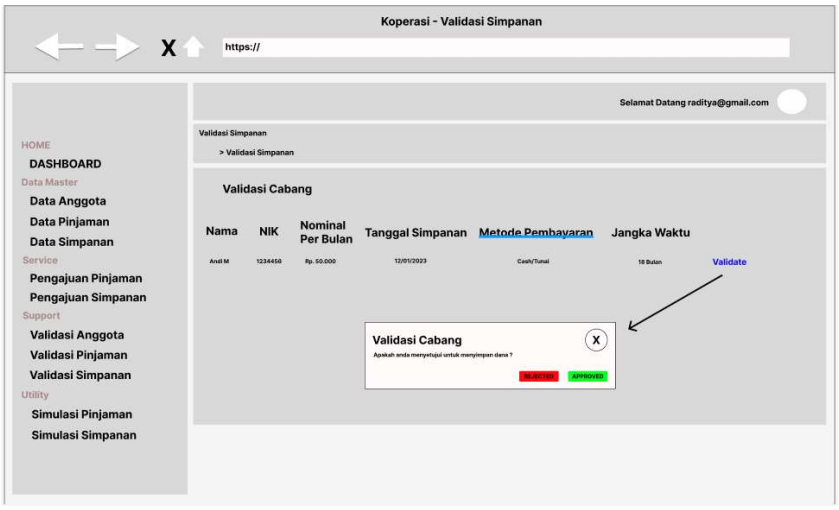
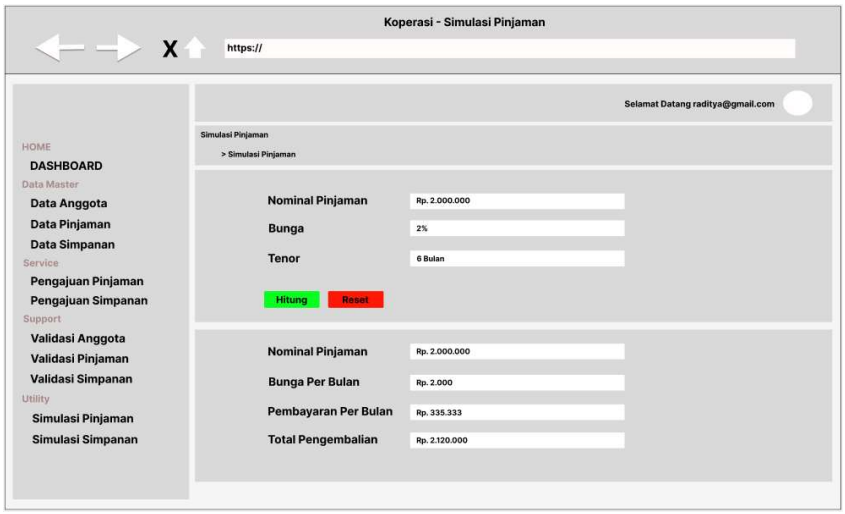
3.2.3.13. Tampilan Validasi Simpanan



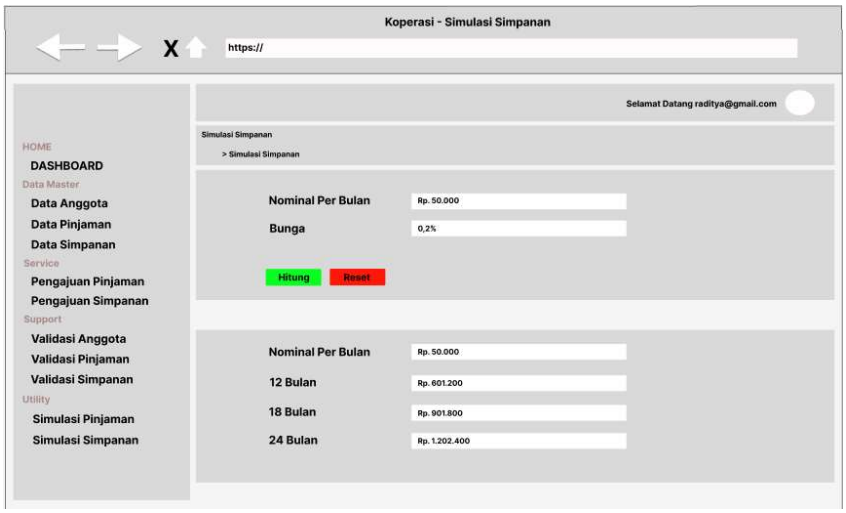
Gambar 3. 18 Tampilan Validasi Simpanan

3.2.3.14. Tampilan Simulasi Pinjaman



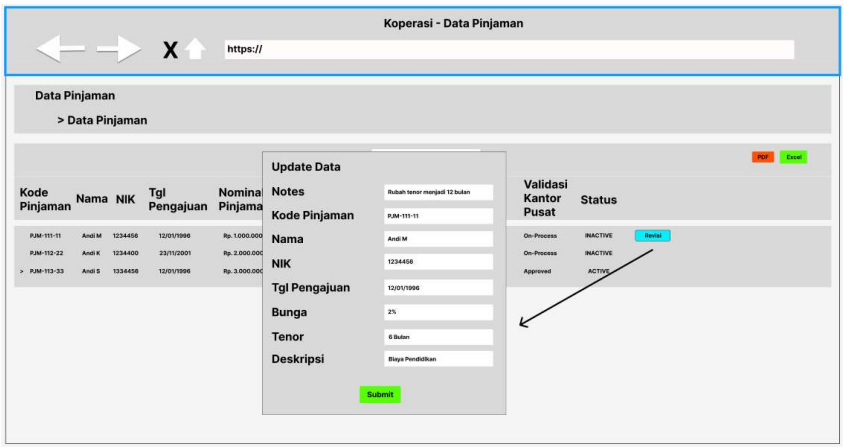
Gambar 3. 19 Tampilan Simulasi Pinjaman

3.2.3.15. Tampilan Simulasi Simpanan



Gambar 3. 20 Tampilan Simulasi Simpanan

3.2.3.16. Tampilan Revisi Data Pinjaman



Gambar 3. 21 Tampilan Revisi Data Pinjaman

BAB IV

IMPLEMENTASI DAN UJICOBA

4.1. Implementasi

Tahap ini merupakan proses mewujudkan rancangan menjadi sistem yang nyata. Hasil akhir dari tahap implementasi adalah sebuah sistem yang telah siap untuk menjalani serangkaian pengujian dan selanjutnya dioperasikan oleh pengguna.

4.1.1. Spesifikasi Produk

Pada bagian spesifikasi produk berisi tentang penjelasan tentang spesifikai produk yang dihasilkan, kebutuhan hardware dan software untuk penerapan sistem.

4.1.1.1. Spesifikasi Perangkat Keras

Tabel 4. 1 Spesifikasi Hardware Laptop / PC

Perangkat Keras	Keterangan
Processor	Dual Core 2.8 Ghz
RAM	DDR3 4GB
Hard disk	200 GB HDD
Monitor	Resolusi 1290x720
Mouse PC	Optic 1 Buah
Keyboard PC	1 Buah

4.1.1.2. Spesifikasi Perangkat Lunak

Tabel 4. 2 Spesifikasi Software Laptop / PC

Perangkat Keras	Keterangan
Windows 10	Sistem Operasi(OS)

4.1.2. Implementasi Database

Hasil implementasi dari desain basis data sesuai dengan perancangan yang telah dibuat sebelumnya, menggunakan aplikasi *phpMyAdmin* untuk mengelola database *MySQL*.

4.1.2.1. Tabel Users

	#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	K
☐	1	id	bigint(20)		UNSIGNED	Tidak	*Tidak ada*	
☐	2	codeUser	varchar(255)	utf8mb4_unicode_ci		Tidak	*Tidak ada*	
☐	3	name	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*	
☐	4	email	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*	
☐	5	password	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*	
☐	6	role	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*	
☐	7	nik	int(11)			Ya	*NULL*	
☐	8	position	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*	
☐	9	employeeStatus	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*	
☐	10	branchName	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*	
☐	11	managerName	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*	
☐	12	joinDate	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*	
☐	13	address	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*	
☐	14	phoneNumber	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*	
☐	15	bankName	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*	
☐	16	accountNumber	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*	
☐	17	validationStatus	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*	
☐	18	registrationStatus	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*	
☐	19	status	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*	

Gambar 4. 1 Tabel users

Berfungsi untuk menyimpan data anggota koperasi yang ada di PT. Rentokil Indonesia Cabang Malang. Data anggota tersebut nantinya akan digunakan untuk otentifikasi login, maupun untuk pinjaman dan simpanan.

4.1.2.2. Tabel loans

1	**id**	bigint(20)		UNSIGNED	Tidak	*Tidak ada*
2	**userId**	bigint(20)		UNSIGNED	Tidak	*Tidak ada*
3	**code**	varchar(255)	utf8mb4_unicode_ci		Tidak	*Tidak ada*
4	**nominal**	double			Ya	*NULL*
5	**interest**	double			Ya	*NULL*
6	**tenor**	int(11)			Ya	*NULL*
7	**date**	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*
8	**description**	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*
9	**loanStatus**	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*
10	**validationLoanStatus**	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*
11	**status**	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*
12	**reason**	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*

Gambar 4. 2 Tabel loans

Berfungsi untuk menyimpan data pengajuan pinjaman yang dilakukan oleh anggota koperasi karyawan di PT. Rentokil Indonesia Cabang Malang.

4.1.2.3. Tabel savings

1	**id**	bigint(20)		UNSIGNED	Tidak	*Tidak ada*
2	**userId**	bigint(20)		UNSIGNED	Tidak	*Tidak ada*
3	**code**	varchar(255)	utf8mb4_unicode_ci		Tidak	*Tidak ada*
4	**nominalPerMonth**	double			Ya	*NULL*
5	**interest**	double			Ya	*NULL*
6	**date**	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*
7	**paymentMethod**	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*
8	**timePeriod**	int(11)			Ya	*NULL*
9	**validationSavingStatus**	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*
10	**status**	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*

Gambar 4. 3 Tabel savings

Berfungsi untuk menyimpan data pengajuan simpanan yang dilakukan oleh anggota koperasi karyawan di PT. Rentokil Indonesia Cabang Malang.

4.1.2.4. Tabel installments

1	**id**	bigint(20)		UNSIGNED	Tidak	*Tidak ada*
2	**userId**	bigint(20)		UNSIGNED	Tidak	*Tidak ada*
3	**loanId**	bigint(20)		UNSIGNED	Tidak	*Tidak ada*
4	**nominalPayment**	double			Ya	*NULL*
5	**date**	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*
6	**paymentMethod**	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*
7	**paymentStatus**	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*
8	**installmentStatus**	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*

Gambar 4. 4 Tabel installments

Berfungsi untuk menyimpan data angsuran yang dimiliki oleh anggota koperasi karyawan di PT. Rentokil Indonesia Cabang Malang.

4.1.2.5. Tabel loantransactions

1	**id**	bigint(20)		UNSIGNED	Tidak	*Tidak ada*	
2	**installmentId**	bigint(20)		UNSIGNED	Tidak	*Tidak ada*	
3	**fileName**	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*	
4	**receipt**	longblob			Ya	*NULL*	

Gambar 4. 5 loantransactions

Berfungsi untuk menyimpan data bukti pembayaran angsuran yang sudah dibayarkan via payroll/gaji karyawan.

4.1.2.6. Tabel savingpayments

1	**id**	bigint(20)		UNSIGNED	Tidak	*Tidak ada*	
2	**userId**	bigint(20)		UNSIGNED	Tidak	*Tidak ada*	
3	**saveId**	bigint(20)		UNSIGNED	Tidak	*Tidak ada*	
4	**nominal**	double			Ya	*NULL*	
5	**payment**	double			Ya	*NULL*	
6	**paymentMethod**	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*	
7	**date**	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*	
8	**status**	varchar(255)	utf8mb4_unicode_ci		Ya	*NULL*	

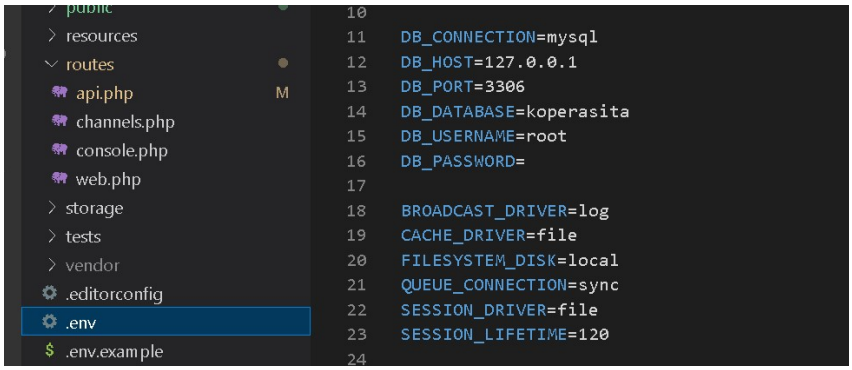
Gambar 4. 6 savingpayments

Berfungsi untuk menyimpan data pembayaran simpanan yang diinputkan oleh admin koperasi, setelah anggota koperasi melakukan pembayaran melalui transfer/tunai ke admin koperasi di cabang malang.

4.1.3. Implementasi Program

Dalam membuat sistem berbasis website penulis memilih menggunkan Microsoft Visual Studio Code dalam membangun sistem berbasis website karena pada Microsoft Visual Studio Code didukung oleh banyak plugin yang memudahkan dalam menulis kode.

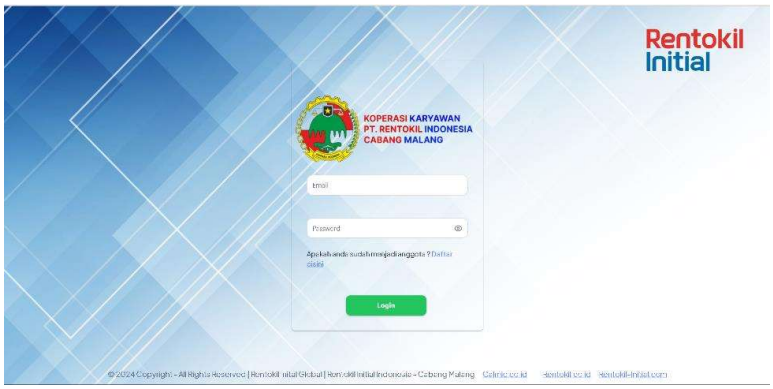
4.1.3.1. Koneksi Database



Segmen Program 4. 1 Koneksi Database

Merubah nama yang tercantum pada baris DB_DATABASE pada file '.env' sesuai dengan nama basis data yang telah direncanakan sebelumnya.

4.1.3.2. Halaman Login



Gambar 4. 7 Halaman Login

Halaman ini menampilkan antarmuka awal aplikasi. Pengguna akan melihat form login yang mengharuskan mereka memasukkan email dan password yang telah terdaftar. Selain itu, tersedia juga tautan untuk mendaftar sebagai anggota baru bagi pengguna yang belum memiliki akun.

```php
function userLogin(Request $request)
{
    $credentials = $request->validate([
        'email' => 'required|email',
        'password' => 'required',
    ]);

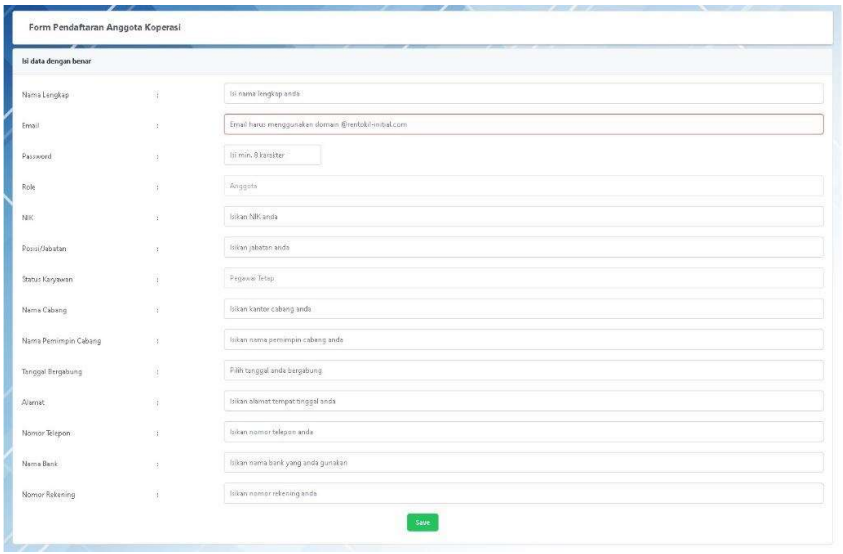
    $user = User::where('email', $request->email)->first();
    if (!$user) {
        return response(["message" => "Maaf, Email tidak terdaftar"], 400);
    }
    if ($user->registrationStatus === 'Rejected' || $user->validationStatus === 'Invalid') {
        return response(["message" => "Maaf, Pendaftaran Anda ditolak"], 400);
    }
    if($user->status === 'INACTIVE'){
        return response(["message" => "Maaf, Pendaftaran sedang dalam proses validasi"], 400);
    }
    if (Auth::attempt($credentials)) {
        $token = $user->createToken('auth_token')->plainTextToken;
        return response()->json([
            'user' => $user,
            'access_token' => $token,
            'token_type' => 'Bearer',
        ]);
    }
    return response()->json(['error' => 'Unauthorized'], 401);
}
```

Segmen Program 4. 2 Kode controller login

Terdapat beberapa proses untuk menangani autentikasi pengguna saat mereka mencoba untuk login pada sisi server. Pertama, memvalidasi kredensial yang diterima dari permintaan, mengonfirmasi apakah email dan password sesuai dengan aturan yang ditetapkan. jika tidak, server merespons dengan pesan kesalahan. Selanjutnya, mencari pengguna dalam basis data berdasarkan alamat email yang diberikan, memberikan tanggapan bahwa email tidak terdaftar jika tidak menemukan pengguna. Kemudian, kode memeriksa jika registrationStatus adalah Rejected dan validationStatus adalah Invalid

maka muncul pesan kesalahan “Maaf, Pendaftaran Anda Ditolak” . Jika kredensial valid dan status akun sesuai, fungsi mencoba mengautentikasi pengguna menggunakan Auth::attempt(), menghasilkan token akses jika berhasil. Namun, jika autentikasi gagal, server memberikan respon 401 (Unauthorized) bersama dengan pesan kesalahan yang sesuai.

4.1.3.3. Halaman Menu Pendaftaran



Gambar 4. 8 Halaman Pendaftaran

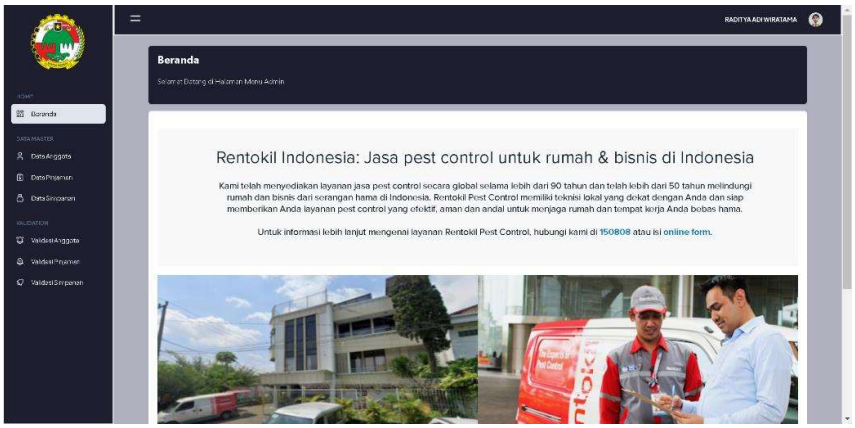
Jika pengguna belum memiliki akun untuk mengakses ke dalam aplikasi, maka pengguna harus mendaftar terlebih dahulu. Terdapat formulir pendaftaran yang harus diisi oleh pengguna.

```php
function userAdd(Request $request)
{
    $existingUser = DB::table('users')->where('nik', $request->input('nik'))->first();
    if ($existingUser) {
        return response(["message" => "User sudah terdaftar"], 400);
    }
    $user = new User;
    $user->name = $request->input('name');
    $user->email = $request->input('email');
    $user->password = $request->input('password');
    $user->role = $request->input('role');
    $user->nik = $request->input('nik');
    $user->codeUser = 'AGT-' . $user->nik . '-00' . User::count() + 1;
    $user->employeeStatus = $request->input('employeeStatus');
    $user->branchName = $request->input('branchName');
    $user->position = $request->input('position');
    $user->managerName = $request->input('managerName');
    $user->joinDate = $request->input('joinDate');
    $user->address = $request->input('address');
    $user->phoneNumber = $request->input('phoneNumber');
    $user->bankName = $request->input('bankName');
    $user->accountNumber = $request->input('accountNumber');
    $user->validationStatus = $request->input('validationStatus') ?? null;
    $user->registrationStatus = $request->input('registrationStatus') ?? null;
    $user->status = $request->input('status') ?? null;
    if ($user->validationStatus === null) {$user->validationStatus = "On-Process";}
    if ($user->registrationStatus === null) {$user->registrationStatus = "On-Process";}
    if ($user->status === null) {$user->status = "INACTIVE";}
    $user->save();
    return response(["message" => "User berhasil ditambahkan", "data" => $user], 200);
}
```

Segmen Program 4. 3 Kode controller pendaftaran

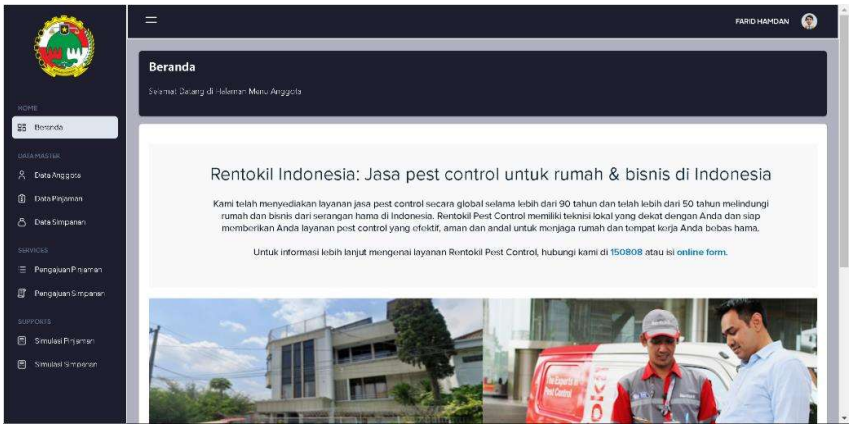
Fungsi userAdd bertanggung jawab untuk menambahkan pengguna baru ke dalam basis data. Jika properti validationStatus, registrationStatus, atau status tidak disediakan dalam permintaan, maka nilai default yang diberikan adalah "On-Process" untuk validationStatus dan registrationStatus, serta "INACTIVE" untuk status. Kemudian, objek pengguna disimpan dalam basis data, dan server merespons dengan pesan bahwa pengguna berhasil ditambahkan bersama dengan data pengguna yang baru ditambahkan.

4.1.3.4. Halaman Menu utama



Gambar 4. 9 Halaman menu utama admin

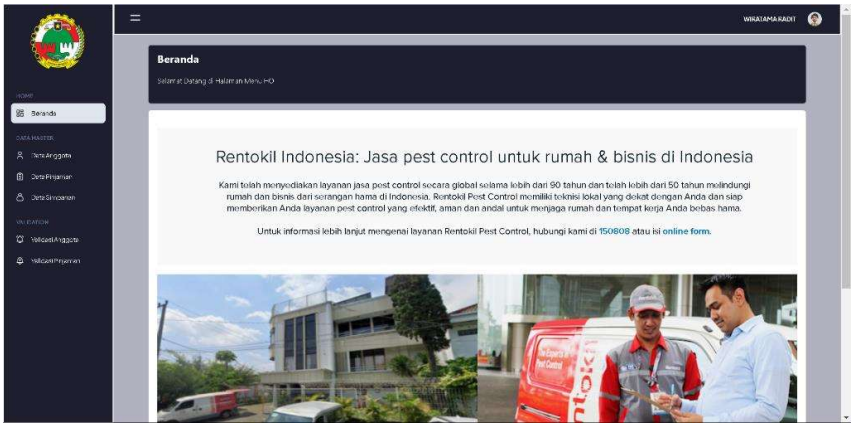
Tampilan menu jika pengguna login sebagai admin. Pada halaman admin terdapat menu data anggota, menu data pinjaman, menu data simpanan, menu validasi anggota, menu validasi pinjaman, serta menu validasi simpanan.



Gambar 4. 10 Halaman menu utama anggota

Tampilan menu jika pengguan login sebagai anggota. Pada menu anggota terdapat data anggota, data pinjaman, data simpanan,

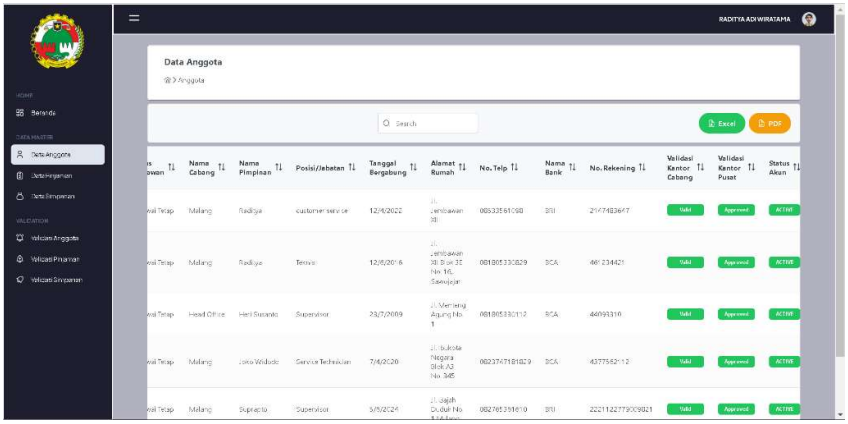
pengajuan pinjaman, pengajuan simpanan, simulasi pinjaman dan simulasi simpanan.



Gambar 4. 11 Halaman menu utama kantor pusat

Tampilan menu jika pengguan login sebagai kantor pusat. Pada halaman kantor pusat terdapat menu data anggota, menu data pinjaman, menu data simpanan, menu validasi anggota, menu validasi pinjaman.

4.1.3.5. Halaman Data Anggota

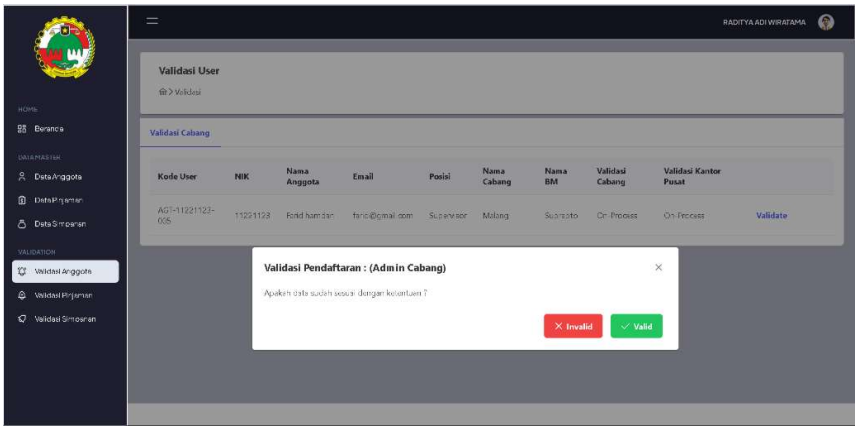


Gambar 4. 12 Halaman data anggota

Halaman untuk melihat data semua anggota koperasi yang sudah terdaftar maupun pendaftar yang masih dalam proses validasi. Terdapat

kolom kode anggota, nama, email, role, nik, status karyawan, nama cabang, nama pimpinan, posisi/jabatan, tanggal bergabung, Alamat rumah, no telp, nama bank, no rekening, status validasi kantor cabang, status validasi kantor pusat, status akun.

4.1.3.6. Halaman validasi anggota

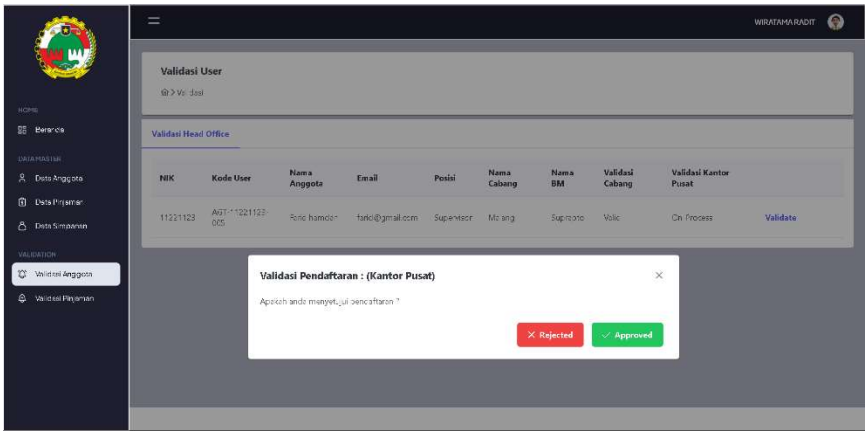


Gambar 4. 13 Halaman validasi anggota oleh admin

Merupakan tampilan proses validasi yang dilakukan oleh admin koperasi di cabang. Terdapat 2 pilihan validasi, yaitu Invalid, dan Valid. Jika data sudah sesuai ketentuan maka admin akan menyatakan pendaftaran anggota sebagai *valid*. Namun jika data tidak sesuai maka admin berhak untuk menolak (*Invalid*) dan pengguna yang melakukan pendaftaran tidak bisa melakukan login.

```php
function userValidationGeneral(Request $request, $id)
{
    $user = User::find($id);
    if (!$user) {
        return response(["message" => "User tidak ditemukan"], 400);
    }
    $registrationStatus = $request->input('registrationStatus');
    if ($registrationStatus === "Approved") {
        $user->registrationStatus = "Approved";
        $user->status = "ACTIVE";
        $user->save();
        return response(["message" => 'Anda disetujui bergabung menjadi anggota koperasi',
            'data' => $user], 200);
    } elseif ($registrationStatus === "Rejected") {
        $user->registrationStatus = "Rejected";
        $user->status = "INACTIVE";
        $user->save();
        return response(["message" => 'Anda ditolak bergabung menjadi anggota koperasi',
            'data' => $user], 400);
    }
    $user->save();
    return response(["message" => "Anda ditolak bergabung menjadi anggota koperasi",
        "data" => $user], 400);
}
```

Segmen Program 4. 4 Kode controller validasi cabang



Gambar 4. 14 Halaman validasi anggota oleh kantor pusat

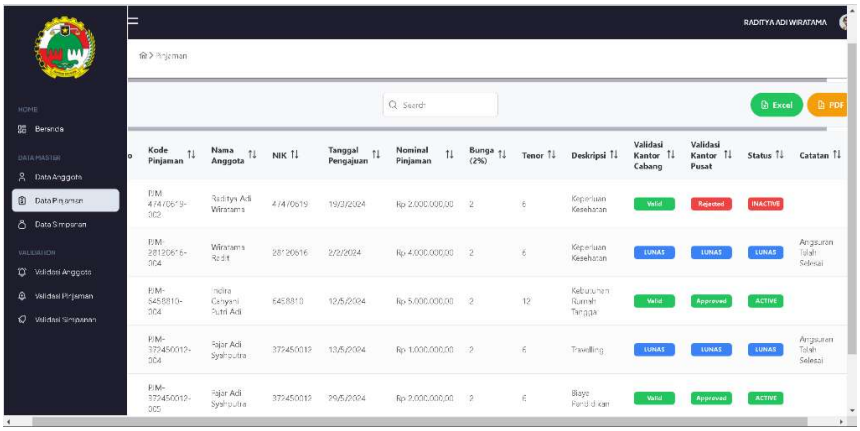
Merupakan tampilan proses validasi yang dilakukan oleh petugas kantor pusat. Sesudah admin melakukan proses validasi dan menyatakan data valid maka akan dilanjutkan dengan proses validasi oleh petugas kantor pusat. Terdapat 2 opsi validasi yaitu, Rejected, dan Approved. Ketika hasil validasi adalah “Rejected” maka status akun akan “INACTIVE” dan pendaftar tidak bisa melakukan login,

sedangkan jika hasil validasi adalah “Approved” maka status akun berubah menjadi “ACTIVE” dan pendaftar bisa melakukan login karena telah menjadi anggota koperasi.

```php
function userValidationGeneral(Request $request, $id)
{
    $user = User::find($id);
    if (!$user) {
        return response(["message" => "User tidak ditemukan"], 400);
    }
    $registrationStatus = $request->input('registrationStatus');
    if ($registrationStatus === "Approved") {
        $user->registrationStatus = "Approved";
        $user->status = "ACTIVE";
        $user->save();
        return response(["message" => 'Anda disetujui bergabung menjadi anggota koperasi',
            'data' => $user], 200);
    } elseif ($registrationStatus === "Rejected") {
        $user->registrationStatus = "Rejected";
        $user->status = "INACTIVE";
        $user->save();
        return response(["message" => 'Anda ditolak bergabung menjadi anggota koperasi',
            'data' => $user], 400);
    }
    $user->save();
    return response(["message" => "Anda ditolak bergabung menjadi anggota koperasi",
        "data" => $user], 400);
}
```

Segmen Program 4. 5 Kode controller validasi kantor pusat

4.1.3.7. Halaman Data Pinjaman

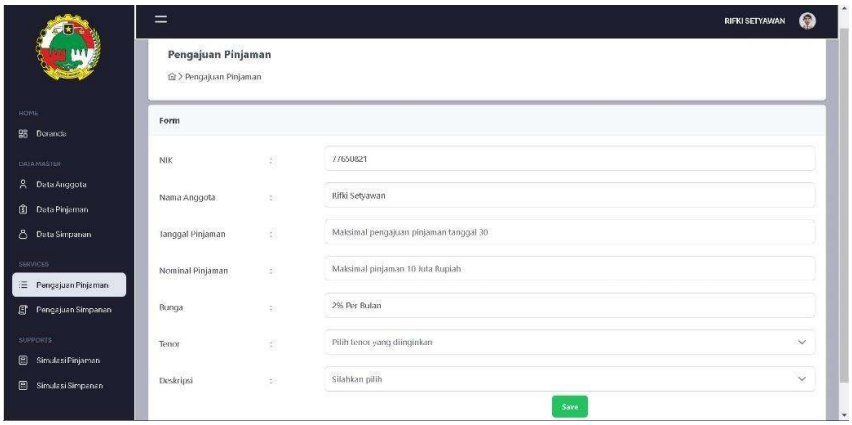


Gambar 4. 15 Halaman Data Pinjaman

Merupakan halaman untuk melihat semua data pinjaman yang sudah aktif maupun pengajuan pinjaman yang masih dalam proses validasi. Terdapat *field* kode pinjaman, nama anggota, nik, tanggal

pengajuan, nominal pinjaman, bunga, tenor, deskripsi, status validasi kantor cabang, status validasi kantor pusat, status pinjaman, dan catatan.

4.1.3.8. Halaman Pengajuan Pinjaman



Gambar 4. 16 Halaman pengajuan pinjaman

Merupakan halaman pengajuan pinjaman untuk anggota koperasi, pada halaman ini anggota koperasi melakukan input tanggal pinjaman, nominal pinjaman, tenor pinjaman, dan deskripsi. Kolom NIK, Nama akan terisi otomatis sesuai dengan data anggota yang login.

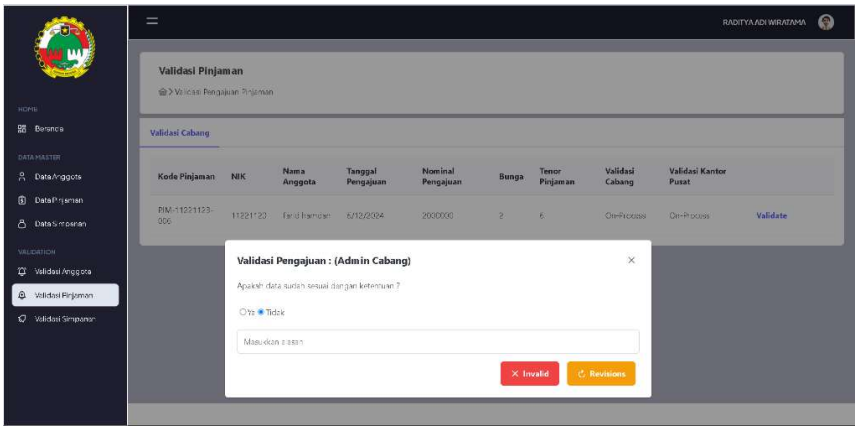
```php
public function loanAdd(Request $request)
{
    $user = User::find($request->input('userId'));

    $existingLoan = $user->loans()->where('status', 'ACTIVE')->first();
    if ($existingLoan) {
        return response([
            "message" => "Anda sudah memiliki pinjaman aktif,
            tidak dapat melakukan pengajuan pinjaman lagi."], 400);
    }
    $loan = new loan;
    $user = User::find($request->input('userId'));
    $loan->userId = $request->input('userId');
    $loan->code = 'PJM-' . $user->nik . '-00' . loan::count() + 1;
    $loan->nominal = $request->input('nominal');
    $loan->interest = $request->input('interest');
    $loan->tenor = $request->input('tenor');
    $loan->date = $request->input('date');
    $loan->description = $request->input('description');
    $loan->validationLoanStatus = $request->input('validationLoanStatus') ?? null;
    $loan->loanStatus = $request->input('loanStatus') ?? null;
    $loan->status = $request->input('status') ?? null;
    if ($loan->validationLoanStatus === null) {$loan->validationLoanStatus = "On-Process";
    }if ($loan->loanStatus === null) {$loan->loanStatus = "On-Process";
    }if ($loan->status === null) {$loan->status = "INACTIVE";
    }
    $loan->reason = $request->input('reason');
    $loan->save();
    return response(["message" => "Berhasil melakukan pengajuan pinjaman", "data" => $user], 200);
}
```

Segmen Program 4. 6 Kode controller pengajuan pinjaman

Fungsi loanAdd bertanggung jawab untuk menambahkan pinjaman baru ke dalam basis data. Jika properti validationLoanStatus, loanStatus, atau status tidak disediakan dalam permintaan, maka nilai default yang diberikan adalah "On-Process" untuk validationLoanStatus dan loanStatus, serta "INACTIVE" untuk status. Sebelum pinjaman disimpan kode akan memvalidasi apakah pengguna memiliki pinjaman “ACTIVE”, jika iya maka pengajuan pinjaman akan ditolak.

4.1.3.9. Halaman Validasi Pinjaman Admin



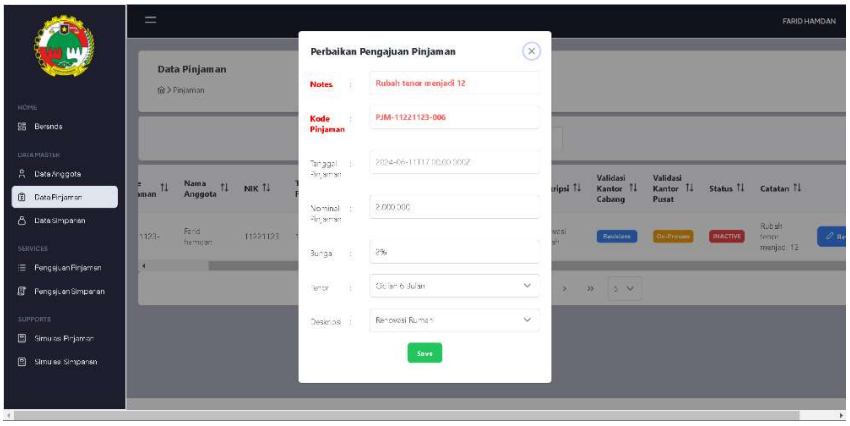
Gambar 4. 17 Halaman validasi pinjaman oleh admin

Terdapat jendela pop-up "Validasi Pengajuan: (Admin Cabang)" yang memungkinkan admin cabang untuk memvalidasi data pengajuan pinjaman. Pop-up ini memiliki opsi "Ya" atau "Tidak" untuk mengonfirmasi apakah data telah sesuai, serta kolom untuk memasukkan alasan jika diperlukan. Terdapat juga tombol "Invalid", "Valid", dan "revisions" untuk menindaklanjuti validasi.

```php
function loanValidationRegion(Request $request, $id)
{
    $loan = loan::find($id);
    if (!$loan) {
        return response(["message" => "Pinjaman tidak ditemukan"], 404);
    }
    $validationLoanStatus = $request->input('validationLoanStatus');
    $reason = $request->input('reason');
    if ($validationLoanStatus === "Valid") {
        $loan->validationLoanStatus = "Valid";
        $loan->save();
        return response(['message' => 'Data sudah sesuai', 'data' => $loan], 200);
    } elseif ($validationLoanStatus === "Invalid") {
        $loan->validationLoanStatus = "Invalid";
        $loan->reason = $reason;
        $loan->save();
        return response(['message' => 'Data tidak sesuai', 'data' => $loan], 200);
    } elseif ($validationLoanStatus === "Revisions") {
        $loan->validationLoanStatus = "Revisions";
        $loan->reason = $reason;
        $loan->save();
        return response(["message" => 'Pinjaman anda sedang di revisi', 'data' => $loan], 200);
    }
    return response(['message' => 'Data tidak sesuai', 'data' => $loan], 400);
}
```

Segmen Program 4. 7 Kode controller validasi pinjaman oleh admin

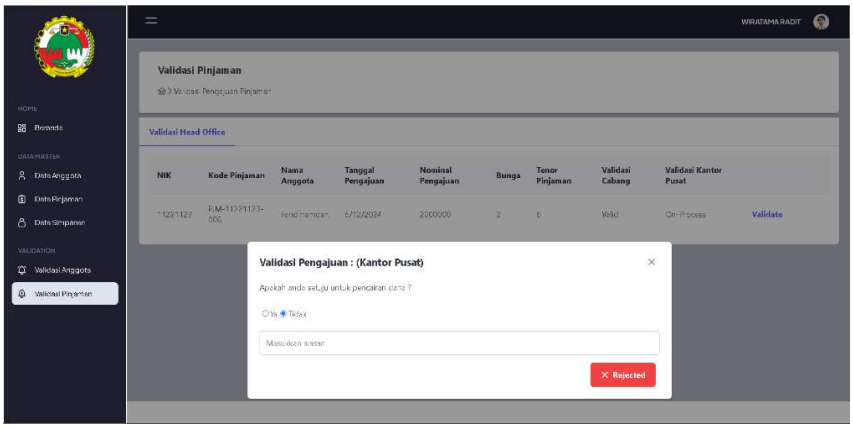
4.1.3.10. Halaman Revisi Pengajuan Pinjaman



Gambar 4. 18 Tampilan revisi pinjaman

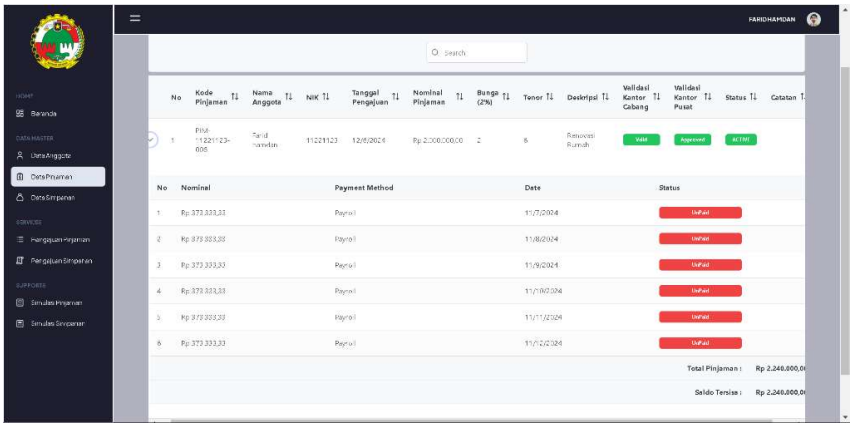
Merupakan tampilan revisi pinjaman untuk anggota koperasi, pada halaman ini anggota koperasi mendapatkan status validasi revisi dari admin cabang. Ketika status berubah menjadi revisi maka akan muncul alasan revisi dan button untuk melakukan perubahan data yang dilakukan anggota koperasi. Setelah anggota melakukan perubahan data, maka admin koperasi akan memvalidasi ulang.

4.1.3.11. Halaman Validasi Pinjaman Kantor Pusat



Gambar 4. 19 Halaman validasi pinjaman kantor pusat

Terdapat jendela pop-up "Validasi Pengajuan: (Kantor Pusat)" yang memungkinkan petugas kantor pusat untuk menyetujui pencairan dana. Terdapat juga tombol "Rejected", "Approved" untuk menindaklanjuti proses validasi. Setelah kantor pusat menyetujui pencairan dana, maka status akan berubah menjadi “Approved”. Ketika status berubah menjadi “Approved”, maka data angsuran pinjaman akan otomatis muncul. Dapat dilihat pada gambar 4.20 Tampilan data angsuran.



Gambar 4. 20 Tampilan data angsuran

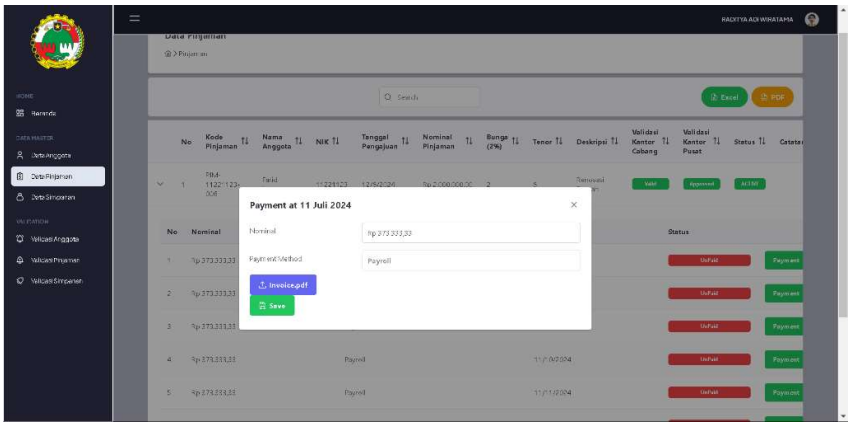
Terdapat status “UnPaid” ketika angsuran belum terbayar dan terdapat button payment untuk melakukan pembayaran.

```php
function loanValidationGeneral(Request $request, $id)
{
    $loan = Loan::find($id);
    if (!$loan) {
        return response(["message" => "Pinjaman tidak ditemukan"], 400);
    }
    $loanStatus = $request->input('loanStatus');
    $reason = $request->input('reason');
    if ($loanStatus === "Approved") {
        $loan->loanStatus = "Approved";
        $loan->status = "ACTIVE";
        $loan->save();
        // Membuat angsuran
        $installments = [];
        for ($i = 1; $i <= $loan->tenor; $i++) {
            $installment = new Installment;
            $installment->loanId = $loan->id;
            $installment->userId = $loan->userId;
            $installment->nominalPayment = ($loan->nominal / $loan->tenor) + ($loan->nominal * 0.02);
            $installment->paymentMethod = 'Payroll';
            $installment->paymentStatus = $request->input('paymentStatus') ?? 'UnPaid';
            $installment->installmentStatus = $request->input('installmentStatus') ?? 'On-Process';
            $installment->date = Carbon::parse($loan->date)->addMonths($i);
            $installment->save();
            $installments[] = $installment;
        }
        return response()->json([
            "message" => "Berhasil melakukan pengajuan pinjaman",
            "data" => [
                "loan" => $loan,
                "installment" => $installments
            ]
        ], 200);
    } elseif ($loanStatus === "Rejected") {
        $loan->loanStatus = "Rejected";
        $loan->reason = $reason;
        $loan->status = "INACTIVE";
        $loan->save();
        return response(["message" => 'Pinjaman anda tidak disetujui', 'data' => $loan], 400);
    }

    return response(["message" => "Pinjaman anda ditolak", "data" => $loan], 400);
}
```

Segmen Program 4. 8 Kode controller validasi pinjaman kantor pusat

4.1.3.12. Tampilan Upload Bukti Pembayaran Angsuran



Gambar 4. 21 Tampilan Upload Bukti Pembayaran Angsuran

Merupakan tampilan ketika button payment ditekan. Terdapat informasi nominal angsuran yang dibayarkan dan upload file berupa invoice yang berformat PDF. Proses pembayaran pada koperasi karyawan di PT. Rentokil Indonesia menggunakan sistem payroll atau pemotongan gaji anggota. Maka untuk sistem pembayaran pada aplikasi hanya melampirkan bukti pemotongan gaji untuk pembayaran angsuran anggota koperasi. Upload bukti pembayaran hanya dilakukan oleh admin cabang.

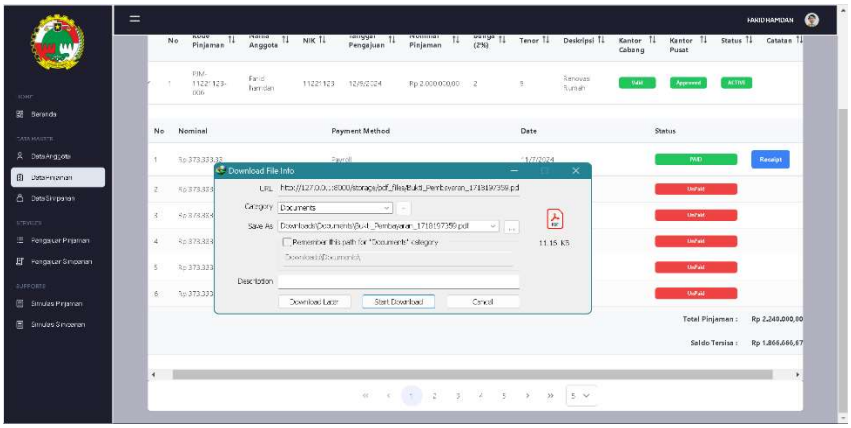
```php
function loantransAdd(Request $request)
{
    $installmentId = $request->input('installmentId');
    $installment = installment::find($installmentId);
    if ($request->hasFile('pdf')) {
        $pdfFile = $request->file('pdf');
        $fileName = 'Bukti_Pembayaran_' . time() . '.' . $pdfFile->getClientOriginalExtension();
        $filePath = $pdfFile->storeAs('public/pdf_files', $fileName);
        $loantrans = new loantransaction;
        $loantrans->installmentId = $installment->id;
        $loantrans->fileName = $fileName;
        $loantrans->receipt = asset('storage/pdf_files/' . $fileName);

        $loantrans->save();
        $installmentQuery = installment::query()->where('id', $installmentId)->first();
        $installmentQuery->paymentStatus = 'PAID';
        $installmentQuery->installmentStatus = 'Lunas';
        $installmentQuery->save();

        return response()->json(['message' => 'File berhasil diunggah.', 'data' => $loantrans], 200);
    } else {

        return response()->json(['message' => 'Tidak ada file yang diunggah.'], 400);
    }
}
```

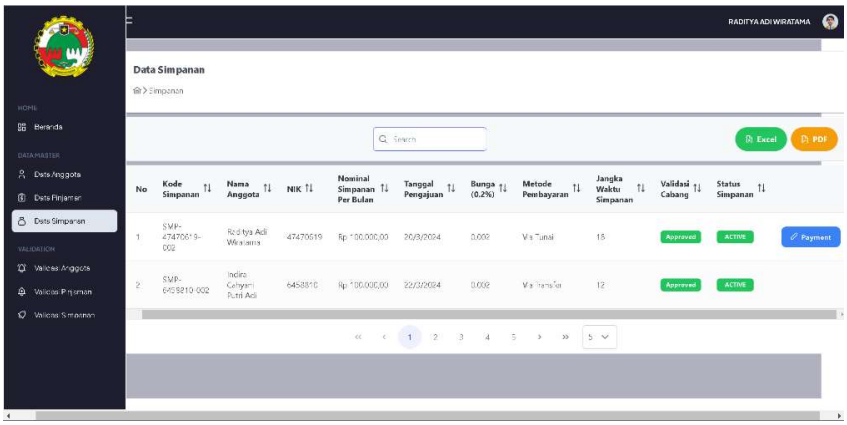
Segmen Program 4. 9 Kode controller pembayaran pinjaman.



Gambar 4. 22 Tampilan unduh bukti pembayaran

Pada gambar 4.22 merupakan tampilan untuk unduh bukti pembayaran angsuran. Setelah melakukan upload bukti pembayaran, status akan berubah menjadi “PAID” dan muncul button untuk unduh bukti pembayaran. Unduh bukti pembayaran dilakukan oleh anggota.

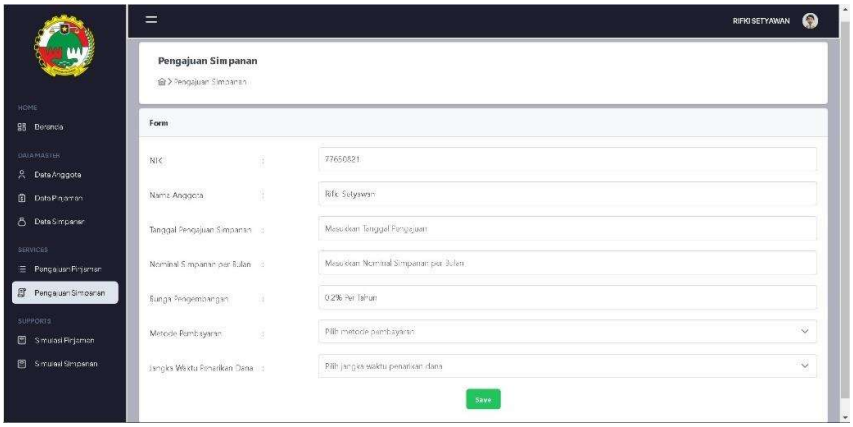
4.1.3.13. Halaman Data Simpanan



Gambar 4. 23 Halaman Data Simpanan

Merupakan halaman untuk melihat semua data simpanan yang sudah aktif maupun pengajuan simpanan yang masih dalam proses validasi. Terdapat *field* kode simpanan, nama anggota, nik, nominal simpanan per bulan, tanggal pengajuan, bunga, metode pembayaran, jangka waktu simpanan, status validasi kantor cabang, status simpanan.

4.1.3.14. Halaman Pengajuan Simpanan



Gambar 4. 24 Halaman pengajuan simpanan

Merupakan halaman pengajuan simpanan untuk anggota koperasi, pada halaman ini anggota koperasi melakukan input tanggal simpanan, nominal simpanan per bulan, metode pembayaran, dan jangka waktu simpanan. Kolom NIK, Nama akan terisi otomatis sesuai dengan data anggota yang login.

```
function saveAdd(Request $request)
{
    $user = User::find($request->input('userId'));

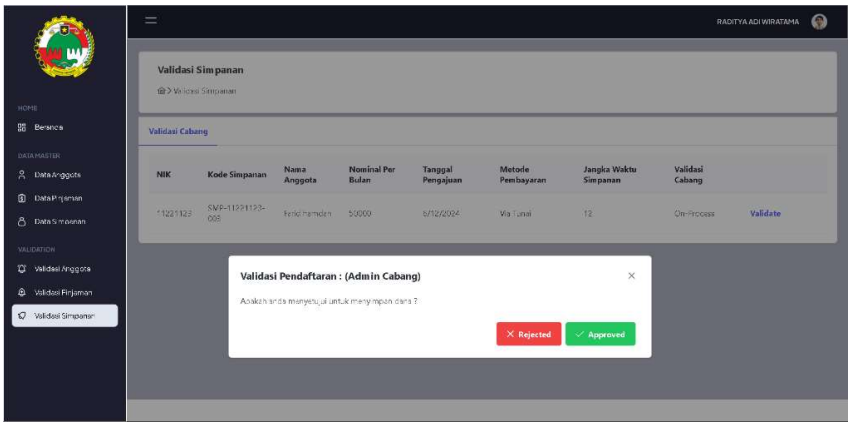
    $existingLoan = $user->loans()->where('status', 'ACTIVE')->first();
    if ($existingLoan) {
        return response([
            "message" => "Anda sudah memiliki simpanan aktif, tidak dapat melakukan pengajuan simpanan lagi."
        ], 400);
    }
    $save = new saving;
    $user = User::find($request->input('userId'));
    $save->userId = $request->input('userId');
    $save->code = 'SMP-' . $user->nik . '-00' . saving::count() + 1;
    $save->nominalPerMonth = $request->input('nominalPerMonth');
    $save->interest = $request->input('interest');
    $save->date = $request->input('date');
    $save->paymentMethod = $request->input('paymentMethod');
    $save->timePeriod = $request->input('timePeriod');
    $save->validationSavingStatus = $request->input('validationSavingStatus') ?? null;
    $save->status = $request->input('status') ?? null;
    if ($save->validationSavingStatus === null) {
        $save->validationSavingStatus = "On-Process";
    }
    if ($save->status === null) {
        $save->status = "INACTIVE";
    }

    $save->save();
    return response()->json(["message" => "Berhasil melakukan pengajuan simpanan", "data" => $save], 200);
}
```

Segmen Program 4. 10 Kode controller pengajuan simpanan

Fungsi saveAdd bertanggung jawab untuk menambahkan simpanan baru ke dalam basis data. Jika properti validationSavingStatus tidak disediakan dalam permintaan, maka nilai default yang diberikan adalah "On-Process" untuk validationSavingStatus dan "INACTIVE" untuk status. Sebelum simpanan disimpan kode akan memvalidasi apakah pengguna memiliki simpanan “ACTIVE”, jika iya maka pengajuan simpanan akan ditolak.

4.1.3.15. Halaman Validasi Simpanan



Gambar 4. 25 Halaman validasi simpanan oleh admin

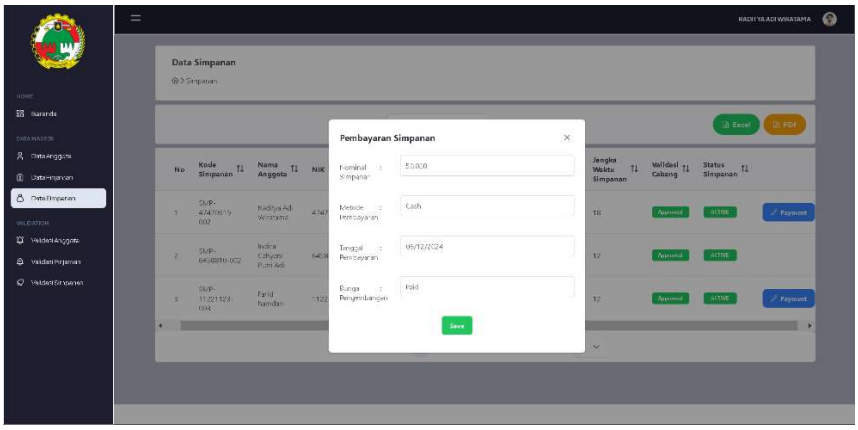
Merupakan tampilan proses validasi yang dilakukan oleh admin koperasi di cabang. Terdapat 2 pilihan validasi, yaitu *Rejected*, dan *Approved*. Jika data sudah sesuai ketentuan maka admin akan menyatakan pengajuan simpanan sebagai *Approved*. Namun jika data tidak sesuai maka admin berhak untuk menolak (*Rejected*).

```php
function saveValidationRegion(Request $request, $id)
{
    $save = saving::find($id);
    if (!$save) {
        return response(["message" => "Simpanan tidak ditemukan"], 404);
    }
    $validationSavingStatus = $request->input('validationSavingStatus');
    if ($validationSavingStatus === "Approved") {
        $save->validationSavingStatus = "Approved";
        $save->status = "ACTIVE";
        $save->save();
        return response(['message' => 'Data sudah sesuai', 'data' => $save], 200);
    } elseif ($validationSavingStatus === "Rejected") {
        $save->validationSavingStatus = "Rejected";
        $save->save();
        return response(['message' => 'Data tidak sesuai', 'data' => $save], 200);
    }

    return response(['message' => 'Data tidak sesuai', 'data' => $save], 400);
}
```

Segmen Program 4. 11 Kode controller validasi simpanan

4.1.3.16. Tampilan Input Data Pembayaran Simpanan



Gambar 4. 26 Tampilan pembayaran simpanan

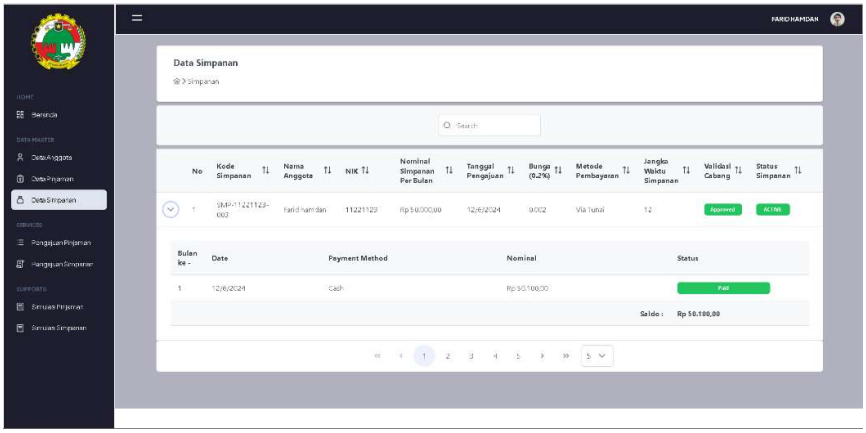
Merupakan halaman pembayaran simpanan. Setelah anggota koperasi melakukan pembayaran kepada admin koperasi, maka admin akan mengisi data pembayaran kedalam aplikasi sebagai informasi kepada anggota.

```php
function savepayAdd(Request $request)
{
    $saveId = $request->input('saveId');
    $save = saving::find($saveId);
    $userId = $request->input('userId');
    $user = User::find($userId);

    if (!$save) {
        return response()->json(['error' => 'Data saving tidak ditemukan.'], 404);
    }
    $savepay = new savingpayment;
    $savepay->saveId = $saveId;
    $savepay->userId = $userId;
    $savepay->nominal = $request->input('nominal');
    $savepay->payment = ($savepay->nominal * $save->interest) + ($savepay->nominal);
    $savepay->paymentMethod = 'Cash';
    $savepay->date = $request->input('date');
    $savepay->status = 'Paid';
    $savepay->save();

    return response()->json(["message" => "Pembayaran Berhasil"], 200);
}
```

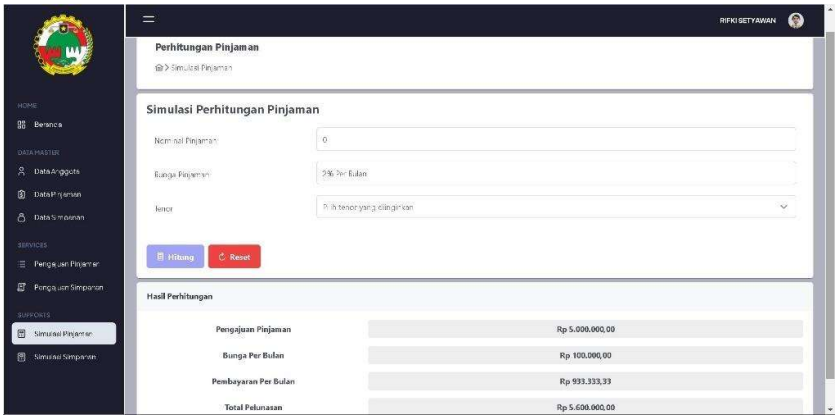
Segmen Program 4. 12 Kode controller pembayaran simpanan



Gambar 4. 27 Tampilan data pembayaran simpanan

Merupakan tampilan pembayaran simpanan yang sudah selesai sesuai dengan jangka waktu yang dipilih oleh anggota. Jika pembayaran simpanan sudah selesai sesuai dengan jangka waktu, maka button payment akan hilang.

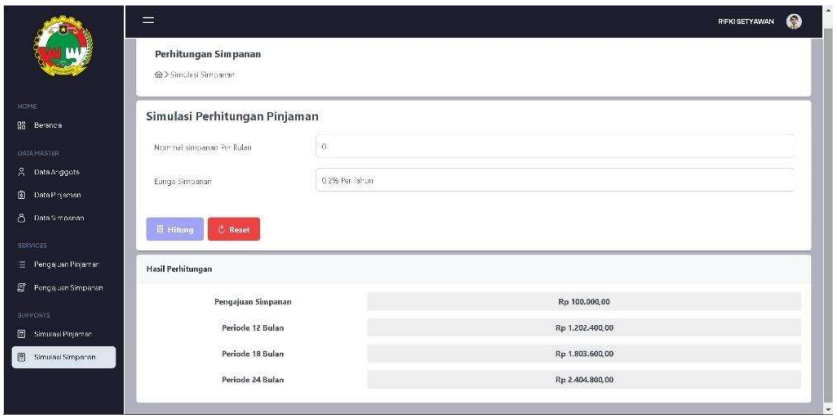
4.1.3.17. Halaman Simulasi Pinjaman



Gambar 4. 28 Halaman Simulasi Pinjaman

Merupakan halaman simulasi pinjaman untuk membantu anggota koperasi dalam memperkirakan besaran angsuran sesuai dengan besaran nominal pinjaman yang dipilih.

4.1.3.18. Halaman Simulasi Simpanan



Gambar 4. 29 Halaman Simulasi Pinjaman

Merupakan halaman simulasi simpanan untuk membantu anggota koperasi dalam memperkirakan besaran saldo yang diterima sesuai dengan jangka waktu yang dipilih.

4.2. Uji Coba

4.2.1. Tahap Pengujian

Pada tahap ini penulis menguji aplikasi ini menggunakan metode black box testing. Metode yang juga dikenal sebagai pengujian perilaku (behavioral testing). Dalam metode ini, penguji berperan sebagai pengguna akhir, mengevaluasi perangkat lunak berdasarkan perilaku dan output yang dihasilkan, tanpa perlu memahami cara kerja internal sistem. Pendekatan ini memungkinkan penguji untuk mengidentifikasi masalah dari perspektif pengguna, dan memastikan aplikasi dapat berjalan sesuai dengan yang diharapkan dalam berbagai skenario penggunaan.

4.2.2. Hasil Pengujian

Menyajikan hasil verifikasi pengujian aplikasi apakah hasil dari input yang dimasukkan menghasilkan output sesuai dengan yang diharapkan. Berikut adalah hasil pengujian aplikasi sebagaimana yang telah dijelaskan pada tabel 4.3.

Tabel 4. 3 Hasil Pengujian

No	Nama Fungsi	Deskripsi Kegunaan	Target Output	Hasil Pengujian
1.	Login	Melakukan pengecekan apakah user berhak masuk/akses ke sistem atau tidak.	Jika berhasil maka akan menampilkan menu utama.	Sukses
2.	Pendaftaran Anggota	Melakukan pendaftaran anggota baru yang dilakukan oleh karyawan tetap.	Jika calon anggota berhasil mengisi data maka data akan tersimpan ke database dan tampil di menu data anggota untuk menunggu proses validasi. Dengan status validasi cabang “On-Process”, validasi kantor pusat “On-Process” dan status “INACTIVE”.	Sukses
3.	Validasi Anggota oleh admin,	Melakukan pemeriksaan data dan memvalidasi calon anggota baru.	Jika admin menyetujui maka status validasi cabang akan berubah menjadi “Valid”, dan dilanjutkan ke proses validasi oleh kantor pusat. Jika admin menolak maka status validasi cabang berubah menjadi “Invalid”. Dengan status tetap “INACTIVE”.	Sukses

4.	Validasi Anggota oleh kantor pusat	Melakukan pemeriksaan data dan memvalidasi calon anggota baru.	Jika kantor pusat menyetujui maka status validasi akan berubah menjadi “Approved” dan status anggota menjadi “ACTIVE”. Jika status “ACTIVE” pendaftar diterima menjadi anggota koperasi, dan bisa melakukan login ke dalam aplikasi. Jika kantor pusat menolak maka status validasi menjadi “Rejected”, dan status tetap “INACTIVE”.	Sukses
5.	Pengajuan Pinjaman	Melakukan pengajuan pinjaman yang dilakukan oleh anggota.	Jika anggota berhasil mengisi data, maka data akan tersimpan ke database dan tampil di menu data pinjaman untuk menunggu proses validasi. Dengan status validasi cabang “On-Process”, validasi kantor pusat “On-Process” dan status “INACTIVE”.	Sukses
6.	Validasi Pinjaman oleh admin	Melakukan pemeriksaan data dan memvalidasi pengajuan pinjaman oleh anggota.	Jika admin menyetujui maka status validasi cabang akan berubah menjadi “Valid”, dan dilanjutkan ke proses validasi oleh kantor pusat. Jika admin menyatakan untuk revisi maka validasi cabang akan berubah menjadi “Revisions” data pinjaman akan muncul catatan revisi yang diisi oleh admin dan tombol revisi	Sukses

			untuk dilakukan perubahan data oleh anggota. Jika admin menolak maka status validasi cabang berubah menjadi "Invalid". Dengan status tetap "INACTIVE".	
7.	Revisi Pinjaman	Memungkinkan anggota untuk merevisi pengajuan pinjaman jika diperlukan.	Ketika tombol ditekan maka akan muncul data pinjaman anggota yang disi sebelumnya. Anggota bisa melakukan perubahan data. Pengajuan pinjaman yang direvisi berhasil disubmit kembali dan menunggu validasi kembali dengan status validasi cabang "On-Process".	Sukses
8.	Validasi Pinjaman oleh kantor pusat	Melakukan pemeriksaan data dan memvalidasi pengajuan pinjaman oleh anggota.	Jika kantor pusat menyetujui maka status validasi akan berubah menjadi "Approved" maka sistem akan membuat angsuran secara otomatis dan muncul tombol payment untuk upload bukti pembayaran dengan status pembayaran "UnPaid". Status pinjaman menjadi "ACTIVE". Jika kantor pusat menolak maka status validasi kantor pusat menjadi "Rejected", dan status tetap "INACTIVE".	Sukses

9.	Upload Bukti Pembayaran	Melakukan unggah bukti pemotongan gaji untuk pembayaran angsuran dengan maksimal upload dokumen 2MB.	Bukti pembayaran berhasil diunggah ke dalam sistem dan database. Jika pembayaran berhasil maka tombol payment berubah menjadi tombol receipt dan status pembayaran berubah menjadi "PAID". Jika pembayaran angsuran selesai, maka validasi cabang, validasi kantor pusat, dan status menjadi "LUNAS".	Sukses
10.	Download Bukti Pembayaran	Melakukan unduh bukti pembayaran oleh anggota, admin, dan kantor pusat.	Bukti pembayaran dapat diunduh dengan sukses.	Sukses
11.	Pengajuan Simpanan	Melakukan pengajuan simpanan yang dilakukan oleh anggota.	Jika anggota berhasil mengisi data, maka data akan tersimpan ke database dan tampil di menu data simpanan untuk menunggu proses validasi. Dengan status validasi cabang "On-Process", dan status "INACTIVE".	Sukses
12	Validasi Simpanan oleh admin	Melakukan pemeriksaan data dan memvalidasi pengajuan simpanan oleh anggota.	Jika admin menyetujui maka status validasi cabang akan berubah menjadi "Valid", status menjadi "ACTIVE", dan muncul tombol payment untuk pembayaran. Jika admin menolak maka status validasi cabang berubah menjadi "Invalid".	Sukses

			Dengan status tetap "INACTIVE".	
13	Input pembayaran simpanan	Melakukan pengisian data pembayaran ke sistem, setelah anggota membayar simpanan secara tunai/transfer ke admin.	Pembayaran simpanan berhasil diinput ke dalam sistem dengan status "Paid". Ketika jumlah pembayaran sudah sesuai dengan jangka waktu simpanan, maka tombol payment akan hilang. Validasi cabang, dan status akan berubah menjadi "Selesai".	Sukses
14	Simulasi Pinjaman	Melakukan perhitungan besaran angsuran sesuai dengan pinjaman yang akan diajukan dan tenor yang dipilih.	Menampilkan hasil perhitungan Bunga per bulan, Pembayaran Per Bulan, dan Total Penulasan. Ketika ditekan tombol reset maka data akan hilang.	Sukses
15	Simulasi Simpanan	Melakukan perhitungan besaran simpanan sesuai dengan jangka waktu yang tersedia.	Menampilkan hasil perhitungan simpanan dengan periode waktu 12 Bulan, 18 Bulan, dan 24 Bulan.	Sukses
16	Logout	Mengakhiri sesi pengguna pada aplikasi.	Jika berhasil akan menampilkan halaman login.	Sukses