

## **“Core Banking Testing Pada Fitur Customer Transaction di BPR Lestari Bali”**

*Putu Arya Novianingsih<sup>1</sup>, Nengah Widya Utami<sup>2</sup>, I Gusti Agung Pramesti Dwi Putri<sup>3</sup>*

*<sup>1,2,3</sup>Sistem Informasi Akuntansi, Universitas Primakara, Denpasar*

*Email: [aryanovia2811@gmail.com](mailto:aryanovia2811@gmail.com), [widya@primakara.ac.id](mailto:widya@primakara.ac.id), [pramesti@primakara.ac.id](mailto:pramesti@primakara.ac.id).*

### **ABSTRACT**

BPR Lestari, salah satu Bank Perkreditan Rakyat (BPR) di Bali tengah merumuskan strategi untuk meningkatkan kualitas layanan dan efisiensi aktivitas perbankan melalui implementasi sistem terintegrasi, yaitu *Core Banking System (CBS)*. CBS merupakan platform yang dirancang untuk mengelola berbagai operasi inti perbankan, seperti penyimpanan data nasabah, transaksi harian, dan layanan digital. Penelitian ini dilakukan dengan tujuan untuk menguji CBS yang telah dikembangkan menggunakan metode *BlackBox Testing* dengan teknik *Equivalence Partitioning*. Pengujian dilakukan fokus pada fitur *Customer Transaction*, yang terdiri dari enam modul utama, yaitu: *Transaction, Deposit, Customer, Collateral, Account & Wallet*, serta *Lending*. Hasil pengujian menunjukkan tingkat keberhasilan mencapai 98,38%. Dari pengujian ini, dapat disimpulkan bahwa metode *BlackBox Testing* dengan teknik *Equivalence Partitioning* efektif dalam memastikan fungsionalitas CBS, mengurangi risiko gangguan layanan, dan meningkatkan kepercayaan pengguna terhadap sistem.

**Kata Kunci:** BPR Lestari, *Core Banking System, Equivalence Partitioning, BlackBox Testing, Pengujian Perangkat Lunak, Modul Customer Transaction.*

### **INTRODUCTION**

Sektor keuangan, terutama sektor perbankan dapat berdampak pada pertumbuhan ekonomi dengan mendorong investasi dan meningkatkan produktivitas modal [1]. Menurut data Otoritas Jasa Keuangan (OJK) per Juli 2024, aset total perbankan di Indonesia mencapai Rp12.012,4 triliun, dimana sebagian besar dari aset tersebut dikelola oleh Bank Umum, sementara sisanya dikelola oleh Bank Perkreditan Rakyat (BPR). Kedua jenis bank ini memiliki fungsi yang berbeda. Sesuai dengan fungsinya, bank umum dapat menjalankan kegiatan usaha secara konvensional dan berperan sebagai perantara dalam lalu lintas pembayaran. Di sisi lain, BPR tidak memiliki kewenangan untuk menjalankan fungsi lalu lintas pembayaran. BPR juga memiliki pasar tersendiri yang mencakup segmen seperti Usaha Mikro, Kecil, dan Menengah (UMKM) serta lainnya [2].

Salah satu BPR yang menjalankan fungsi tersebut adalah BPR Lestari. Saat ini, BPR Lestari tengah merumuskan strategi untuk meningkatkan kualitas layanan dan aktivitas perbankan melalui implementasi sistem yang terintegrasi, yaitu dengan memanfaatkan *Core Banking System (CBS)*.

*Core Banking System (CBS)* merupakan *platform* terintegrasi yang mengelola seluruh operasi inti perbankan, mulai dari penyimpanan data nasabah, transaksi harian, hingga layanan digital seperti *mobile banking* dan *corporate banking*. Sebelum *Core Banking System (CBS)* diimplementasikan dan digunakan oleh *user*, penting dilakukan pengujian menyeluruh untuk memastikan sistem berjalan dengan aman dan efisien.

Pengujian disini maksudnya adalah pengujian perangkat lunak atau pengujian terhadap *Core Banking System (CBS)* tersebut. Pengujian dilakukan untuk mengidentifikasi kesalahan, memvalidasi fungsionalitas, dan

mengevaluasi kinerja perangkat lunak sebelum diserahkan kepada pengguna akhir [3]. Walaupun pengujian tidak sepenuhnya menjamin kualitas, proses ini dapat meningkatkan tingkat kepercayaan pengguna terhadap perangkat lunak [4].

Terdapat beberapa metode pengujian, diantaranya *BlackBox Testing*, *whitebox testing*, dan *greybox testing* [4]. Metode yang digunakan dalam pengujian ini adalah metode *black-box testing*. *Black-box testing* sendiri adalah metode pengujian berdasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi dan kesesuaian alur fungsi dengan sistem kerja yang diinginkan perancangannya [5]. Metode ini dipilih karena sesuai dengan kondisi pengujian yaitu: 1). Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu, 2). Pengujian dilakukan dari sudut pandang pengguna, ini membantu mengungkapkan ambiguitas dan inkonsistensi dalam spesifikasi persyaratan, 3). *Programmer* dan *tester* keduanya saling bergantung[6].

Pengujian dengan metode *BlackBox Testing* memiliki dua teknik utama yaitu *Equivalence Partitioning* dan *Boundary Value Analysis*. Dalam pengujian ini, teknik yang digunakan adalah *Equivalence Partitioning*. Penggunaan metode ini bertujuan untuk mengidentifikasi potensi *bug* dan memastikan setiap fungsi sistem bekerja sesuai dengan yang diharapkan, sehingga risiko gangguan layanan bisa diminimalkan.

Pengujian dengan metode *Blackbox Equivalence Partitioning* ini dilakukan terbatas pada fitur-fitur *Customer Transaction*. Fitur *Customer Transaction* meliputi *Modul Transaction*, *Modul Deposit*, *Modul Customer*, *Modul Collateral*, *Modul Account* dan *Wallet*, *Modul Lending*.

## **METHODS**

Metode yang digunakan dalam pengujian adalah metode *BlackBox Testing* dengan Teknik *Equivalence Partitioning*. Berikut adalah tahapannya.



*Gambar 1 Alur Penelitian*

### **1. Menentukan Test Case**

Tahap ini dimulai dengan identifikasi fungsi-fungsi yang akan diuji dalam masing-masing modul. Test case difokuskan pada pengujian terhadap masukan (*input*) dan keluaran (*output*) yang diharapkan dari sistem.

- Modul *Transaction*: Menguji validasi transaksi seperti transfer, pembayaran, dan mutasi rekening.
- Modul *Deposit*: Menguji fungsi setoran tabungan dan penarikan.
- Modul *Customer*: Menguji pendaftaran, pembaruan data nasabah, dan penghapusan nasabah.
- Modul *Collateral*: Menguji pengelolaan jaminan, seperti pendaftaran, pembaruan, dan pelaporan data jaminan.
- Modul *Account & Wallet*: Menguji fungsi pembukaan rekening baru, manajemen saldo, dan pengelolaan *e-wallet*.
- Modul *Lending*: Menguji pengajuan, pencairan, dan pelunasan pinjaman.

### **2. Menetapkan Kriteria**

Kriteria pengujian ditentukan untuk mengukur keberhasilan atau kegagalan setiap test case.

- Kriteria Keberhasilan: Sistem harus memberikan hasil yang sesuai dengan spesifikasi, seperti respons yang tepat, validasi yang benar, atau keluaran yang sesuai.
- Kriteria Kegagalan: Sistem memberikan hasil yang tidak sesuai, seperti pesan kesalahan yang salah, validasi yang gagal, atau hasil yang tidak diinginkan.

### **3. Mendefinisikan Partisi**

Teknik *Equivalence Partitioning* digunakan untuk membagi masukan menjadi beberapa partisi atau kelas yang saling eksklusif. Setiap partisi mewakili sekumpulan nilai masukan yang diharapkan menghasilkan perilaku serupa.

- *Valid Partition*: Nilai masukan yang sesuai dengan spesifikasi, seperti nomor rekening yang valid atau jumlah transfer dalam batas maksimum.
- *Invalid Partition*: Nilai masukan yang tidak sesuai, seperti jumlah transfer di bawah nol atau nomor rekening yang tidak terdaftar.

#### 4. Menyusun Data Uji

Setelah partisi didefinisikan, data uji dirancang untuk mencakup semua skenario yang relevan dari partisi valid dan invalid.

#### 5. Membuat Kasus Uji

Berdasarkan data uji, kasus uji disusun secara terperinci. Setiap kasus uji mencakup:

- Identifikasi kasus uji: Modul yang diuji, ID skenario
- Skenario pengujian: Langkah sistematis yang harus dilakukan.
- Hasil yang diharapkan: Output atau respons yang diharapkan dari sistem.
- Status pengujian

#### 6. Melakukan Pengujian dan Evaluasi

Setelah semua kasus uji disiapkan, pengujian dilakukan pada sistem:

- Eksekusi Kasus Uji: Setiap *test case* dijalankan sesuai langkah yang ditentukan.
- Pencatatan Hasil: Hasil aktual dibandingkan dengan hasil yang diharapkan.
- Evaluasi: Jika terdapat perbedaan, *bug* atau masalah dicatat untuk diperbaiki oleh *developer*.
- *Retesting*: Setelah perbaikan, skenario yang gagal diuji ulang untuk memastikan masalah telah terselesaikan.

## RESULTS

### A. Hasil Pengujian

Setelah dilaksanakan pengujian sesuai dengan *testcase*, didapatkan hasil sebagai berikut:

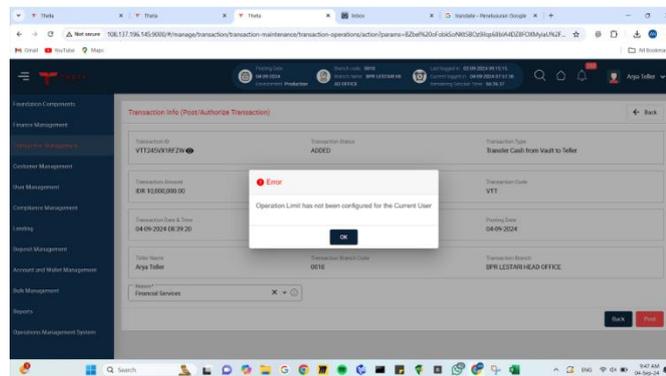
#### 1. Modul *Transaction*

Pada modul ini, terdapat 10 skenario pengujian. Setelah dilakukan pengujian, 9 dari 10 skenario berhasil dijalankan sesuai dengan hasil yang diharapkan, sedangkan 1 skenario tidak berhasil (gagal). Berikut adalah detail hasil pengujian yang telah dilakukan:

*Tabel 1 Modul Transaction*

ID Uji	Fitur	Skenario	Hasil yang diharapkan	Status
A-01	<i>Transaction</i>	Melakukan transaksi <i>Vault</i> ke <i>Teller</i>	Transaksi dari <i>Vault</i> ke <i>Teller</i> berhasil	Gagal
A-02	<i>Transaction</i>	Melihat pertanyaan transaksi	Pengguna dapat melihat transaksi yang sudah	Sukses

ID Uji	Fitur	Skenario	Hasil yang diharapkan	Status
			diposting	
A-03	<i>Transaction</i>	Cetak transaksi	Bukti Transaksi berhasil dicetak	Sukses
A-04	<i>Transaction</i>	Melakukan transaksi Setoran Tunai ke rekening tabungan Nasabah	Setoran Tunai ke Rekening Tabungan Nasabah berhasil	Sukses
A-05	<i>Transaction</i>	Melakukan transfer intracabang	Transfer intracabang berhasil	Sukses
A-06	<i>Transaction</i>	Melakukan transaksi <i>Teller to Vault</i>	Transaksi dari <i>Teller</i> ke <i>Vault</i> berhasil	Sukses
A-07	<i>Transaction</i>	Melakukan transaksi Penarikan Tunai ke rekening tabungan Nasabah	Transaksi Penarikan Tunai ke rekening tabungan Nasabah berhasil dilakukan	Sukses
A-08	<i>Transaction</i>	Melakukan transfer <i>Intrabranh</i> Akun Tabungan ke Akun Tabungan	Transfer <i>Intrabranh</i> Akun Tabungan ke Akun Tabungan berhasil	Sukses
A-09	<i>Transaction</i>	Lakukan transfer <i>Intrabranh Saving Account</i> ke <i>Recurring Deposit</i>	Transfer <i>Intrabranh Saving Account</i> ke <i>Recurring Deposit</i> berhasil	Sukses
A-010	<i>Transaction</i>	Melakukan transaksi setor Tunai dari <i>Walkin Customer</i>	Transaksi setor tunai dari <i>Walkin Customer</i> berhasil	Sukses



**Gambar 2 Screenshot Case Gagal (Authorize Transaction)**

Gambar diatas merupakan salah satu contoh *case* gagal yakni saat *Teller* ingin melakukan *post/authorize Transaction* dari *Vault* ke *Teller* muncul *pop up* “*Operation limit has not been configured for the Current User*”, yang berarti terdapat kesalahan yakni antara akun dan limit transaksi yang belum terkonfigurasi.

## 2. Modul *Deposit*

Pada modul ini, terdapat 10 skenario pengujian. Setelah dilakukan pengujian, 10 dari 10 skenario berhasil dijalankan sesuai dengan hasil yang diharapkan. Berikut adalah detail hasil pengujian yang telah dilakukan:

**Tabel 2 Modul Deposit**

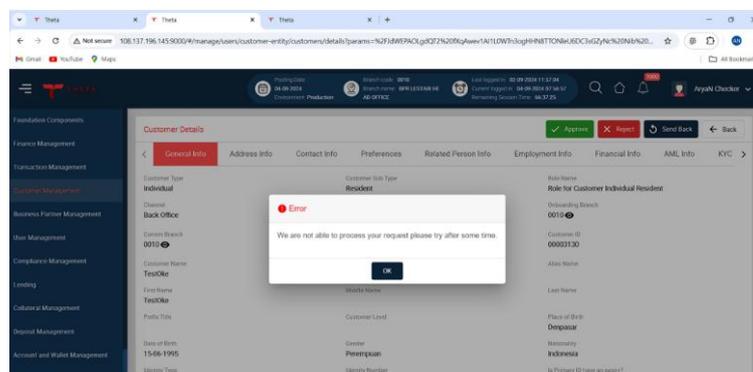
ID Uji	Fitur	Skenario	Hasil yang diharapkan	Status
B-01	<i>Deposit</i>	Membuat Akun Deposit baru	Akun Deposit Baru berhasil dibuat	Sukses
B-02	<i>Deposit</i>	Buat Akun Deposit Baru, dan sumber akunnya dari CIF lain	Akun Deposit baru yang bersumber dari akun CIF lain berhasil dibuat	Sukses
B-03	<i>Deposit</i>	Melihat detail deposit	Detail Deposit muncul	Sukses
B-04	<i>Deposit</i>	Mengubah jangka waktu deposit	Jangka waktu deposit berhasil diubah	Sukses
B-05	<i>Deposit</i>	Mengubah nominal <i>Recurring Deposit</i> lebih besar dari sebelumnya	Nominal <i>Recurring Deposit</i> berhasil diubah	Sukses
B-06	<i>Deposit</i>	Mengubah nominal <i>Recurring Deposit</i> lebih kecil dari sebelumnya	Nominal <i>Recurring Deposit</i> tidak diubah	Sukses
B-07	<i>Deposit</i>	Mengubah kode marketing	Kode Marketing berhasil diubah	Sukses
B-08	<i>Deposit</i>	Mengubah sumber akun untuk penempatan bunga	Sumber akun untuk penempatan bunga bisa diubah	Sukses
B-09	<i>Deposit</i>	Menutup deposit	Penutupan deposit berhasil	Sukses
B-010	<i>Deposit</i>	Menggunakan kalkulator deposit	Perhitungan kalkulator menunjukkan nominal yang sesuai	Sukses

### 3. Modul Customer

Pada modul ini, terdapat 8 skenario pengujian. Setelah dilakukan pengujian, 4 dari 8 skenario berhasil dijalankan sesuai dengan hasil yang diharapkan, sedangkan 4 skenario lainnya tidak berhasil (gagal). Berikut adalah detail hasil pengujian yang telah dilakukan:

**Tabel 3 Modul Customer**

ID Uji	Fitur	Skenario	Hasil yang diharapkan	Status
C-01	<i>Customer</i>	Membuat <i>Customer</i> baru	<i>Customer</i> Baru berhasil dibuat	Gagal
C-02	<i>Customer</i>	Melihat detail <i>Customer</i>	Detail <i>Customer</i> muncul	Sukses
C-03	<i>Customer</i>	Mengedit detail <i>Customer</i>	Detail <i>Customer</i> dapat diedit	Sukses
C-04	<i>Customer</i>	Mengganti alamat <i>Customer</i>	Alamat <i>Customer</i> bisa diganti	Gagal
C-05	<i>Customer</i>	Mengganti email ID <i>Customer</i>	Email ID <i>Customer</i> bisa diganti	Gagal
C-06	<i>Customer</i>	Mengganti status <i>Customer</i>	Status <i>Customer</i> bisa diganti	Sukses
C-07	<i>Customer</i>	Mengganti no telepon <i>Customer</i>	No Telepon <i>Customer</i> bisa diganti	Sukses



**Gambar 3 Screenshot Case Gagal (Approve New Customer)**

ID Uji	Fitur	Skenario	Hasil yang diharapkan	Status
C-08	<i>Customer</i>	Mengganti asal cabang <i>Customer</i>	Asal cabang <i>Customer</i> bisa diganti	Gagal

Gambar di atas menunjukkan salah satu *case* gagal, yaitu saat *role Checker* mencoba melakukan *approval* atas skenario yang diinisiasi oleh *role Maker*. Dalam skenario ini, *role Maker* berhasil membuat akun *customer* baru, namun ketika *role Checker* mencoba melakukan *approval* agar akun tersebut aktif, muncul *pop-up* seperti yang ditampilkan pada gambar. Hal ini menunjukkan bahwa skenario tersebut tidak dapat berjalan sesuai dengan yang diharapkan (gagal).

#### 4. Modul *Collateral*

Pada modul ini, terdapat 7 skenario pengujian. Setelah dilakukan pengujian, 7 dari 7 skenario berhasil dijalankan sesuai dengan hasil yang diharapkan, Berikut adalah detail hasil pengujian yang telah dilakukan:

**Tabel 4 Modul *Collateral***

ID Uji	Fitur	Skenario	Hasil yang diharapkan	Status
D-01	<i>Collateral</i>	Membuat angsuran baru	Angsuran baru berhasil dibuat	Sukses
D-02	<i>Collateral</i>	Melihat detail angsuran	Detail angsuran dapat dilihat	Sukses
D-03	<i>Collateral</i>	Membuat <i>collateral engagement</i>	<i>Collateral engagement</i> berhasil dibuat	Sukses
D-04	<i>Collateral</i>	Melihat Detail <i>Collateral engagement</i>	Detail <i>Collateral engagement</i> bisa dilihat	Sukses
D-05	<i>Collateral</i>	Mengaitkan <i>collateral</i> dengan <i>loan</i>	<i>Collateral</i> dengan <i>Loan</i> berhasil dikaitkan	Sukses
D-06	<i>Collateral</i>	Mengaitkan satu <i>collateral</i> dengan beberapa <i>loan</i>	Satu <i>collateral</i> bisa dikaitkan dengan beberapa <i>loan</i>	Sukses
D-07	<i>Collateral</i>	Mengaitkan beberapa <i>collateral</i> dengan satu <i>loan</i>	Beberapa <i>Collateral</i> bisa dikaitkan dengan satu <i>loan</i>	Sukses

#### 5. Modul *Account dan Wallet*

Pada modul ini, terdapat 16 skenario pengujian. Setelah dilakukan pengujian, 15 dari 16 skenario berhasil dijalankan sesuai dengan hasil yang diharapkan, sedangkan 1 skenario tidak berhasil (gagal). Berikut adalah detail hasil pengujian yang telah dilakukan:

**Tabel 5 Modul *Account dan Wallet***

ID Uji	Fitur	Skenario	Hasil yang diharapkan	Status
E-01	<i>Saving</i>	Membuat akun Tabungan baru	Akun Tabungan berhasil dibuat	Gagal
E-02	<i>Saving</i>	Melihat akun <i>report list</i>	Akun yang telat dibuat muncul di <i>report list</i>	Sukses
E-03	<i>Saving</i>	Mengecek status <i>dormancy</i>	Terdapat <i>dormant</i>	Sukses
E-04	<i>Saving</i>	<i>Update</i> Akun Status	Status Akun berhasil <i>update</i>	Sukses

ID Uji	Fitur	Skenario	Hasil yang diharapkan	Status
E-05	<i>Saving</i>	Mengecek detail laporan transaksi	Suku Bunga Akun yang sudah di <i>update</i> muncul	Sukses
E-06	<i>Saving</i>	Menutup akun Tabungan yang ada	Akun Tabungan berhasil ditutup	Sukses
E-07	<i>Saving</i>	Periksa biaya penutupan	Terdapat biaya penutupan	Sukses
E-08	<i>Saving</i>	Periksa pembayaran bunga yang masih harus dibayar	Terdapat pembayaran bunga yang masih harus dibayar	Sukses
E-09	<i>Saving</i>	Blokir beberapa jumlah untuk Rekening Tabungan tertentu	Pemblokiran berhasil	Sukses
E-010	<i>Saving</i>	Periksa jumlah yang tersedia untuk rekening tabungan	Terdapat jumlah yang tersedia setelah dilakukan pemblokiran	Sukses
E-011	<i>Saving</i>	Periksa total saldo rekening tabungan	Total saldo rekening berubah karena ada pemblokiran	Sukses
E-012	<i>Saving</i>	Mengatur biaya administrasi untuk produk tertentu (Produk akun Tabungan)	Biaya administrasi untuk produk tertentu berhasil diatur	Sukses
E-013	<i>Saving</i>	Pastikan sistem akan secara otomatis mengeksekusi biaya administrasi.	Setiap memproses produk akun tertentu biaya administrasi akan otomatis tereksekusi sesuai dengan inputan	Sukses
E-014	<i>Saving</i>	Ubah kode Pemasaran untuk Rekening Tabungan Tertentu	Kode Pemasaran berhasil dirubah	Sukses
E-015	<i>Saving</i>	Periksa info akun, pastikan kode pemasaran sudah diperbarui	Terdapat informasi bahwa kode pemasaran telah diperbarui	Sukses
E-016	<i>Saving</i>	Periksa data historis untuk akun itu.	Pengguna dapat melihat perubahan kode pemasaran, setelah membandingkan data baru dengan data lama.	Sukses

## 6. Modul Lending

Pada modul ini, terdapat 11 skenario pengujian. Setelah dilakukan pengujian, 11 dari 11 skenario berhasil dijalankan sesuai dengan hasil yang diharapkan, Berikut adalah detail hasil pengujian yang telah dilakukan:

**Tabel 6 Modul Lending**

ID Uji	Fitur	Skenario	Hasil yang diharapkan	Status
F-01	<i>Loan Account</i>	Buat Akun Pinjaman baru	Akun Pinjaman baru berhasil dibuat	Sukses
F-02	<i>Loan Account</i>	Buat akun Pinjaman baru dengan Sertifikat Rumah sebagai jaminan	Akun Pinjaman baru dengan Sertifikat Rumah sebagai jaminan berhasil dibuat	Sukses
F-03	<i>Loan Account</i>	Buat rekening pinjaman baru dengan Surat Keterangan	Rekening pinjaman baru dengan Surat Keterangan Kendaraan	Sukses

ID Uji	Fitur	Skenario	Hasil yang diharapkan	Status
		Kendaraan sebagai jaminannya	sebagai jaminannya berhasil dibuat	
F-04	<i>Loan Account</i>	Buat rekening pinjaman baru dengan <i>Recurring Deposit/Fixed Deposit Advice</i> sebagai jaminan	Rekening pinjaman baru dengan <i>Recurring Deposit/Fixed Deposit Advice</i> sebagai jaminan berhasil dibuat	Sukses
F-05	<i>Loan Account</i>	Buat rekening pinjaman baru, dan pastikan Sumber rekening mencukupi	Skenario berhasil dilakukan	Sukses
F-06	<i>Loan Facility</i>	Membuat Fasilitas Pinjaman untuk CIF tertentu	Fasilitas Pinjaman untuk CIF tertentu berhasil dibuat	Sukses
F-07	<i>Loan Facility</i>	Nasabah menarik Fasilitas Pinjaman	Nasabah bisa menarik Fasilitas Pinjaman	Sukses
F-08	<i>Loan Account</i>	Buat rekening Pinjaman baru untuk Nasabah Bank	Rekening Pinjaman baru untuk Nasabah Bank berhasil dibuat	Sukses
F-09	<i>Loan Account</i>	Buat Rekening Pinjaman, dan rekening sumber dari Cabang yang berbeda	Pembuatan Rekening Pinjaman, dan rekening sumber dari Cabang yang berbeda berhasil	Sukses
F-010	<i>Loan Account</i>	Buat Rekening Pinjaman, dan rekening sumber dari Cabang yang berbeda. Atur akun Sumber menjadi tidak mencukupi.	Skenario tidak bisa diproses	Sukses
F-011	<i>Manual Loan</i>	Melakukan Pelunasan Pinjaman Secara Manual ke Rekening Pinjaman tertentu yang mempunyai tunggakan (Pokok, Bunga, Biaya)	Skenario bisa diproses	Sukses

## DISCUSSION

Setelah dilakukan pengujian dengan total 62 skenario yang diuji pada 6 modul berbeda, diperoleh hasil rekapitan sebagai berikut:

*Tabel 7 Hasil Rekapitan*

No	Modul	Total Skenario	Status		Persentase keberhasilan
			Gagal	Sukses	
1	<i>Transaction</i>	10	0	10	100%
2	<i>Deposit</i>	10	0	10	100%
3	<i>Customer</i>	8	1	7	87,5%
4	<i>Collateral</i>	7	0	7	100%
5	<i>Account &amp; Wallet</i>	16	0	16	100%
6	<i>Lending</i>	11	0	11	100%

Tingkat keberhasilan keseluruhan modul mencapai 100% dengan rata-rata persentase keberhasilan dari seluruh modul adalah 98,38% sehingga bisa dikatakan proses pengujian (*testing*) dinyatakan berhasil. Persentase tersebut didapat dari rumus Teknik *Equivalence Partitioning*, yakni:

$$EP = \frac{\text{Jumlah skenario sukses}}{\text{Total keseluruhan Skenario}} \times 100\%$$

$$EP = \frac{61}{62} \times 100\%$$

$$EP = 98,38\%$$

## CONCLUSION

Tingkat keberhasilan pengujian secara keseluruhan mencapai 98,38%. Hasil ini diperoleh dari pengujian pada enam modul utama, yaitu Modul *Transaction* dengan 10 skenario, Modul *Deposit* dengan 10 skenario, Modul *Collateral* dengan 7 skenario, Modul *Account & Wallet* dengan 16 skenario, dan Modul *Lending* dengan 11 skenario, yang semuanya mencapai tingkat keberhasilan 100%. Sementara itu, Modul *Customer* dengan 8 skenario memiliki tingkat keberhasilan 87,5%, di mana terdapat satu skenario yang gagal akibat *bug* yang memerlukan perbaikan lebih lanjut dari tim *developer*.

## LIMITATION

Penelitian ini memiliki beberapa batasan yang perlu diperhatikan. Pertama, terdapat keterbatasan dalam mengakses sistem *Core Banking System* karena beberapa peran (*role*) memerlukan kredensial khusus yang tidak tersedia selama pengujian. Kedua, pengujian hanya dilakukan menggunakan *data dummy* yang telah disediakan, sehingga terdapat perbedaan antara kondisi pengujian dan implementasi sistem dengan data asli, yang dapat memengaruhi akurasi dan keandalan hasil pengujian. Ketiga, hasil pengujian tidak dapat digeneralisasi, mengingat fitur-fitur pada *Core Banking System* berbeda-beda tergantung pada kebutuhan spesifik masing-masing bank. Terakhir, pengujian hanya menggunakan metode *blackbox testing*, yang memiliki cakupan terbatas, untuk pengujian yang lebih menyeluruh, metode lain seperti *whitebox testing* atau *greybox testing* dapat dipertimbangkan untuk diterapkan.

## REFERENCES

- [1] F. Nabela and H. Thamrin, "Kontribusi Perbankan Syariah terhadap Pertumbuhan Ekonom Negara di Asia Tenggara," *Jurnal Tabarru' : Islamic Banking and Finance*, vol. 5, 2022.
- [2] B. R. Radita Galang Prasetyo, "Strategi Peningkatan Pemasaran Produk Kredit di Perumda BPR Bank Magelang," *Jurnal Nasional Pemasaran & Sumber Daya Manusia*, vol. 3, Oct. 2022.

- [3] D. S. , & P. B. Taley, “Studi Komprehensif tentang Teknik dan Strategi Pengujian Perangkat Lunak: Sebuah Tinjauan,” *Jurnal Internasional Riset & Teknologi Rekayasa (IJERT)*, pp. 817–822, 2020.
- [4] I. R. , M. M. , S. A. A. , & Y. M. A. Dhaifullah, “Survei Teknik Pengujian Perangkat Lunak,” *JACIS: Journal Automation Computer Information System* , vol. 2, no. 1, pp. 31–38, 2022.
- [5] N. W. Rahadi and C. Vikasari, “Pengujian Software Aplikasi Perawatan Barang Milik Negara Menggunakan Metode Black Box Testing Equivalence Partitions,” *Infotekmesin*, vol. 11, no. 1, pp. 57–61, Jan. 2020, doi: 10.35970/infotekmesin.v11i1.124.
- [6] E. Novalia and A. Voutama, “Black Box Testing dengan Teknik Equivalence Partitions Pada Aplikasi Android M-Magazine Mading Sekolah,” 2022.