

## **BAB 3**

### **PELAKSANAAN KERJA MAGANG**

#### **3.1 Kedudukan dan Koordinasi**

Kegiatan magang ini dilaksanakan pada divisi *Community Engagement Apps Development* yang berada di bawah naungan *Research, Innovation, and Sustainability* (RIS) Universitas Multimedia Nusantara (UMN). Divisi ini memiliki tanggung jawab dalam pengembangan dan pemeliharaan sistem serta aplikasi berbasis web yang ditujukan untuk mendukung kegiatan pengabdian kepada masyarakat di lingkungan universitas.

Dalam pelaksanaan magang, penulis berperan sebagai *Backend Developer* dengan sistem kerja *Work From Home* (WFH). Penulis melanjutkan pengembangan proyek sistem informasi Koperasi Simpan Pinjam Wiyata Mandala yang telah dikembangkan oleh tim magang pada periode sebelumnya, dengan fokus utama pada modul pengajuan pinjaman.

Koordinasi selama kegiatan magang dilakukan bersama rekan satu tim dan pembimbing lapangan dari pihak RIS UMN. Proses komunikasi serta pembagian tugas dilaksanakan secara daring dengan memanfaatkan *Zoom Meeting* sebagai media rapat koordinasi, *WhatsApp Group* untuk komunikasi harian, serta *GitHub* sebagai platform kolaborasi dalam pengelolaan dan pengembangan kode sumber. Pembimbing lapangan berperan dalam memberikan arahan teknis, melakukan evaluasi terhadap hasil pengembangan, serta memastikan bahwa implementasi sistem berjalan sesuai dengan kebutuhan operasional koperasi dan tujuan pengembangan divisi.

#### **3.2 Tugas yang Dilakukan**

Pada periode magang ini, tugas penulis difokuskan sepenuhnya pada penyelesaian pekerjaan utama di sisi *backend* yang berpusat pada perbaikan dan pengembangan modul pengajuan peminjaman pada sistem informasi Koperasi Simpan Pinjam Wiyata Mandala. Pekerjaan ini mencakup penyesuaian logika bisnis, validasi aturan koperasi, serta integrasi perhitungan deduksi bulanan agar sesuai dengan ketentuan operasional koperasi. Secara umum, tugas yang dikerjakan dikelompokkan ke dalam dua fokus utama, yaitu perbaikan logika pengajuan dan penyesuaian mekanisme deduksi simpan pinjam.

### **3.2.1 Perbaikan Logika Pengajuan Simpan dan Pinjam**

Logika pengajuan simpan pinjam yang diimplementasikan oleh tim sebelumnya masih memiliki beberapa ketidaksesuaian terhadap aturan koperasi, terutama pada aspek validasi data dan batasan pengajuan. Oleh karena itu, dilakukan penyesuaian pada modul *backend* agar proses pengajuan dapat berjalan lebih akurat dan konsisten.

1. Tujuan: Menyesuaikan logika pengajuan simpan pinjam agar sesuai dengan ketentuan koperasi, termasuk fleksibilitas *input* tenor pinjaman dan pembatasan nominal maksimal pinjaman berdasarkan jenis pengajuan, terutama untuk pengajuan KPKA.
2. Cakupan: Penyesuaian sistem pengajuan pinjaman agar tenor bulan dapat diinput *user* secara *manual*, penerapan validasi nominal maksimum pada jenis pinjaman KPKA sebesar Rp. 25.000.000 di sisi *backend*, penambahan informasi status pinjaman pada halaman awal simpan pinjam, berupa keterangan angsuran ke- untuk meningkatkan kejelasan data bagi *user*.

### **3.2.2 Penyesuaian Perhitungan Deduksi Bulanan Simpan Pinjam**

Pada sistem sebelumnya, perhitungan deduksi bulanan masih digabung secara keseluruhan antara simpanan dan pinjaman, serta belum mengakomodasi skema deduksi khusus bagi pengurus koperasi. Hal ini menyebabkan perhitungan deduksi belum sepenuhnya mencerminkan kondisi riil transaksi anggota.

1. Tujuan: Memperbaiki mekanisme perhitungan deduksi bulanan agar hanya mencakup komponen simpanan, serta mengintegrasikan deduksi khusus bagi pengurus koperasi berdasarkan hasil pengajuan seluruh anggota pada periode berjalan.
2. Cakupan: Penyesuaian logika perhitungan deduksi bulanan pada halaman awal simpan pinjam agar hanya menghitung simpanan anggota, pengembangan opsi deduksi pengurus koperasi yang dihitung sebagai hasil agregasi pengajuan simpan pinjam seluruh anggota dalam satu bulan, integrasi deduksi pengurus ke dalam alur *backend* agar terproses secara konsisten bersama data pengajuan anggota.

### 3.3 Uraian Pelaksanaan Magang

Rangkaian kegiatan yang dilakukan selama pelaksanaan kerja magang disusun dan dirangkum berdasarkan pembagian waktu per minggu. Uraian pekerjaan yang telah dilaksanakan pada setiap minggu selama periode magang dapat disajikan dalam Tabel 3.1 berikut.

Tabel 3.1. Pekerjaan yang dilakukan dalam proses magang

Minggu ke-	Pekerjaan yang dilakukan
1	Melakukan pertemuan dengan supervisor magang dan tim developer proyek sebelumnya, pengenalan proyek dan serah terima kode proyek.
2-5	Melakukan peninjauan dan analisis logika bisnis modul pengajuan simpan pinjam pada sisi backend, termasuk mekanisme deduksi bulanan, validasi tenor pinjaman, serta batasan nominal pengajuan sesuai ketentuan koperasi.
6-7	Mengimplementasikan perbaikan dan penyesuaian logika backend pada modul pengajuan simpan pinjam, meliputi pemisahan perhitungan deduksi simpanan, penerapan input tenor pinjaman secara manual, validasi maksimal nominal pinjaman KP KA, serta penambahan informasi status angsuran pinjaman.
8-10	Melakukan pengujian fungsional, debugging, dan verifikasi hasil perhitungan deduksi serta pengajuan simpan pinjam untuk memastikan sistem berjalan sesuai aturan koperasi dan kebutuhan operasional pengguna.
11-15	Melakukan penyusunan dan perapihan dokumen pengembangan sistem serta penulisan laporan akhir magang, disertai analisis hasil pekerjaan yang telah dilakukan dan perumusan saran pengembangan sistem di masa mendatang.

Metode pengembangan yang dipakai dalam proses magang ini adalah model Waterfall yang selaras dengan tim magang pada periode sebelumnya. Berikut adalah 5 tahapan model *Waterfall* yang digunakan:

1. Analisis Kebutuhan (*Requirement Analysis*): Tahap awal dilakukan dengan meninjau dan menganalisis modul simpan pinjam yang telah dikembangkan

oleh tim magang sebelumnya. Analisis difokuskan pada identifikasi ketidaksesuaian logika bisnis dengan kebutuhan operasional koperasi, khususnya pada:

- Perhitungan deduksi bulanan yang masih menggabungkan simpanan, pinjaman, dan penarikan.
- Keterbatasan *input* tenor pinjaman yang bersifat statis.
- Tidak adanya pembatasan nominal maksimal pada jenis pinjaman tertentu (KPKA).
- Kurangnya informasi progres angsuran pada data pinjaman.
- Belum terintegrasinya mekanisme deduksi khusus pengurus koperasi.

2. Perancangan Sistem (*System Design*): Berdasarkan hasil analisis kebutuhan, dilakukan perancangan ulang logika sistem pada modul pengajuan simpan pinjam. Perancangan ini mencakup:

- Pemisahan logika deduksi agar hanya simpanan yang dihitung sebagai deduksi bulanan.
- Desain fleksibilitas *input* tenor pinjaman secara *manual* pada *form* pengajuan.
- Penetapan batas maksimal nominal pinjaman KPKA sebesar Rp. 25.000.000 pada *level* logika *backend*.
- Penambahan struktur informasi angsuran (angsuran ke-n dari total tenor).
- Perancangan mekanisme deduksi pengurus sebagai hasil agregasi pengajuan seluruh anggota dalam satu periode bulan.

3. Implementasi (*Implementation*): Tahap implementasi dilakukan pada sisi *backend* sistem menggunakan *Node.js*, *Express.js*, dan *PostgreSQL*. Implementasi meliputi:

- Logika Deduksi: Penyesuaian perhitungan deduksi bulanan pada modul Simpan Pinjam agar hanya mencakup simpanan anggota.
- Form Pengajuan Pinjaman: Implementasi *input* tenor bulan secara manual serta validasi batas maksimal nominal pinjaman KPKA.

- Informasi Pinjaman: Penambahan data keterangan angsuran ke-n pada *endpoint* pinjaman untuk ditampilkan pada halaman awal Simpan Pinjam.
- Deduksi Pengurus: Implementasi logika agregasi deduksi bulanan berdasarkan total pengajuan anggota dan integrasinya ke dalam modul pengurus.
- Integrasi Sistem: Penyesuaian *endpoint API* agar perubahan *backend* dapat digunakan oleh *frontend React* secara konsisten.

4. Pengujian (*Testing*): Setelah implementasi, dilakukan pengujian fungsional terhadap seluruh perubahan yang diterapkan. Pengujian mencakup:

- Validasi hasil perhitungan deduksi simpanan dan pinjaman.
- Pengujian batas nominal pinjaman dan *input tenor manual*.
- Verifikasi tampilan dan keakuratan informasi angsuran.
- Pengujian integrasi deduksi pengurus berdasarkan data agregat anggota.

5. Pemeliharaan (*Maintenance*) Tahap akhir meliputi kegiatan *debugging*, penyempurnaan logika berdasarkan hasil pengujian, serta penyusunan dokumentasi teknis. Selain itu, dilakukan penyesuaian kecil berdasarkan masukan dari pembimbing lapangan agar sistem lebih stabil, mudah dikembangkan, dan siap untuk dilanjutkan oleh tim pengembang berikutnya.

### 3.3.1 Sistem dan Lingkungan Kerja

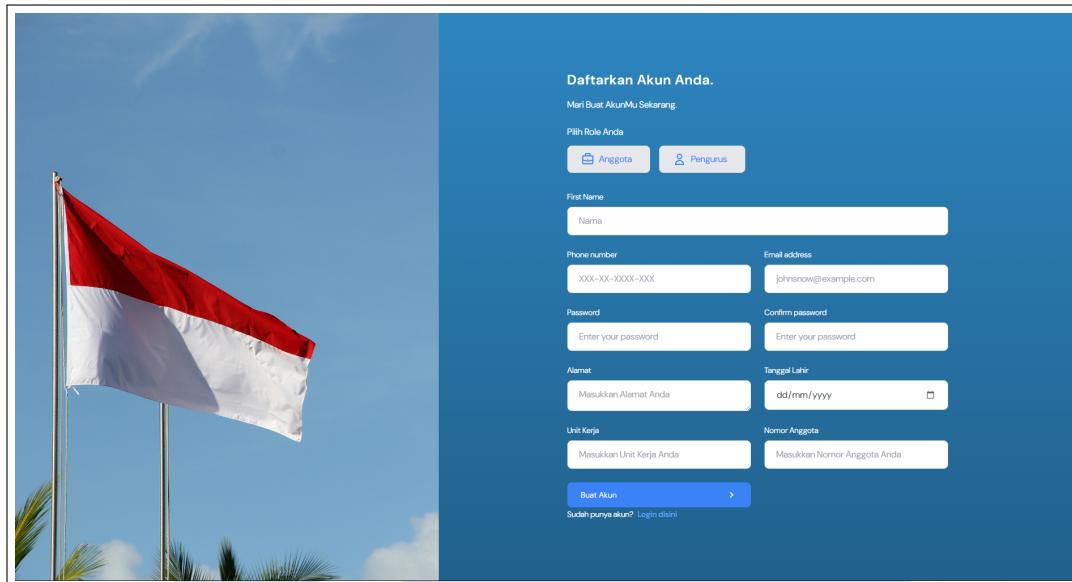
Seluruh pengembangan modul simpan pinjam dilakukan dengan memanfaatkan teknologi *backend* yang telah ditetapkan pada proyek Sistem Informasi Koperasi Simpan Pinjam Wiyata Mandala. Pemilihan teknologi difokuskan pada kecepatan pengembangan, kemudahan integrasi, serta stabilitas sistem. Adapun lingkungan kerja yang digunakan adalah sebagai berikut:

1. *Platform Backend*: Pengembangan layanan *server* memanfaatkan *Node.js* sebagai *runtime JavaScript* di sisi *backend*. *Framework Express.js* digunakan untuk menyusun arsitektur API berbasis REST, mengatur *routing request*, serta menangani *middleware* secara modular dan efisien.

2. Sistem Manajemen Basis Data: Proyek menggunakan *PostgreSQL* sebagai DBMS (*Database Management System*) relasional. *PostgreSQL* dipilih karena kemampuan pengelolaan *query* kompleks serta keandalan dalam menjaga konsistensi data transaksi koperasi.
3. ORM (*Object-Relational Mapping*): Untuk menjembatani interaksi antara kode *JavaScript* dengan tabel pada *PostgreSQL*, digunakan *Sequelize.js* sebagai ORM. Library ini mendukung operasi CRUD, pemodelan relasi tabel, serta fitur lanjutan seperti eksekusi ekspresi khusus melalui *Sequelize.literal*, yang digunakan dalam penyesuaian logika simpan pinjam.
4. Alat *Debugging* dan Pengujian Backend: Pengujian *endpoint* dilakukan menggunakan *Visual Studio Code REST Client* melalui file berekstensi .rest, yang memungkinkan pengiriman permintaan HTTP langsung ke server pengembangan lokal (localhost:5000). Selain itu, inspeksi basis data dilakukan menggunakan *pgAdmin/Postgres Tools* untuk memastikan data tersimpan dan terperbarui sesuai logika bisnis.

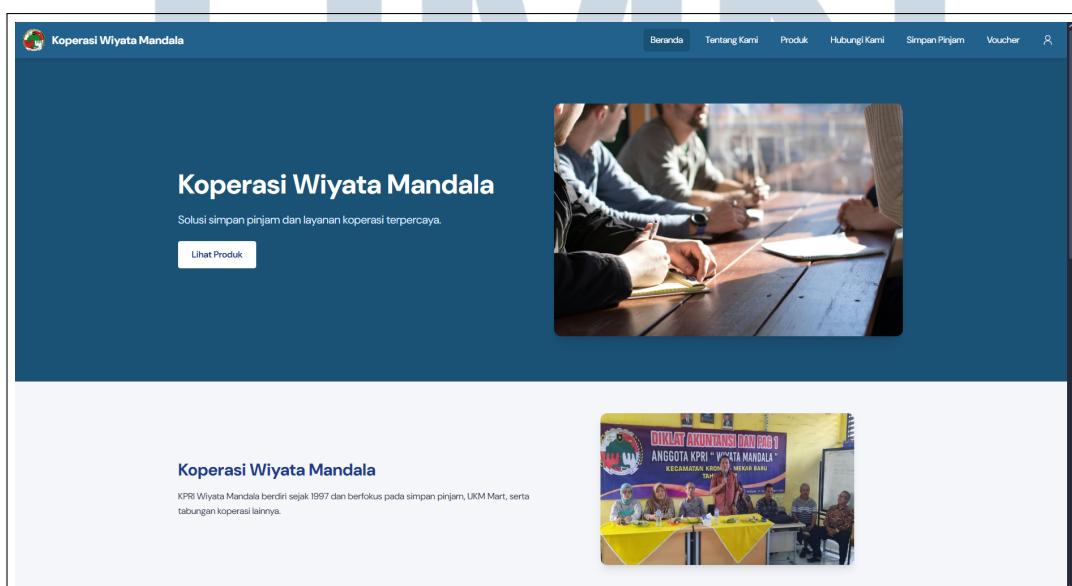
### 3.4 Sitemap Website Koperasi

Untuk mengakses *website* Koperasi Simpan Pinjam Wiyata Mandala, anggota dan pengurus akan diarahkan menuju tampilan registrasi dan *login* untuk masuk ke akun *user website* koperasi. Tampilan *register user* dapat dilihat pada Gambar 3.1 berikut.



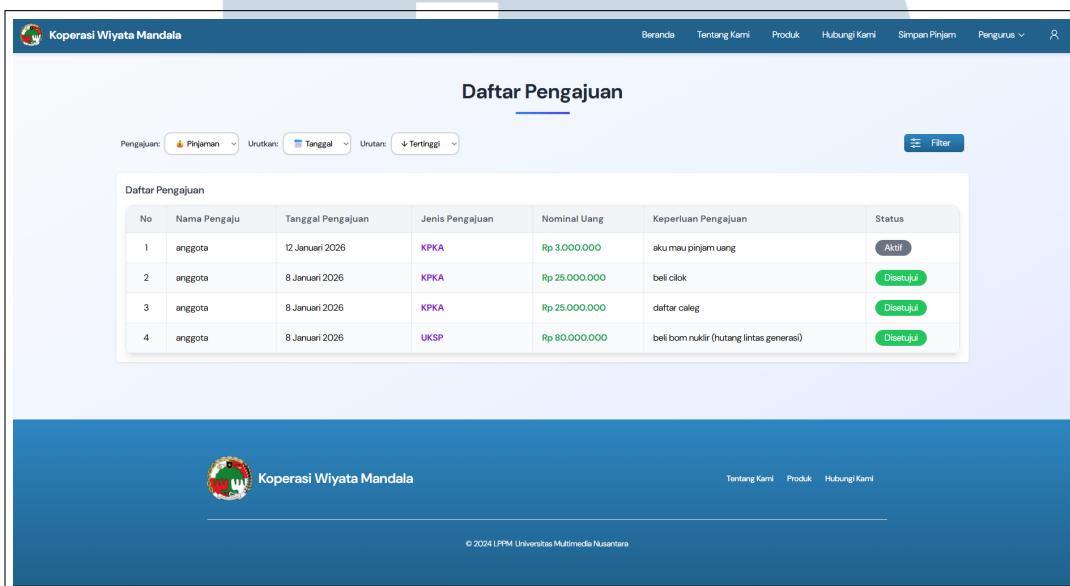
Gambar 3.1. Halaman *register user* koperasi

Setelah proses autentikasi berhasil, *user* yang memilih menu simpan pinjam akan diarahkan ke halaman utama fitur koperasi simpan pinjam. Pada halaman ini ditampilkan berbagai informasi ringkas, antara lain total dana simpanan koperasi, serta besaran potongan bulanan anggota. Bagi *user* dengan peran "Anggota", sistem menyediakan empat menu utama, yaitu "Ajukan Pinjaman", "Ajukan Simpanan", "Ajukan Penarikan", dan "Lihat Pengajuan". Sementara itu, *user* dengan peran "Pengurus" hanya diberikan akses ke menu "Lihat Pengajuan".



Gambar 3.2. Halaman utama *website* koperasi

Menu "Ajukan Pinjaman" mengarahkan *user* ke halaman formulir pengajuan pinjaman untuk mengisi dan mengirimkan data permohonan. Fungsi serupa diterapkan pada menu "Ajukan Simpanan", yang membawa *user* ke halaman pengajuan simpanan. Menu "Ajukan Penarikan" digunakan untuk mengakses formulir penarikan simpanan sesuai dengan ketentuan yang berlaku. Adapun menu "Lihat Pengajuan" menampilkan daftar seluruh pengajuan simpanan dan pinjaman dalam bentuk tabel yang bersumber dari *database* sistem.



The screenshot shows a web application interface for managing loan applications. At the top, there is a navigation bar with links for Beranda, Tentang Kami, Produk, Hubungi Kami, Simpan Pinjam, Pengurus, and a search icon. Below the navigation bar, the title 'Daftar Pengajuan' is centered above a table. The table has columns: No, Nama Pengaju, Tanggal Pengajuan, Jenis Pengajuan, Nominal Uang, Keperluan Pengajuan, and Status. There are four rows of data:

No	Nama Pengaju	Tanggal Pengajuan	Jenis Pengajuan	Nominal Uang	Keperluan Pengajuan	Status
1	anggota	12 Januari 2026	KPKA	Rp 3.000.000	aku mau pinjam uang	Aktif
2	anggota	8 Januari 2026	KPKA	Rp 25.000.000	beltiok	Disediakan
3	anggota	8 Januari 2026	KPKA	Rp 25.000.000	daftar caleg	Disediakan
4	anggota	8 Januari 2026	UKSP	Rp 80.000.000	beli bom nukir (hutang lintas generasi)	Disediakan

Below the table, there is a footer section with the logo of Koperasi Wiyata Mandala, links for Tentang Kami, Produk, and Hubungi Kami, and a copyright notice: © 2024 UPM Universitas Multimedia Nusantara.

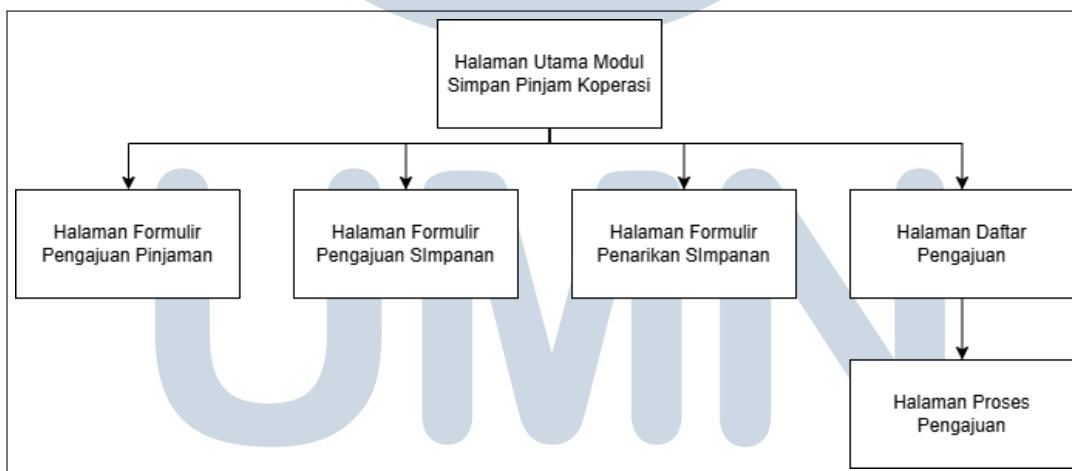
Gambar 3.3. Halaman daftar pengajuan pinjaman anggota

Setiap baris data pengajuan pada tabel tersebut dapat dipilih untuk membuka halaman detail proses pengajuan. Halaman ini menyajikan informasi status pengajuan, data identitas pemohon, serta rincian formulir yang telah diajukan. Khusus bagi *user* dengan peran "Pengurus", tersedia tombol "Setuju" dan "Tolak" yang digunakan untuk memberikan keputusan terhadap pengajuan yang masuk.

**U**  
**N**  
**I**  
**V**  
**E**  
**R**  
**S**  
**I**  
**T**  
**A**

Gambar 3.4. Halaman persetujuan pengajuan pinjaman anggota

Alur navigasi *sitemap* keseluruhan sistem koperasi ini ditunjukkan pada Gambar 3.5.



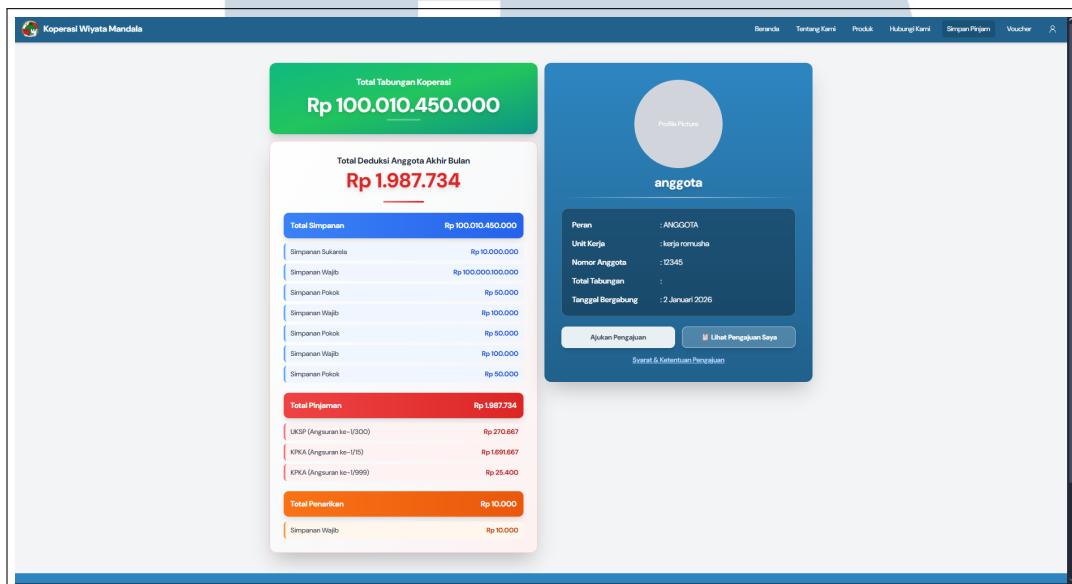
Gambar 3.5. Halaman *sitemap* sistem koperasi

### 3.5 Modul Halaman Awal Simpan Pinjam

*File HalamanAwalSimpanPinjam.jsx* merupakan komponen *frontend* berbasis *React* yang berfungsi sebagai halaman utama (*dashboard*) pada modul "Simpan Pinjam" sistem informasi Koperasi Simpan Pinjam Wiyata Mandala. Halaman ini menampilkan ringkasan data simpanan, pinjaman, penarikan, serta

deduksi bulanan yang sedang aktif, baik untuk anggota maupun pengurus koperasi, dengan penyesuaian hak akses.

Komponen ini memanfaatkan *React Hooks* seperti `useState` dan `useEffect` untuk pengelolaan *state* dan pengambilan data secara asinkron, *Axios* untuk komunikasi dengan *backend API*, serta *React Router* untuk navigasi antar halaman. Data yang ditampilkan diperoleh dari beberapa *endpoint backend* yang menyediakan informasi pengajuan aktif dan perhitungan deduksi.

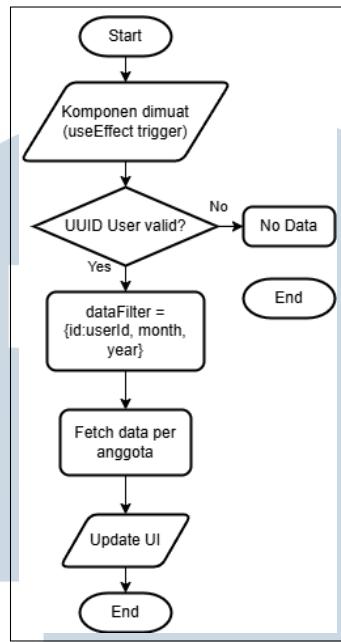


Gambar 3.6. Tampilan halaman awal simpan pinjam

Pada periode magang, penulis melakukan beberapa perbaikan dan penyesuaian logika tampilan serta pengambilan data, khususnya untuk kasus pengguna dengan peran pengurus koperasi. Fokus utama pekerjaan pada *file* ini adalah penyesuaian perhitungan dan tampilan deduksi bulanan agar sesuai dengan kebutuhan operasional koperasi.

### 3.5.1 Penyesuaian Logika Deduksi Bulanan

Pada implementasi sebelumnya, deduksi bulanan masih menjumlahkan seluruh transaksi termasuk pinjaman dan penarikan, yang seharusnya dihitung hanya simpanan wajib anggota. Logika deduksi bulanan yang belum diperbaiki dapat divisualisasikan dengan *flowchart* pada Gambar 3.7 berikut.



Gambar 3.7. Logika deduksi bulanan sebelum diperbaiki

## A Perubahan Komponen KeuanganAnggota

Komponen KeuanganAnggota memiliki kesalahan pada perhitungan deduksi bulanan yang ada. Komponen tersebut mengambil + dataPinjaman.TOTAL\_ANGSURAN dan dataPenarikan.TOTAL\_PENARIKAN ke dalam perhitungan total deduksi bulanan, yang seharusnya nilai tersebut tidak dihitung ke dalam perhitungan total.

```

1 <h1 className="text-5xl font-bold text-red-800 mb-4 text-center">
2   {"Rp " + formatRupiah(String(dataSimpanan.TOTAL_SIMPANAN +
3   dataPinjaman.TOTAL_ANGSURAN + dataPenarikan.TOTAL_PENARIKAN)) }

```

Kode 3.1: Kode komponen keuangan anggota terdahulu

Diubah menjadi:

```

1 <h1 className="text-5xl font-bold text-red-800 mb-4 text-center">
2   {"Rp " + formatRupiah(String(dataSimpanan.TOTAL_SIMPANAN)) }
3 </h1>

```

Kode 3.2: Kode komponen keuangan anggota yang diperbarui

Menghapus + dataPinjaman.TOTAL\_ANGSURAN dan dataPenarikan.TOTAL\_PENARIKAN sehingga total deduksi hanya menghitung simpanan.

## B Perubahan Komponen KeuanganKoperasi

Serupa dengan komponen KeuanganAnggota, komponen KeuanganKoperasi memiliki kesalahan pada perhitungan deduksi bulanan yang ada. Komponen tersebut mengambil + dataPinjaman.TOTAL\_ANGSURAN dan dataPenarikan.TOTAL\_PENARIKAN ke dalam perhitungan total deduksi bulanan, yang seharusnya nilai tersebut tidak dihitung ke dalam perhitungan total.

```
1 <h1 className="text-5xl font-bold text-red-800 mb-4 text-center">
2   {"Rp " + formatRupiah(String(dataSimpanan.TOTAL_SIMPANAN +
3     dataPinjaman.TOTAL_ANGSURAN + dataPenarikan.TOTAL_PENARIKAN))}</h1>
```

Kode 3.3: Kode komponen keuangan koperasi terdahulu

Diubah menjadi:

```
1 <h1 className="text-5xl font-bold text-red-800 mb-4 text-center">
2   {"Rp " + formatRupiah(String(dataSimpanan.TOTAL_SIMPANAN))}</h1>
```

Kode 3.4: Kode komponen keuangan koperasi yang diperbarui

Serupa dengan perubahan 1, menghapus + dataPinjaman.TOTAL\_ANGSURAN dan dataPenarikan.TOTAL\_PENARIKAN dari total keuangan koperasi.

## C Integrasi Deduksi Pengurus dari Seluruh Anggota

Pada iterasi sebelumnya, deduksi bulanan koperasi pada tampilan pengurus masih melakukan *filter* informasi dari *user* tertentu. Hal tersebut tidak sesuai dengan kapabilitas yang diinginkan. Pada tampilan pengurus, deduksi bulanan harus mengambil data deduksi bulanan dari seluruh anggota untuk melihat total deduksi bulanan koperasi.

```
1 const KeuanganKoperasi = ({ userData }) => {
2   // ... state declarations ...
3   useEffect(() => {
4     if (!isNaN(userData?.UUID_MS_USER)) {
5       const fetchDataType = async () => {
6         try {
7           const now = new Date();
8           const month = now.getMonth() + 1;
9           const year = now.getFullYear();
10          const dataFilter = {
11            "id": userData?.UUID_MS_USER,
```

```

12     "month": month,
13     "year": year
14   }
15   const dataPinjaman = await axios.post('http://localhost
16 :5000/getActivePengajuanPinjamanAnggota', dataFilter);
17   const dataSimpanan = await axios.post('http://localhost
18 :5000/getActivePengajuanSimpananAnggota', dataFilter);
19   const dataPenarikan = await axios.post('http://localhost
20 :5000/getActivePengajuanPenarikanAnggota', dataFilter);
21   // ...
22   }
23 }, [userData])

```

Kode 3.5: Kode deduksi pengurus terdahulu

Diubah manjadi:

```

1 const KeuanganKoperasi = ({ userData }) => {
2   // ... state declarations ...
3   useEffect(() => {
4     const fetchDataType = async () => {
5       try {
6         const now = new Date();
7         const month = now.getMonth() + 1;
8         const year = now.getFullYear();
9         const dataFilter = {
10           "month": month,
11           "year": year
12         }
13         const dataPinjaman = await axios.post('http://localhost
14 :5000/getActivePengajuanPinjamanSemuaAnggota', dataFilter);
15         const dataSimpanan = await axios.post('http://localhost
16 :5000/getActivePengajuanSimpananSemuaAnggota', dataFilter);
17         const dataPenarikan = await axios.post('http://localhost
18 :5000/getActivePengajuanPenarikanSemuaAnggota', dataFilter);
19         // ...
20       }
21     };
22     fetchDataType();
23   }, [])

```

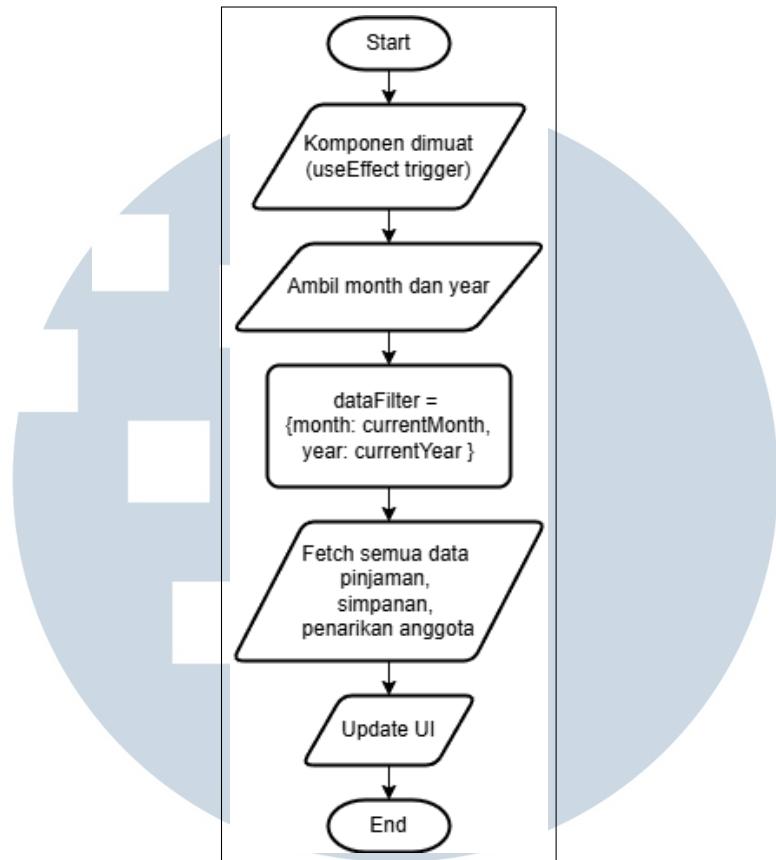
Kode 3.6: Kode deduksi pengurus yang diperbarui

Perubahan yang dilakukan:

- Menghapus kondisi `if (!isNaN(userData?.UUID_MS_USER))` karena pengurus tidak perlu *filter* per *user*
- Menghapus `"id": userData?.UUID_MS_USER` dari `dataFilter` karena mengambil data semua anggota
- Mengubah *endpoint* dari:
  - `getActivePengajuanPinjamanAnggota` →  
`getActivePengajuanPinjamanSemuaAnggota`
  - `getActivePengajuanSimpananAnggota` →  
`getActivePengajuanSimpananSemuaAnggota`
  - `getActivePengajuanPenarikanAnggota` →  
`getActivePengajuanPenarikanSemuaAnggota`
- Mengubah *dependency array* dari `[userData]` menjadi `[]` karena tidak bergantung pada `userData`

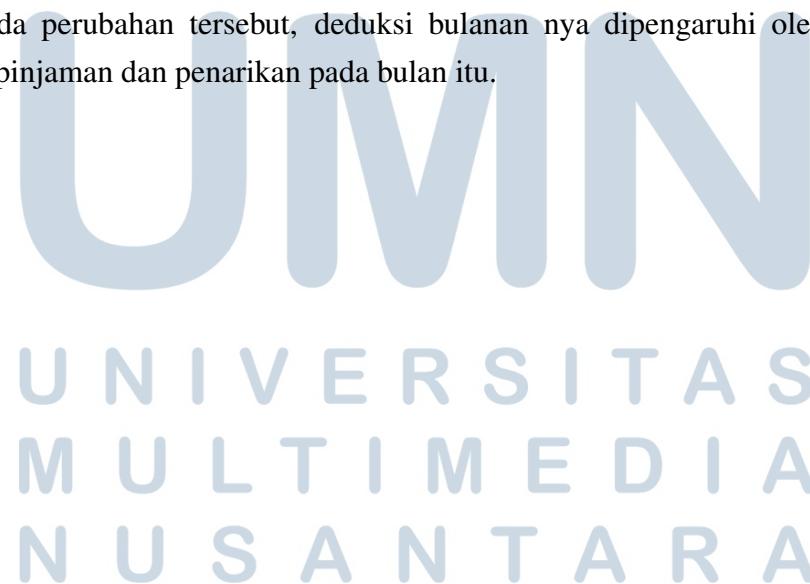
Berdasarkan perubahan tersebut, logika deduksi bulanan dapat ditampilkan dengan *flowchart* pada Gambar 3.8 berikut.





Gambar 3.8. Logika deduksi bulanan setelah diperbaiki

Pada perubahan tersebut, deduksi bulanan nya dipengaruhi oleh jumlah angsuran pinjaman dan penarikan pada bulan itu.





Gambar 3.9. Tampilan deduksi bulanan yang telah diperbaiki

### 3.5.2 Penyesuaian Logika Keterangan Angsuran

Sebelumnya, sistem belum menyediakan informasi posisi cicilan (angsuran keberapa dari total tenor), sehingga anggota maupun pengurus tidak dapat mengetahui progres pelunasan sebuah pinjaman langsung melalui tampilan *dashboard*. Untuk meningkatkan transparansi informasi serta mendukung proses *monitoring*, dilakukan penyesuaian pada komponen tampilan agar menampilkan status angsuran secara lebih detail.

## A Menambahkan Keterangan Angsuran pada KeuanganAnggota

Untuk menambahkan informasi keterangan angsuran pada KeuanganAnggota pada tampilan anggota, diimplementasikan perubahan kode berikut.

```
1 <div className="flex justify-between">
2   <p className="text-lg">{item.TYPE_NAME}</p>
3   <p className="text-lg">"Rp " + formatRupiah(String(item.
4     ANGSURAN))</p>
5 </div>
```

Kode 3.7: Kode keterangan angsuran keuangan anggota terdahulu

Diubah menjadi:

```
1 <div className="flex justify-between">
2   <p className="text-lg">{item.TYPE_NAME} (Angsuran ke-{item.
3     CURRENT_INSTALLMENT}/{item.TENOR})</p>
4   <p className="text-lg">"Rp " + formatRupiah(String(item.
5     ANGSURAN))</p>
6 </div>
```

Kode 3.8: Kode keterangan angsuran keuangan anggota yang diperbarui

Menambahkan informasi (Angsuran ke-n/total) pada nama tipe pinjaman untuk anggota.

## B Menambahkan Keterangan Angsuran pada KeuanganKoperasi

Informasi keterangan angsuran pada KeuanganKoperasi juga diimplementasikan untuk kejelasan pembayaran angsuran seluruh koperasi. Oleh karena itu, diimplementasikan perubahan kode berikut:

```
1 <div className="flex justify-between">
2   <p className="text-lg">{item.TYPE_NAME}</p>
3   <p className="text-lg">"Rp " + formatRupiah(String(item.
4     ANGSURAN))</p>
5 </div>
```

Kode 3.9: Kode keterangan angsuran keuangan koperasi terdahulu

Diubah menjadi:

```
1 <div className="flex justify-between">
2   <p className="text-lg">{item.TYPE_NAME} (Angsuran ke-{item.
3     CURRENT_INSTALLMENT}/{item.TENOR})</p>
```

```
3 <p className="text-lg">{"Rp " + formatRupiah(String(item.  
ANGSURAN))}</p>  
4 </div>
```

Kode 3.10: Kode keterangan angsuran keuangan koperasi yang diperbarui

Menambahkan informasi (Angsuran ke-n/total) pada nama tipe pinjaman untuk pengurus koperasi.

Perubahan kode tersebut berhasil menambahkan tampilan informasi "angsuran ke-" pada tampilan angsuran. Bentuk UI "angsuran ke-" tersebut dapat dilihat di Gambar 3.10.



Gambar 3.10. Tampilan informasi "angsuran ke-" pada tampilan pengurus dan anggota

### 3.6 Modul Form Pengajuan Pinjaman

File `FormPengajuanPinjaman.jsx` merupakan komponen *frontend* berbasis *React* yang berfungsi sebagai formulir pengajuan pinjaman anggota koperasi pada Sistem Informasi Koperasi Simpan Pinjam Wiyata Mandala. Komponen ini digunakan oleh anggota untuk mengajukan pinjaman dengan memilih jenis pinjaman, nominal, tenor, serta informasi pendukung lainnya sebelum data dikirim ke *backend* untuk diproses.

Komponen ini memanfaatkan *React Hooks* (`useState`, `useEffect`) untuk pengelolaan *state form*, *Axios* untuk pengiriman data ke *backend API*, serta validasi sisi *frontend* untuk memastikan data yang dikirim sesuai dengan ketentuan koperasi sebelum disimpan ke basis data.

Gambar 3.11. Tampilan halaman *form* pengajuan pinjaman

### 3.6.1 Penyesuaian Handler Input Tenor

Pada implementasi sebelumnya, nilai tenor pada *form* pengajuan pinjaman bersifat tetap (*read-only*) dan tidak dapat diubah oleh pengguna. Hal ini membatasi fleksibilitas pengajuan pinjaman sesuai kebijakan koperasi.

#### A Menambahkan Handler untuk Input Tenor di handleChange

*Input* tenor memerlukan *handler* pada kode (*handleChange*) di (*FormPengajuanPinjaman*) untuk dapat beroperasi. Berikut perubahan kode yang dilakukan.

```

1 const handlers = {
2   nominalPinjaman: () => {
3     const numericValue = value.replace(/\D/g, '');
4     // Only accepts
5     // digit 0-9 inputs
6     return formatRupiah(numericValue);
7   },
8   checkbox: () => checked,
9   default: () => value,
10 };

```

Kode 3.11: Kode handler input tenor terdahulu

Diubah menjadi:

```

1 const handlers = {
2   nominalPinjaman: () => {
3     const numericValue = value.replace(/\D/g, '');
4     // Only accepts
5     // digit 0-9 inputs
6     return formatRupiah(numericValue);
7   },
8   tenor: () => {
9     const numericValue = value.replace(/\D/g, '');
10    // Only accepts
11    // digit 0-9 inputs
12    return numericValue;
13  },
14  checkbox: () => checked,
15  default: () => value,
16};

```

Kode 3.12: Kode handler input tenor yang diperbarui

Menambahkan *handler* khusus untuk *field* 'tenor' yang hanya menerima *input* angka (0-9).

## B Mengubah Input Tenor dari ReadOnly menjadi Editable

Selain membutuhkan *handler*, *input* tenor juga memerlukan perubahan pada *field input* yang sebelumnya *ReadOnly* menjadi *Editable*. Berikut perubahan kode yang mengimplementasikan perubahan tersebut.

```

1 <div className='flex flex-col justify-end'>
2   <label className="block mb-1 font-medium">Tenor (Bulan)</label>
3   <input
4     type="text"
5     name="tenor"
6     value={formData.tenor}
7     onChange={handleChange}
8     className="w-full p-2 border rounded bg-gray-100 cursor-not-
9     allowed"
10    readOnly
11    required
12  />

```

Kode 3.13: Kode input tenor terdahulu

Diubah menjadi:

```

1 <div className='flex flex-col justify-end'>
2   <label className="block mb-1 font-medium">Tenor (Bulan)</label>

```

```

3 <input
4   type="text"
5   name="tenor"
6   value={formData.tenor}
7   onChange={handleChange}
8   placeholder="Masukkan tenor dalam bulan"
9   className="w-full p-2 border rounded bg-white focus:outline-
10 none focus:ring focus:ring-blue-300 shadow-md"
11 required
12 />

```

Kode 3.14: Kode input tenor yang diperbarui

Perubahan yang dilakukan:

- Menghapus atribut *readOnly*
- Mengubah *background* dari *bg-gray-100* menjadi *bg-white*
- Menghapus *cursor-not-allowed*
- Menambahkan *placeholder*
- Menambahkan *focus styling* (*ring* biru saat diklik)

Setelah perubahan ini, *user* dapat memasukkan tenor secara *manual*, dan *input* hanya akan menerima angka saja.

The screenshot shows a user interface for a loan application. At the top, there are two dropdown menus: 'Tipe Pinjaman' (selected 'KPKA (Kredit Pelayanan Khusus Anggota)') and 'Tenor (Bulan)' (selected '99999'). Below these, there are two input fields: 'Nominal Uang' (value '0') and 'Deduksi Bulanan' (value '0'). A note '(Bunga: 1.5%)' is shown next to the deduksi field. The background features a watermark with the text 'MULTIMEDIA NUSANTARA'.

Gambar 3.12. Tenor peminjaman yang sudah diperbaiki

### 3.6.2 Penyesuaian Limit Pinjaman KPKA

Pada implementasi sebelumnya, pengajuan KPKA tidak diberi *limit* secara pasti jumlah total pinjaman yang diberikan. Pihak koperasi memberikan *limit*

peminjaman KPKA berupa Rp. 25.000.000. Berikut perubahan kode yang mengimplementasikan perubahan tersebut.

```
1 const handleSelectChange = (e) => {
2   const selectedValue = e.target.value;
3   const selectedOption = typePinjaman?.length > 0 ? typePinjaman.
4     find(item => item.TYPE_NAME === selectedValue) : null;
5
6   setFormData((prevData) => ({
7     ...prevData,
8     typePinjaman: selectedValue,
9     typePinjamanDesc: selectedOption?.TYPE_DESC ?? prevData.
10    typePinjamanDesc,
11    tenor: selectedOption?.TENOR ?? prevData.tenor,
12    minimalPinjaman: selectedOption ? formatRupiah(selectedOption.
13      MINIMUM_PINJAMAN) : prevData.minimalPinjaman,
14    maksimalPinjaman: selectedOption ? formatRupiah(selectedOption
15      .MAXIMUM_PINJAMAN) : prevData.maksimalPinjaman,
16    bunga: selectedOption?.INTEREST_RATE ?? prevData.bunga,
17    typePinjamanID: selectedOption?.UUID_TYPE_PINJAMAN ?? prevData
18      .typePinjamanID,
19    }));
20  });
21};
```

Kode 3.15: Kode limit KPKA terdahulu

Diubah manjadi:

```
1 const handleSelectChange = (e) => {
2   const selectedValue = e.target.value;
3   const selectedOption = typePinjaman?.length > 0 ? typePinjaman.
4     find(item => item.TYPE_NAME === selectedValue) : null;
5
6   // Override maksimal pinjaman untuk KPKA menjadi 25 juta
7   const maksimalPinjaman = selectedValue === 'KPKA' ? '25000000' :
8     selectedOption?.MAXIMUM_PINJAMAN;
9
10  setFormData((prevData) => ({
11    ...prevData,
12    typePinjaman: selectedValue,
13    typePinjamanDesc: selectedOption?.TYPE_DESC ?? prevData.
14    typePinjamanDesc,
15    tenor: selectedOption?.TENOR ?? prevData.tenor,
16    minimalPinjaman: selectedOption ? formatRupiah(selectedOption.
17      MINIMUM_PINJAMAN) : prevData.minimalPinjaman,
```

```

14     maksimalPinjaman: selectedOption ? formatRupiah(
15         maksimalPinjaman) : prevData.maksimalPinjaman ,
16     bunga: selectedOption?.INTEREST RATE ?? prevData.bunga ,
17     typePinjamanID: selectedOption?.UUID_TYPE_PINJAMAN ?? prevData
18     .typePinjamanID ,
19   }));
20 };

```

Kode 3.16: Kode limit KPKA yang diperbarui

Penyesuaian logika dilakukan dengan menambahkan kondisi untuk mengecek apakah tipe pinjaman yang dipilih adalah KPKA. Apabila tipe pinjaman tersebut terpilih, maka nilai maksimal pengajuan pinjaman akan di *limit* menjadi sebesar Rp. 25.000.000. Sebaliknya, jika tipe pinjaman yang dipilih bukan KPKA, sistem tetap menggunakan nilai (MAXIMUM\_PINJAMAN) yang diambil dari *database* sebagaimana sebelumnya. Nilai batas maksimal ini ditampilkan secara langsung pada UI pengguna dan digunakan sebagai acuan validasi saat proses pengajuan pinjaman, sehingga memastikan bahwa nominal yang diajukan sesuai dengan ketentuan yang berlaku.

Gambar 3.13. *Limit* peminjaman KPKA yang telah diperbarui

### 3.7 Kendala dan Solusi yang Ditemukan

#### 3.7.1 Kendala yang dihadapi

Selama melakukan pengembangan pada modul peminjaman, penulis menghadapi sejumlah tantangan teknis dan non-teknis yang berhubungan dengan pemahaman sistem dan kejelasan spesifikasi pekerjaan. Kendala tersebut mencakup:

1. Kompleksitas Arsitektur Sistem dan Minimnya Dokumentasi: Sistem telah melalui beberapa iterasi oleh tim sebelumnya, namun belum dilengkapi dokumentasi teknis maupun komentar kode secara memadai. Hal ini menyulitkan penulis dalam memahami dependensi antar fungsi, struktur model data, serta alur bisnis yang berjalan pada backend.
2. Deskripsi Pekerjaan Kurang Spesifik: Jobdesk yang diberikan pada awal periode pengembangan masih bersifat umum, sehingga diperlukan konfirmasi tambahan untuk memahami batasan pekerjaan serta arah penyelesaian yang sesuai kebutuhan koperasi.
3. Kode *Boilerplate* dan Struktur Fungsi yang Tidak Seragam: Beberapa bagian kode ditulis hanya sebagai *placeholder* dari tim sebelumnya, menyebabkan ketidakkonsistenan implementasi dan menyulitkan identifikasi fungsi yang benar-benar aktif dipakai oleh sistem.

### 3.7.2 Solusi yang diterapkan

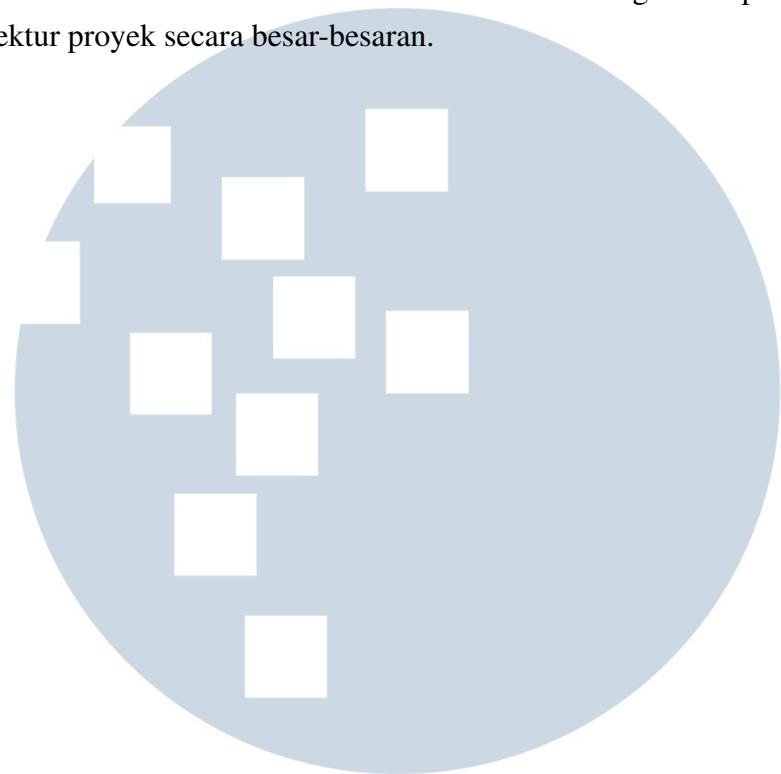
Untuk menghadapi hambatan tersebut, langkah-langkah penyelesaian berikut dilakukan:

1. Eksplorasi Kode Mendalam (*Codebase Deep Dive*) Tim proyek melakukan penelusuran struktur proyek secara menyeluruh, mencakup:
  - Pemetaan *controller* dan *route* yang aktif
  - Identifikasi relasi *tabel* pada *database*
  - Melakukan *manual testing* melalui *REST Client*

Pendekatan ini memakan waktu, tetapi berhasil memberikan pemahaman menyeluruh terhadap alur bisnis sistem.

2. Konsultasi Teknis Secara Berkala dengan Pembimbing: Penulis secara aktif meminta klarifikasi mengenai tujuan pengembangan modul, prioritas kebutuhan koperasi, dan fungsi mana yang benar-benar dibutuhkan, sehingga penggeraan dapat tetap berjalan sesuai arah yang diharapkan.
3. *Refactor Minor* dan Validasi Fungsi: Penulis melakukan pengujian pada fungsi fungsi *boilerplate* untuk menentukan fungsi yang perlu dipertahankan

dan dikembangkan lebih lanjut dan fungsi yang dapat diabaikan karena tidak terpakai. Hal ini memastikan kualitas kode meningkat tanpa mengubah arsitektur proyek secara besar-besaran.



**UMN**  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA