outlined, color: Colors.white),

onPressed: () async {

FocusScope.of(context).unfocus();

setState(() => loading = ButtonState.loading);

if (\_formKey.currentState.validate()){

pin = await showPinDialog(

context,

Variables.primaryColor,

Variables.colorGreen,

(val) => {},

pin,

'Enter Your PIN');

pinFromDB =

await DatabaseService(uid: user.uid).getPin();

if (pin = pinFromDB){

tambahUtang = widget.utang.utang + int.parse(utang);

await DatabaseService(uid: user.uid).tambahUtangOrPiutang(

widget.utang.uid,

tambahUtang,

deskripsi,

'utang',

DateTime.now().microsecondsSinceEpoch,

int.parse(utang),

'tambah');

setState(() => loading = ButtonState.success);

await Future.delayed(Duration(seconds: 1));

Navigator.of(context).pushAndRemoveUntil(

MaterialPageRoute(

builder: (context) => BottomNavigation()),

(Route<dynamic> route) => false);

} else {

setState(() => loading = ButtonState.fail);

await Future.delayed(Duration(seconds: 1));

setState(() => loading = ButtonState.fail);

}

} else {

setState(() => loading = ButtonState.fail);

await Future.delayed(Duration(seconds: 1));

setState(() => loading = ButtonState.fail);

}

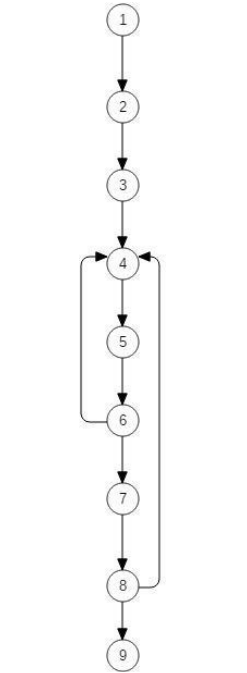
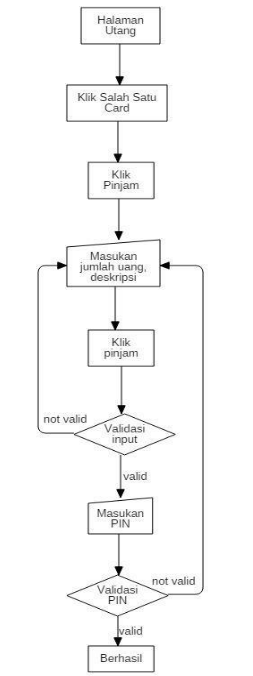
},

stateButton: loading,

),

)

1. Diagram Path



Untuk jalur yang digunakan :

* Jalur path 1 = 1-2-3-4-5-6-7

(1)Berarti halaman awal menampilkan halaman buat utang , (2)kemudian memasukan nama , deskripsi , jutuh tempo, setelah itu(3) klik button pinjam (4)lalu validasi input (valid) , (5)masukan pin(6) setelah itu validasi pin(valid),(7) kemudian berhasil.

* Jalur path 2 = 1-2-3-4-2-3-4-5-6-7

(1)Masuk kehalaman buat utang,(2) lalu masukan nama , deskripsi , jumlah jatuh tempo ,(3) lalu klik button pinjam , (4)validasi input (not valid) ,(2) lalu masukan nama , deskripsi , jumlah jatuh tempo ,(3) lalu klik button pinjam ,(4) validasi input (valid) , (5) masukan pin , (6) validasi pin (valid) ,(7) berhasil.

* Jalur path 3 = 1-2-3-4-5-6-2-3-4-5-6-7

(1)Berarti halaman awal menampilkan halaman buat utang , (2)kemudian memasukan nama , deskripsi , jutuh tempo, setelah itu(3) klik button pinjam (4)lalu validasi input (valid) , (5)masukan pin(6) setelah itu validasi pin (not valid),(2) kemudian memasukan nama , deskripsi , jutuh tempo, setelah itu(3) klik button pinjam (4)lalu validasi input (valid) , (5)masukan pin(6) setelah itu validasi pin (valid), (7) berhasil.

1. **Hasil Perhitungan Kompleksitas Cyclomatic**

V(G) = (E-N)+2

V(G) = jumlah region

E = jumlah edge yang ditentukan dengan gambar panah

N = jumlah simpul grafik (node) dengan gambar lingkaran

V(G) = (8-7)+2

V(G) = 3

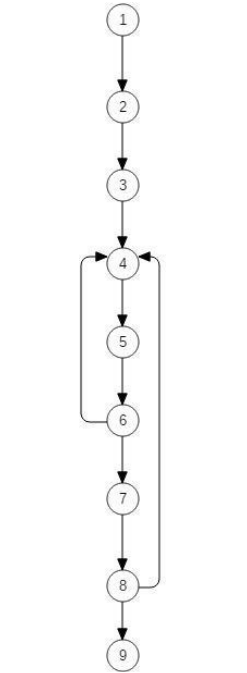
1. **Tabel Jalur Bebas Sejumlah Kompleksitas Cyclomatic**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 |  | 1 |  |  |  |  |  |
| 2 |  |  | 1 |  |  |  |  |
| 3 |  |  |  | 1 |  |  |  |
| 4 |  |  |  |  | 1 |  |  |
| 5 |  |  |  |  |  | 1 |  |
| 6 |  |  |  |  |  |  | 1 |
| 7 |  |  |  |  |  | 1 |  |

|  |  |
| --- | --- |
| Path | 1 |
| Jalur | 1-2-3-4-5-6-7 |
| Skenario | 1. Halaman buat utang  2. Masukkan nama,deskripsi, jumlah uang, jatuh tempo.  3. Klik button pinjam  4. Validasi input (valid)  5. Masukan PIN  6. Validasi PIN (valid)  7. Berhasil |
| Hasil Pengujian | Berhasil |
| Path | 2 |
| Jalur | 1-2-3-4-2-3-4-5-6-7 |
| Skenario | 1. Halaman buat utang  2. Masukkan nama,deskripsi,jumlah uang, jatuh tempo.  3. Klik button pinjam  4. Validasi input (not valid)  2. Masukkan nama,deskripsi,jumlah uang, jatuh tempo.  3. Klik button pinjam  4. Validasi input (valid)  5. Masukan PIN  6. Validasi PIN (valid)  7. Berhasil |
| Hasil Pengujian | Berhasil |
| Path | 3 |
| Jalur | 1-2-3-4-5-6-2-3-4-5-6-7 |
| Scenario | 1. Halaman buat utang  2. Masukkan nama,deskripsi, jumlah uang, jatuh tempo.  3. Klik button pinjam  4. Validasi input (valid)  5. Masukan PIN  6. Validasi PIN (not valid)  2. Masukkan nama,deskripsi, jumlah uang, jatuh tempo.  3. Klik button pinjam  4. Validasi input (valid)  5. Masukan PIN  6. Validasi PIN (valid) |

1. **Skenario uji dengan menggunakan Teknik pengujian BVA**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Kegiatan** | **Hasil yang diharapkan** | **Hasil** |
| 1 | User memilih halaman buat utang dan memasukan nama, deskripsi,jumlah uang,jatuh tempo, kemudian user memasukan PIN yang valid | Berhasil menambahkan utang dan data berhasil ditambahkan ke database | Sistem menampilkan pesan sukses menambahkan utang dan Kembali ke halaman utama |
| 2 | User memilih halaman buat utang dan tidak memasukan semua kolom input lalu user mengklik tombol pinjam. Lalu user mamasukan Kembali data input,setelah itu user memasukkan PIN yang valid. | Sebelum data berhasil ditambahkan ke database,user mendapatkan pesan error karena tidak mengisi semua kolom input. Kemudian setelah berhasil input dan memasukkan PIN maka data berhasil ditambahkan ke database | Sistem menampilkan pesan error terlebih dahulu karena semua kolom input tidak diisi,baru menampilkan pesan sukses setelah berhasil semua validasi. |
| 3 | User memilih halaman buat utang dan memasukkan semua data input lalu user menglik button pinjam namun salah memasukkan PIN setelah itu user mengulangi mengklik tombol pinjam dan memasukkan PIN yang benar | Muncul pesan error Ketika salah memasukkan PIN, dan berhasil Ketika user sudah memasukkan PIN yang benar | Sistem menampilkan pesan error Ketika user salah memasukkan PIN dan berhasil memasukkan data ke database Ketika user sudah berhasill memasujjan PIN yang benar. |

1. **Hasil Uji White Box**

V(G) = (E-N)+2

V(G) = jumlah region

E = jumlah edge yang ditentukan dengan gambar panah

N = jumlah simpul grafik (node) dengan gambar lingkaran

V(G) = (8-7)+2

V(G) = 3

Jadi jalur path yang didapat ada 3 yaitu :

Jalur 1 = 1-2-3-4-5-6—7

Jalur 2 = 1-2-3-4-2-3-4-5-6-7

Jalur 3 = 1-2-3-4-5-6-2-3-4-5-6-7

|  |  |
| --- | --- |
| Path | 1 |
| Jalur | 1-2-3-4-5-6-7 |
| Skenario | 1. Halaman buat utang  2. Masukkan nama,deskripsi, jumlah uang, jatuh tempo.  3. Klik button pinjam  4. Validasi input (valid)  5. Masukan PIN  6. Validasi PIN (valid)  7. Berhasil |
| Hasil Pengujian | Berhasil |
| Path | 2 |
| Jalur | 1-2-3-4-2-3-4-5-6-7 |
| Skenario | 1. Halaman buat utang  2. Masukkan nama,deskripsi,jumlah uang, jatuh tempo.  3. Klik button pinjam  4. Validasi input (not valid)  2. Masukkan nama,deskripsi,jumlah uang, jatuh tempo.  3. Klik button pinjam  4. Validasi input (valid)  5. Masukan PIN  6. Validasi PIN (valid)  7. Berhasil |
| Hasil Pengujian | Berhasil |
| Path | 3 |
| Jalur | 1-2-3-4-5-6-2-3-4-5-6-7 |
| Scenario | 1. Halaman buat utang  2. Masukkan nama,deskripsi, jumlah uang, jatuh tempo.  3. Klik button pinjam  4. Validasi input (valid)  5. Masukan PIN  6. Validasi PIN (not valid)  2. Masukkan nama,deskripsi, jumlah uang, jatuh tempo.  3. Klik button pinjam  4. Validasi input (valid)  5. Masukan PIN  6. Validasi PIN (valid) |

1. **Hasil Uji Black Box**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Kegiatan** | **Hasil yang diharapkan** | **Hasil** | **Keterangan** |
| 1 | User memilih halaman buat utang dan memasukan nama, deskripsi,jumlah uang,jatuh tempo, kemudian user memasukan PIN yang valid | Berhasil menambahkan utang dan data berhasil ditambahkan ke database | Sistem menampilkan pesan sukses menambahkan utang dan Kembali ke halaman utama | Valid |
| 2 | User memilih halaman buat utang dan tidak memasukan semua kolom input lalu user mengklik tombol pinjam. Lalu user mamasukan Kembali data input,setelah itu user memasukkan PIN yang valid. | Sebelum data berhasil ditambahkan ke database,user mendapatkan pesan error karena tidak mengisi semua kolom input. Kemudian setelah berhasil input dan memasukkan PIN maka data berhasil ditambahkan ke database | Sistem menampilkan pesan error terlebih dahulu karena semua kolom input tidak diisi,baru menampilkan pesan sukses setelah berhasil semua validasi. | Valid |
| 3 | User memilih halaman buat utang dan memasukkan semua data input lalu user menglik button pinjam namun salah memasukkan PIN setelah itu user mengulangi mengklik tombol pinjam dan memasukkan PIN yang benar | Muncul pesan error Ketika salah memasukkan PIN, dan berhasil Ketika user sudah memasukkan PIN yang benar | Sistem menampilkan pesan error Ketika user salah memasukkan PIN dan berhasil memasukkan data ke database Ketika user sudah berhasill memasujjan PIN yang benar. | Valid |

1. **Kesimpulan**

Dapat disimpulkan tujuan aplikasi Debt Manager dilakukan pengujian White-box adalah untuk menguji suatu aplikasi atau software dengan cara melihat modul untuk dapat meneliti dan menganalisa kode dari program yang dibuat ada yang salah atau tidak jadi dengan memeriksa komponen perangkat lunak apakah berjalan semestinya dengan melihat internal kode (source code) dari perangkat lunak tersebut.Berdasarkan hasil pengujian “Debt Manager” dapat disimpulkan bahwa aplikasi ini telah lolos tanpa adanya error untuk fungsionalitas utamanya, hal itu terlihat pada pengujian bagian buat utang , bayar utang , dan tambah pinjaman semuanya dinyatakan valid dengan kompleksitas yang rendah karena hanya memiliki 2-3 jalur yang ada pada setiap fungsi.

1. **Review**

Berdasarkan hasil pengujian yang dilakukan dengan menggunakan white box dan black box. Sistem “Debt Manager” menerapkan metode white-box testing agar hasil pengujiain yang telah didapatkan akan dilakukan evaluasi Kembali,sehingga mampu menyimpulkan sebuah perangkat lunak dapat berjalan sesuai dengan kebutuhan yang diharapkan. Pada pengujian path fungsi buat utang setelah diketahui jalur independennya , dilakukan dengan perbandingan dengan menggunakan hubungan antara cyclomatic complexity dan resiko. Dan didapatkan fungsi buat utang ini memiliki resiko yang rendah dengan tingkat prosedur yang sederhana karena memiliki jalur independent berjumlah 3. Pada table test case bayar utang dibuat berdasarkan jalur independent yang telah didapatkan hasil valid untuk ketiga pengujian tidak meemukan error.

1. **Rekomendasi**

Pada aplikasi Debt Manager di rekomendasikan bagi pengguna yang membutuhkan pengelolaan, yang dapat mudah digunakan untuk mengelola hutang Anda dan hutang orang lain (kredit Anda).