

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Kegiatan magang dilaksanakan di Departemen *IT Solution* dengan jabatan sebagai *IT Front End Developer Intern*. Bimbingan teknis disampaikan secara langsung oleh Bapak Fathariq selaku Mentor, yang bertanggung jawab memberikan pengarahan dan menjaga standar kualitas teknis setiap pekerjaan, serta memastikan proyek sesuai dengan standar perusahaan. Pemberian tugas diatur oleh Bapak Berlianto sebagai *Business Analyst* dan Bapak Suryadi yang memegang peran sebagai *Senior IT Front End Developer* sekaligus Supervisor, yang turut menilai kinerja magang.

Koordinasi pekerjaan dilakukan baik secara tatap muka maupun secara daring menggunakan platform Google Meet. Proses pengembangan dilakukan secara bertahap dalam beberapa *sprint* dengan pemantauan yang difasilitasi oleh sistem manajemen proyek Jira. Untuk memastikan kelancaran proses dan efektivitas koordinasi tim, pertemuan *daily stand-up* diselenggarakan setiap seminggu sekali. Proyek melibatkan kerja sama lintas peran, termasuk *UI/UX Designer*, *Back End Developer*, *Front End Developer*, dan *Quality Assurance*, di bawah pengawasan seorang *Business Analyst*.

3.2 Tugas yang Dilakukan

Program magang di PT Dana Kini Indonesia pada posisi *IT Front End Developer Intern* melibatkan pelaksanaan berbagai tugas, antara lain:

1. *Upgrade libraries* dan *plugin* yang sudah *deprecated* pada *website* operasional

Memastikan sistem tetap berjalan dengan stabil dan kompatibel terhadap versi terbaru dari *library* yang digunakan, seperti jQuery, Bootstrap, serta berbagai *plugin* pendukung lainnya. Proses pembaruan juga melibatkan penyesuaian terhadap perubahan sintaks dan dependensi agar tampilan serta fungsionalitas *website* tetap optimal dan berjalan seperti sebelumnya.

2. Pengembangan modul Pembiayaan Usaha pada *website* operasional

Memfasilitasi proses pengajuan, verifikasi, dan persetujuan pembiayaan yang dilakukan oleh pengguna melalui *website* operasional. Pengembangan dilakukan berdasarkan desain hasil kerja *UI/UX Designer* agar antarmuka mudah digunakan, serta memastikan alur pengajuan berjalan terintegrasi dengan sistem *backend* perusahaan.

3. Pembuatan halaman Inquiry untuk penanganan berbagai produk pada *website* operasional

Halaman ini dirancang agar mampu menampilkan dan memproses data dari berbagai jenis produk pembiayaan yang dimiliki perusahaan. Dalam proses pengembangannya, dilakukan penyesuaian logika tampilan dan pemanggilan API berdasarkan kode produk, sehingga sistem dapat secara dinamis menampilkan data yang relevan sesuai dengan jenis produk yang dilakukan pengajuan oleh pengguna.

3.3 Uraian Pelaksanaan Magang

Gambaran mengenai pelaksanaan kerja magang ditampilkan pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Mengembangkan tampilan dan melakukan integrasi API halaman Inquiry pada <i>website</i> operasional.
2	Mengembangkan tampilan dan melakukan integrasi API Modul Produktivitas pada <i>website</i> operasional.
3	Melakukan <i>upgrade libraries</i> dan <i>plugin</i> yang sudah <i>deprecated</i> pada <i>website</i> operasional.
4	Melakukan penyesuaian tampilan pada halaman Inquiry bagian Search dan Info Profil.
5	Melakukan integrasi API untuk halaman Inquiry bagian Search dan Info Profil untuk menangani produk berbeda pada <i>website</i> operasional.
Lanjut pada halaman berikutnya	

Tabel 3.1 Pekerjaan yang dilakukan tiap minggu selama magang (lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
6	Melakukan integrasi API untuk halaman Inquiry bagian Aktivitas Sebagai Peminjam dan menyesuaikan tampilan pada <i>website</i> operasional sesuai <i>feedback</i> QA.
7	Melanjutkan <i>upgrade libraries</i> dan <i>plugin</i> yang sudah <i>deprecated</i> pada <i>website</i> operasional.
8	Melakukan penanganan produk untuk halaman Inquiry bagian Aktivitas Sebagai Peminjam untuk produk berbeda pada <i>website</i> operasional.
9	Mengikuti <i>project grooming</i> untuk modul Pembiayaan Usaha pada <i>website</i> operasional.
10	Melakukan integrasi API untuk halaman Data Pokok pada <i>website</i> operasional.
11	Mengembangkan tampilan halaman Data Pembiayaan pada <i>website</i> operasional.
12	Melakukan integrasi API untuk halaman Data Pembiayaan pada <i>website</i> operasional.
13	Mengembangkan tampilan halaman Approval Pembiayaan Usaha pada <i>website</i> operasional.
14	Melakukan integrasi API untuk halaman Approval Pembiayaan pada <i>website</i> operasional.
15	Melanjutkan <i>upgrade libraries</i> dan <i>plugin</i> yang sudah <i>deprecated</i> pada <i>website</i> operasional.
16	Melakukan <i>refactor</i> hasil <i>remarks</i> .

Pada minggu pertama, dilakukan pengembangan tampilan halaman Inquiry bagian Sejarah Pengajuan serta pengembangan halaman Komite Kredit yang disertai dengan proses integrasi *API*. Pada minggu kedua, kegiatan difokuskan pada *refactor* halaman Inquiry, sekaligus pengembangan antarmuka dan integrasi *API* pada modul Produktivitas. Selanjutnya, pada minggu ketiga hingga minggu kedelapan, dilakukan penyesuaian tampilan dan penyesuaian *API* pada seluruh *tab* di halaman Inquiry untuk dapat menangani kondisi dan alur produk yang berbeda.

Pada minggu kesembilan, dilaksanakan kegiatan *project grooming* untuk membahas rancangan modul Pembiayaan Usaha pada *website* operasional perusahaan, sekaligus memulai pengembangan tampilan halaman Data Pokok. Proses integrasi API untuk halaman tersebut dilakukan pada minggu berikutnya. Memasuki minggu kesebelas, dilakukan pengembangan tampilan halaman *Data Pembiayaan*, yang kemudian diikuti dengan integrasi API pada minggu selanjutnya. Pada minggu ketigabelas, dilakukan pembuatan halaman Approval Pembiayaan Usaha, dan integrasi API untuk halaman tersebut dilaksanakan pada minggu setelahnya.

Pada minggu kelimabelas, dilakukan *refactor* terhadap hasil *remarks* setelah *User Acceptance Test* (UAT) untuk memastikan kualitas dan konsistensi sistem secara keseluruhan. Pada minggu keenambelas, kegiatan difokuskan pada melakukan *upgrade* terhadap *libraries* dan *plugin* yang sudah *deprecated* pada *website* operasional perusahaan. Tugas tersebut bersifat tugas tambahan yang dikerjakan apabila tidak terdapat pekerjaan utama yang sedang berlangsung.

3.3.1 Pengembangan Modul Inquiry dan Integrasi API

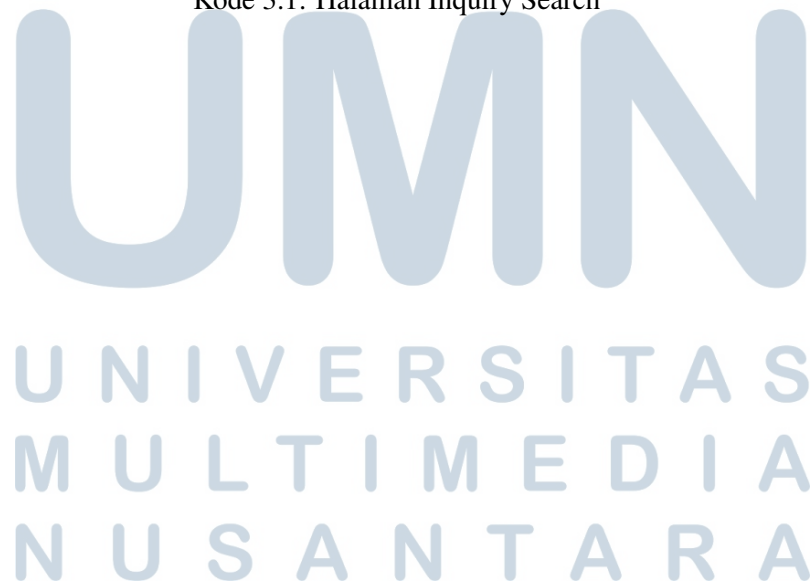
Pengembangan modul Inquiry menjadi fokus utama pada awal pengerjaan, khususnya pada bagian Search, Info Profil, dan Aktivitas Sebagai Peminjam. Setiap bagian disesuaikan agar dapat menangani produk yang berbeda sesuai kebutuhan bisnis, sehingga sistem dapat menampilkan informasi yang relevan bagi setiap tipe pengguna. Proses ini juga melibatkan penyesuaian tampilan agar antarmuka tetap konsisten dan mudah digunakan. Kegiatan ini merupakan kelanjutan dari pengembangan yang dilakukan pada magang periode 1, sehingga banyak modul dasar telah tersedia sebagai fondasi untuk implementasi lanjutan.

Selain itu, dilakukan integrasi API untuk menampilkan data secara dinamis dari *server*, sehingga setiap informasi pada *tab* dapat selalu diperbarui secara *real-time*. Integrasi ini membantu menjaga akurasi dan konsistensi data antar-tahap proses, serta mempermudah pengembangan fitur lanjutan di masa mendatang. Pengujian dilakukan secara berkala untuk memastikan API dapat berjalan dengan baik dan tidak menimbulkan gangguan pada tampilan maupun fungsionalitas.

Dari berbagai *tab* yang tersedia, salah satu yang paling krusial adalah fitur Search, yang memungkinkan pengguna menemukan informasi debitur dengan cepat dan efisien. Kode 3.1 memperlihatkan implementasi integrasi API untuk fitur Search pada modul Inquiry. Kode tersebut menangani aksi klik pada tombol detail di tabel Inquiry, kemudian mengambil data identitas debitur dan kode produk untuk membentuk parameter permintaan. Parameter tersebut diencode dan digunakan dalam penyusunan URL tujuan, sehingga sistem dapat mengarahkan pengguna ke halaman detail Inquiry yang sesuai berdasarkan data yang dipilih.

```
1 $( '#Inquiry_Table' ).on( 'click', 'button.detail', function () {  
2     const cif = this.id;  
3     const enval = btoa(cif);  
4     let prdCode = $(this).data( 'prd-code' );  
5     prdCode = prdCode ? btoa(prdCode) : null;  
6  
7     let url = "${local_server}/inquiry/${type}" + enval;  
8     if (prdCode) {  
9         url += "?prd-code=" + prdCode;  
10    }  
11  
12    window.location.href = url;  
13 });  
14
```

Kode 3.1: Halaman Inquiry Search



Gambar 3.1 menunjukkan hasil implementasi integrasi API untuk fitur Search, yang menampilkan data debitur secara *real-time*.

The screenshot shows a web application interface for 'Danakini'. The main section is titled 'Informasi Customer - Semua'. It contains several input fields for searching: CIF, Nama (with 'vien' entered), Nomor Handphone, Email/Surel, NIP, No Kontrak, KTP, and No Akun. A 'Search' button is located to the right of these fields. Below the search filters, there is a table displaying search results. The table has columns for CIF, Nama Lengkap, Alamat, Tanggal Lahir, Ponsel, Status, and Tindakan. Three entries are shown, all with a status of 'Active' and a green checkmark in the 'Tindakan' column.

CIF	Nama Lengkap	Alamat	Tanggal Lahir	Ponsel	Status	Tindakan
C1513200861162516	vien	jakarta	13 Agustus 2025	080000927272	Active	✓
C3231201055132543	vien	jakarta	31 Oktober 2004	080000001122	Active	✓
C3905200599102513	vien company	sriwss	7 Mei 2025	087810988807	Active	✓

Showing 1 to 3 of 3 entries

Gambar 3.1. Halaman Inquiry Search

3.3.2 Implementasi Antarmuka Modul Produktivitas dan Integrasi API

Modul Produktivitas dikembangkan dengan fokus pada tampilan yang responsif, agar setiap tugas dan informasi produktivitas dapat diakses dengan cepat dan mudah oleh pengguna. Desain antarmuka disesuaikan agar data dapat ditampilkan dengan jelas dan rapi, sehingga pengguna dapat memantau informasi produktivitas tanpa kesulitan. Setiap tampilan diuji untuk memastikan konsistensi dengan modul lain yang telah dikembangkan sebelumnya.

Integrasi API dilakukan agar data tugas dan produktivitas ditampilkan secara *real-time*. Pendekatan ini memastikan informasi yang disajikan selalu akurat dan terkini, tanpa perlu pembaruan manual. Dengan integrasi ini, modul dapat menampilkan data secara efisien, sekaligus memudahkan pemantauan progres pekerjaan oleh pengguna. Kode 3.2 menunjukkan bagaimana API dipanggil dan data dikembalikan ke *frontend* untuk ditampilkan dalam tabel secara responsif.

```

1 $('#productivity-tbl').DataTable({
2   ajax: {
3     url: '${local_server}/productivity/get-list',
4     type: 'GET',
5     dataType: 'json',

```

```

6      dataSrc: function(json){
7          return json.data.data_array;
8      }
9  },
10     responsive: true ,
11     columns: [ ... ] // Diisi key data dari JSON
12 });
13

```

Kode 3.2: Modul Produktivitas

Gambar 3.2 merupakan tampilan hasil implementasi modul Produktivitas, yang menunjukkan bagaimana data produktivitas ditampilkan secara rapi dan responsif pada halaman *website* operasional.

Produktivitas

ID Pengguna	Nama	Total Akun	Akun Yang Sudah Dicoba	Jumlah Tindakan	Terhubung	Persentase Terhubung	Jang Bayar	Persentase PTP	Akun Yang Membayar	Jumlah Pembayaran
USR001	Anonim A	10	8	20	15	75	5	62.5	3	1500000
USR002	Anonim B	12	9	22	18	82	7	70	4	2000000

Showing 1 to 2 of 2 entries

Average Time Monitoring

ID Pengguna	Nama	Jarak Antar Call (Average)	Jumlah Panggilan	Jarak Antar Tindakan	Jumlah Input Tindakan	Intensitas Input Tindakan	Tindakan
ATM001	Anonim X	30	12	45	20	2.34	23
ATM002	Anonim Y	40	10	50	18	2.11	23

Showing 1 to 2 of 2 entries

Gambar 3.2. Halaman Produktivitas

3.3.3 Upgrade Libraries dan Plugin Website Operasional

Selama pengembangan *website* operasional, dilakukan *upgrade libraries* dan *plugin* yang telah *deprecated* untuk menjaga keamanan, stabilitas, dan kompatibilitas seluruh modul yang terdapat pada sistem. *Upgrade* ini mencakup *libraries* yang digunakan di seluruh modul, dan setiap pembaruan diuji untuk memastikan tidak mengganggu fungsionalitas modul yang telah berjalan. Proses *upgrade* juga memungkinkan pemanfaatan fitur terbaru dari *libraries* dan *plugin* yang diperbarui, sehingga performa sistem meningkat.

Sebagai ilustrasi penerapan *upgrade libraries*, Kode 3.3 menunjukkan merupakan halaman Ubah Kata Sandi sebelum dilakukan *upgrade*, khususnya untuk *field* Kata Sandi Lama. *Upgrade* ini mencakup penyesuaian kelas CSS dan struktur HTML agar sesuai dengan *library* Bootstrap terbaru, serta pembaruan *library* ikon dari Font Awesome 4.7.0 menjadi Font Awesome 7.1.0. Pembaruan tersebut dilakukan untuk meningkatkan konsistensi tampilan antarmuka, kompatibilitas komponen, dan dukungan terhadap ikon-ikon modern yang lebih beragam.

```

1 <div class="form-group">
2   <label class="col-sm-4 control-label">Kata Sandi Lama</label>
3   <div class="col-sm-6 input-group">
4     <input type="password" name="old_pass"
5       id="old_pass" class="form-control" required />
6     <span class="input-group-addon fa fa-eye-slash"
7       id="toggleOldPassword" style="cursor: pointer;"></span>
8   </div>
9 </div>
10

```

Kode 3.3: Sebelum dilakukan *upgrade libraries*

Setelah dilakukan *upgrade*, struktur *field* ini diperbarui menjadi seperti pada Kode 3.4, dengan penyesuaian kelas agar lebih responsif, konsisten, dan sesuai dengan standar Bootstrap modern.

```

1 <div class="row mb-3">
2   <div class="col-sm-3 text-md-end">
3     <label for="old_pass" class="col-form-label">Kata Sandi
4     Lama</label>
5   </div>
6   <div class="col-sm-7">
7     <div class="input-group">
8       <input type="password" name="old_pass"
9         id="old_pass" class="form-control" required />
10      <span id="toggleOldPassword" class="input-group-text
11        fa-regular fa-eye-slash pt-2" style="cursor: pointer;"></span>
12    </div>
13  </div>
14 </div>

```

Kode 3.4: Setelah dilakukan *upgrade libraries*

3.3.4 Proses Implementasi Antarmuka Modul Pembiayaan Usaha

Pengerjaan modul dilakukan berdasarkan *Business Requirement Document* (BRD) yang merupakan dokumen internal perusahaan yang berisi spesifikasi bisnis, alur proses, serta spesifikasi fungsional yang harus diimplementasikan pada sistem. BRD tersebut menjadi acuan utama dalam proses pengembangan, dan dibahas melalui serangkaian *meeting* internal yang melibatkan seluruh anggota *squad*, yang terdiri dari *Business Analyst*, *Quality Assurance* (QA), *Backend Developer*, *Frontend Developer*, serta *UI/UX Designer* untuk memastikan kesamaan pemahaman terhadap kebutuhan sistem.

Setiap anggota *squad* memiliki tanggung jawab sesuai dengan perannya masing-masing. Tim *Business Analyst* bertugas menjelaskan kebutuhan bisnis dan proses yang tertuang dalam BRD, tim *UI/UX* menyusun rancangan tampilan antarmuka, sedangkan tim *Frontend* dan *Backend Developer* mengimplementasikan serta melakukan integrasi antar komponen sistem. Proses pengembangan dijalankan menggunakan pendekatan *Agile Scrum*, yang dibagi ke dalam empat *sprint* dengan durasi dua minggu untuk setiap *sprint*.

Selama setiap *sprint*, dilakukan koordinasi rutin berupa *daily meeting* dan *sprint review* untuk memantau perkembangan, mengevaluasi hasil implementasi, serta memastikan bahwa pengembangan modul telah sesuai dengan kebutuhan yang tertuang dalam BRD Pembiayaan Usaha [4], yang merupakan dokumen internal perusahaan dengan sifat rahasia. Pada *sprint* terakhir, dilakukan sesi *retrospective* untuk meninjau keseluruhan proses kerja, mengevaluasi kendala yang dihadapi, serta merumuskan perbaikan yang dapat diterapkan pada pengembangan modul di masa mendatang.

A Desain Figma Modul Pembiayaan Usaha

Bagian ini menampilkan rancangan antarmuka modul Pembiayaan Usaha yang dirancang oleh tim UI/UX Designer menggunakan platform Figma. Desain tersebut berperan sebagai acuan utama dalam proses implementasi antarmuka pada *website* operasional perusahaan agar hasil pengembangan selaras dengan kriteria visual dan interaksi pengguna yang telah ditetapkan. Detail rancangan antarmuka disajikan pada Lampiran 9, yang dikembangkan oleh Pak Elbert dan Ibu Graciell sebagai UI/UX Designer.

Berikut merupakan salah satu rancangan antarmuka yang dijadikan acuan dalam proses implementasi, yaitu halaman Data Pokok, yang diisi oleh tim marketing dan tim legal dengan data identitas perusahaan debitur. Halaman ini berisi tiga *tab*, yaitu *tab* Data Debitur – Marketing, *tab* Data Debitur – Legal, dan *tab* Data Pengurus/Pemilik. Pada *tab* Data Debitur – Marketing, pengguna dapat mengisi informasi dasar perusahaan seperti jenis badan hukum, nama debitur, NPWP, alamat, serta kontak yang dapat dihubungi. Gambar 3.3 menunjukkan desain tampilan *tab* Data Debitur - Marketing.

The screenshot shows a web application interface for 'Pembiayaan Usaha - Data Pokok'. At the top, there's a navigation bar with a hamburger menu on the left and a user profile 'ADMIN DANAKINI OPS DEV' on the right. Below the navigation bar, the main content area is titled 'Pembiayaan Usaha - Data Pokok'. It features three tabs: 'Data Debitur - Marketing' (active), 'Data Debitur - Legal', and 'Data Pengurus/Pemilik'. Under the 'Data Debitur - Marketing' tab, there's a section titled 'Identitas Perusahaan'. This section contains several form fields: 'Jenis Badan Hukum' (a dropdown menu with 'PT' selected), 'Nama Debitur' (text input with 'Informa Electronic'), 'Email Debitur' (text input with 'danakini@mail.com'), 'NPWP Debitur' (text input with '1111111111111111'), 'Nomor Telepon' (text input with '081234567890'), 'Tempat Pendirian Badan Usaha' (a dropdown menu with 'Jakarta' selected), 'Nomor Akta Pendirian Usaha' (text input with '1111111111'), 'Tanggal Akta Pendirian Usaha' (a date picker showing '29-02-2011'), 'Nomor Akta Perubahan Terakhir' (text input with '1111111111'), and 'Tanggal Akta Perubahan Terakhir' (a date picker showing '11-05-2022'). Below the 'Identitas Perusahaan' section is another section titled 'Alamat Perusahaan' with a single 'Alamat' text input field.

Gambar 3.3. Desain tab Data Debitur - Marketing halaman Data Pokok
Sumber: Dokumentasi Pribadi (Lampiran 9)

B Hasil Implementasi Modul Pembiayaan Usaha

Bagian ini menjelaskan hasil implementasi antarmuka modul Pembiayaan Usaha berdasarkan rancangan yang telah dibuat oleh tim *UI/UX Designer*. Implementasi dilakukan menggunakan *framework* Spring Boot, dengan penyesuaian pada struktur HTML, CSS, serta integrasi API agar tampilan dan fungsionalitas sistem sesuai dengan rancangan awal yang tercantum dalam *Business Requirement Document* (BRD). Modul Pembiayaan Usaha terdiri atas beberapa halaman utama yang saling terintegrasi dan berfungsi untuk menampilkan serta mengelola data yang berkaitan dengan proses pembiayaan.

Setiap halaman memiliki peran dan fungsionalitas yang berbeda, namun tetap mengikuti standar desain dan alur penggunaan yang telah ditetapkan. Selain itu, seluruh elemen antarmuka dirancang agar responsif dan mudah digunakan, sehingga dapat mendukung efisiensi kerja pengguna dalam proses pengajuan hingga persetujuan pembiayaan. Berikut merupakan halaman-halaman yang diimplementasikan:

1. Halaman Data Pokok

Halaman ini diimplementasikan untuk menampilkan serta mengelola informasi utama terkait identitas perusahaan debitur. Antarmuka halaman dirancang menyerupai rancangan awal pada Figma, dengan beberapa penyesuaian terhadap struktur dan elemen tampilan agar mendukung integrasi data melalui API. Melalui integrasi tersebut, informasi penting seperti profil perusahaan, data legalitas, serta data pengurus atau pemilik dapat diakses dan diperbarui secara dinamis sesuai dengan data terkini yang tersimpan pada sistem.

Untuk mempermudah navigasi dan pengelolaan data, halaman ini dibagi menjadi beberapa bagian utama dalam bentuk *tab* yang mengelompokkan informasi berdasarkan kategorinya. Masing-masing *tab* memiliki fungsi dan jenis data yang berbeda sesuai peran tim pengisi data. Implementasi navigasi antartab menggunakan komponen Bootstrap Tabs, seperti ditunjukkan pada Kode 3.5. Kode tersebut berfungsi untuk membangun struktur navigasi berbasis *tab* yang memungkinkan pengguna berpindah antarbagian data, yaitu Data Debitur–Marketing, Data Debitur–Legal, dan Data Pengurus–Pemilik, tanpa melakukan pemuatan ulang halaman.

```

1 <ul class="nav nav-tabs tabpad">
2   <li class="active">
3     <a data-toggle="tab" href="#debitur-marketing">
4       Data Debitur - Marketing
5     </a>
6   </li>
7   <li class="">
8     <a data-toggle="tab" href="#debitur-legal">
9       Data Debitur - Legal
10    </a>
11  </li>
12  <li class="">
13    <a data-toggle="tab" href="#pengurus-pemilik">
14      Data Pengurus - Pemilik
15    </a>
16  </li>
17</ul>
18
19<div class="tab-content">
20  <div class="tab-pane infoPadding active"
21    id="debitur-marketing">
22    <form id="form-input-marketing" ...>
23  </div>
24
25  <div class="tab-pane infoPadding"
26    id="debitur-legal">
27    <form id="form-input-legal" ...>
28  </div>
29
30  <div class="tab-pane infoPadding"
31    id="pengurus-pemilik">
32    <jsp:include page="/data-pokok-pemilik-pengusaha-view
33    .jsp"/>
34  </div>
35</div>

```

Kode 3.5: Navigasi tab pada halaman Data Pokok

Salah satu bagian penting pada halaman ini adalah *tab* Data Debitur – Marketing yang menampilkan informasi umum mengenai profil perusahaan, seperti nama usaha, bidang usaha, alamat, dan kontak perusahaan. Struktur HTML untuk bagian ini ditunjukkan pada Kode 3.6.

```

1 <div class="tab-content">
2   <div class="tab-pane infoPadding active"
3     id="debitur-marketing">
4     <form id="form-input-marketing">
5       <section>
6         <div class="sectionHeader">
7           <h4>Identitas Perusahaan</h4>
8         </div>
9         <div class="row" ...>
10        </section>
11        <section>
12          <div class="sectionHeader">
13            <h4>Alamat Perusahaan</h4>
14          </div>
15          <div class="row" ...>
16          </section>
17          <section>
18            <div class="sectionHeader">
19              <h4>Data PIC Perusahaan</h4>
20            </div>
21            <div class="row" ...>
22            </section>
23            <section>
24              <div class="sectionHeader">
25                <h4>Dokumen Perusahaan</h4>
26              </div>
27              <div class="row" ...>
28              </section>
29            </form>
30          </div>
31 </div>
32

```

Kode 3.6: Tab Data Debitur - Marketing pada halaman Data Pokok

Tampilan hasil implementasi dari *tab* Data Debitur – Marketing ditunjukkan pada Gambar 3.4, yang memperlihatkan kesesuaian antara rancangan antarmuka dengan hasil pengembangan pada *website* operasional perusahaan.

Gambar 3.4. Tab Data Debitur - Marketing halaman Data Pokok

Selanjutnya, *tab* Data Debitur – Legal berfungsi untuk menampilkan dan mengelola informasi terkait legalitas perusahaan, seperti nomor izin usaha, akta pendirian, serta berbagai dokumen pendukung lainnya. Bagian ini dirancang untuk memastikan bahwa seluruh data hukum dan perizinan perusahaan terekam secara lengkap dan valid sebelum proses pembiayaan dilanjutkan. Struktur utama dari bagian ini diimplementasikan menggunakan elemen HTML berbasis komponen Bootstrap Form, seperti ditunjukkan pada Kode 3.7.

```

1 <form id="form-input-legal">
2   <section>
3     <div class="sectionHeader">
4       <h4>Identitas Perusahaan</h4>
5     </div>
6     <div class="row">
7       <div class="col-sm-4">
8         <div class="form-group">
9           <label class="mandatory">
10            Sektor Usaha
11          </label>
12          <select name="working-field"
13            id="working-field"
14            class="form-control selectpicker"
15            data-live-search="true" required>

```

```

16         <option value="" selected>
17             Silakan Pilih
18         </option>
19     </select>
20 </div>
21 </div>
22 </div>
23 <!-- Elemen HTML lainnya -->
24 </section>
25 </form>
26

```

Kode 3.7: Tab Data Debitur - Legal pada halaman Data Pokok

Melalui struktur tersebut, sistem dapat mengatur dan menampilkan informasi legalitas perusahaan secara sistematis sesuai dengan ketentuan yang berlaku. Tampilan hasil implementasi dari *tab* Data Debitur – Legal ditunjukkan pada Gambar 3.5.

Gambar 3.5. Tab Data Debitur - Legal halaman Data Pokok

Tab terakhir, yaitu Data Pemilik/Pengurus, menampilkan daftar individu yang memiliki peran penting dalam perusahaan, baik sebagai pemilik maupun pengurus aktif. Bagian ini digunakan untuk mengatur data personal yang dibutuhkan dalam analisis pembiayaan, termasuk nama, jabatan, dan status kepemilikan dalam perusahaan. Pemanggilan data dilakukan secara dinamis

melalui integrasi API yang menghubungkan lapisan antarmuka dengan *controller*. Proses ini ditunjukkan pada Kode 3.8, yang memanfaatkan *JavaScript* untuk memuat komponen daftar pengurus perusahaan secara otomatis ke dalam halaman.

```

1 $.get('${local_server}/pembiayaan-usaha/load-component-
  pengurus-perusahaan', function (data) {
2   pengurusPerusahaanInner = data;
3   $("#pengurus-container").append(pengurusPerusahaanInner);
4 }).fail(function (xhr, status, error) {
5   console.error('Error loading file:', xhr.status, error);
6 });
7

```

Kode 3.8: Pemanggilan komponen Data Pengurus/Pemilik

Permintaan tersebut diarahkan ke *endpoint* `/pembiayaan-usaha/load-component-pengurus-perusahaan` dan diproses oleh *controller* yang bertanggung jawab untuk memuat serta mengembalikan komponen JSP terkait, seperti ditunjukkan pada Kode 3.9.

```

1 @RequestMapping(value = "/pembiayaan-usaha/load-component-
  pengurus-perusahaan")
2 public String loadComponentPerusahaan() {
3   return "component-pengurus-perusahaan";
4 }
5

```

Kode 3.9: Controller pemanggilan komponen Data Pengurus/Pemilik

Melalui mekanisme ini, data dan komponen tampilan dapat dimuat secara modular tanpa perlu melakukan pemuatan ulang halaman secara keseluruhan. Pendekatan tersebut mendukung efisiensi proses pengembangan serta meningkatkan pengalaman pengguna dalam mengelola data. Gambar 3.6 menunjukkan tampilan hasil implementasi dari *tab* Data Pemilik/Pengurus.

Gambar 3.6. Tab Data Pemilik/Pengurus halaman Data Pokok

Proses input dan validasi data dilakukan secara langsung pada masing-masing *tab* dengan antarmuka yang responsif dan mudah digunakan. Setiap elemen formulir telah dilengkapi dengan validasi otomatis untuk memastikan kelengkapan dan kesesuaian data sebelum disimpan ke sistem. Selain itu, pembaruan data dilakukan secara dinamis melalui integrasi API, sehingga perubahan yang dilakukan oleh pengguna dapat langsung tersinkronisasi dengan basis data utama tanpa perlu melakukan pemuatan ulang halaman secara keseluruhan.

2. Halaman Data Pembiayaan

Halaman ini merealisasikan fungsi pengelolaan data pembiayaan sesuai kebutuhan pengguna internal. Implementasi dilakukan dengan menambahkan elemen interaktif seperti tabel, *form* input, serta validasi otomatis untuk memastikan kesesuaian data dengan aturan pembiayaan yang berlaku. Melalui integrasi API, seluruh data tersebut dapat diperbarui secara dinamis, sehingga memudahkan proses pengelolaan dan *monitoring* pembiayaan oleh pihak internal perusahaan.

Struktur navigasi pada halaman ini menggunakan komponen *tab* untuk memisahkan dua bagian utama, yaitu Barang Agunan dan Detail Pembiayaan. Kode 3.10 menunjukkan struktur *tab* tersebut.

```

1 <ul class="nav nav-tabs tabpad">
2   <li class="active">
3     <a data-toggle="tab" href="#barang_agunan">
4       Barang Agunan
5     </a>
6   </li>
7   <li class="">
8     <a data-toggle="tab" href="#detail_pembiayaan">
9       Detail Pembiayaan
10    </a>
11  </li>
12</ul>
13
14<div class="tab-content">
15  <div class="tab-pane infoPadding active"
16    id="barang_agunan">
17    <jsp:include page='/kuh_barang_agunan_view.jsp' />
18  </div>
19
20  <div class="tab-pane infoPadding"
21    id="detail_pembiayaan">
22    <jsp:include page='/kuh_detail_pembiayaan_view.jsp' />
23  </div>
24</div>
25

```

Kode 3.10: Tab halaman Data Pembiayaan

Bagian Barang Agunan digunakan untuk menampilkan dan mengelola data agunan yang menjadi jaminan atas pembiayaan yang diajukan. Pengguna dapat menambahkan, memperbarui, atau menghapus data agunan melalui *form* yang telah dilengkapi dengan validasi otomatis. Pemanggilan komponen dilakukan secara dinamis melalui fungsi JavaScript seperti ditunjukkan pada Kode 3.11.

```

1 $.get( '${local_server}/pembiayaan-usaha/load-component-barang
  -agunan', function (data) {
2   barangAgunanInner = data;
3   $("#barang-agunan-container").append(barangAgunanInner);
4 }).fail(function (xhr, status, error) {
5   console.error('Error loading file:', xhr.status, error);
6 });
7

```

Kode 3.11: Pemanggilan komponen Barang Agunan

Permintaan tersebut diarahkan menuju salah satu metode pada *controller*, seperti ditunjukkan pada Kode 3.12. Metode ini berfungsi untuk mengembalikan komponen JSP `component_barang_agunan` agar dapat dimuat ke dalam halaman utama secara dinamis.

```

1 @RequestMapping(value = "/load-component-barang-agunan")
2 public String loadComponentBarangAgunan() {
3     return "/kuh-component-barang-agunan";
4 }
5

```

Kode 3.12: Controller pemanggil komponen Barang Agunan

Gambar 3.7 menunjukkan tampilan hasil implementasi untuk *tab* Barang Agunan.

The screenshot shows the 'Pembiayaan Usaha - Data Pembiayaan' page in the DanaKini application. The page is divided into several sections:

- Header:** DanaKini logo and user profile 'Vianca Vanesia Barhan'.
- Form Fields:**
 - Nama Debitur: vien company
 - Email Debitur: L.vianca.barhan@danakini.co.id
 - NPWP Debitur: 111111111111111111
 - Status Data Pembiayaan: FINALIZED - Approved
- Barang Agunan (Collateral):**
 - Barang 1:
 - Nomor Agunan: 01001010
 - Nama Barang: bangku
 - Status Agunan: Tersedia
 - Jenis Agunan: Barang Konsumsi - Kendaraan Bermotor Roda Dua
 - Nama Pemilik Agunan: vien
 - Bukti Kepemilikan: kepo
 - Alamat Agunan: jakarta
 - Kota/Kabupaten Lokasi Agunan: Kota Adm. Jakarta Pusat
 - Nilai Agunan sesuai NJOP (Rp): 929
 - Nilai Agunan menurut Pelapor (Rp): 992
 - Tanggal Penilaian Agunan Menurut Pelapor: 21-08-2025
 - Diasuransikan: Ya
 - Metode CKPN: Individual
- Asuransi Agunan (Collateral Insurance):**
 - Nama Asuransi: vien
 - Biaya Asuransi: 101,011

Gambar 3.7. Tab Barang Agunan halaman Data Pembiayaan

Sementara itu, bagian Detail Pembiayaan menampilkan informasi utama terkait struktur pembiayaan, seperti kode produk, jumlah pinjaman, tenor, dan status pengajuan. Bagian ini memberikan gambaran besar terkait kondisi pembiayaan yang diajukan oleh debitur, sehingga memudahkan tim terkait dalam melakukan analisis dan pengambilan keputusan. Kode 3.13 memperlihatkan implementasi elemen formulir untuk menampilkan dan mengelola data pembiayaan.

```

1 <div class="row">
2   <div class="col-sm-4">
3     <div class="form-group">
4       <label class="mandatory">
5         Tanggal Mulai Pinjaman
6       </label>
7       <div class="layout-modal-date">
8         <input type="text"
9           class="form-control datepicker-loan"
10          onkeydown="return false;"
11          name="loan-start-date"
12          id="loan-start-date" required>
13         <i class="fa fa-calendar-o"></i>
14       </div>
15     </div>
16   </div>
17 </div>
18

```

Kode 3.13: Struktur HTML bagian Detail Pembiayaan

Untuk memastikan ketepatan input tanggal, sistem memanfaatkan fungsi `initDateTimePicker()` yang memanggil pustaka Bootstrap Datepicker. Fungsi ini membatasi tanggal yang dapat dipilih sesuai dengan parameter waktu yang ditentukan. Kode 3.14 menunjukkan implementasi dari Bootstrap Datepicker.

```

1 function initDateTimePicker(startDate , maxDate){
2   $(''.datepicker-loan').datetimepicker({
3     useCurrent: false ,
4     format: "DD-MM-YYYY" ,
5     minDate: startDate ,
6     maxDate: maxDate
7   });
8 }
9

```

Kode 3.14: Inisialisasi komponen Datepicker pada Detail Pembiayaan

Hasil implementasi dari bagian Detail Pembiayaan ditunjukkan pada Gambar 3.8, yang memperlihatkan antarmuka formulir untuk input data pembiayaan.

Pembiayaan Usaha - Data Pembiayaan

Nama Debitur *
vien company

Email Debitur *
l_vianca.barhan@danakini.co.id

NPWP Debitur *
111111111111111111

Barang Agunan Detail Pembiayaan

Status Data Pembiayaan: FINALIZED - Approved

Detail Pembiayaan

Total Pembelian Barang *	Uang Muka (Rp) *	Uang Muka *
10	1,000,000,000	10000000000 %
Nilai Pembiayaan (Rp) *	Jangka Waktu (Bulan) *	Bunga per Tahun *
0	12	3 %
Angsuran per Bulan *	Payment Frekuensi *	Denda Keterlambatan *
0	Arutas	0.1 %
Nilai Proyek *		
1,000,000		
Biaya		
Biaya Notaris (Rp) *	Biaya Administrasi (Rp) *	
10,000,000	2,500,000	
Informasi Merchant		
Bank Merchant *	Nomor Rekening Merchant *	
PT. BANK SAHABAT SAMPOERNA	1000000	
Informasi Mulai Pinjaman		

Gambar 3.8. Tab Detail Pembiayaan halaman Data Pembiayaan

3. Halaman Approval Pembiayaan Usaha

Halaman ini berfungsi sebagai tahap akhir dalam proses pengajuan pembiayaan, yang digunakan oleh pihak manajemen untuk melakukan pemeriksaan dan memberikan keputusan akhir terhadap data yang telah diinput oleh tim Marketing dan tim Legal. Secara struktur, halaman ini memuat tiga *tab* utama, yaitu Approval, Data Pokok, dan Data Pembiayaan. Berbeda dengan dua halaman sebelumnya, *tab* Data Pokok dan Data Pembiayaan pada tahap ini bersifat *read-only* dan hanya digunakan untuk meninjau kembali informasi debitur serta rincian pembiayaan yang telah diajukan.

Struktur dasar pembagian *tab* pada halaman ini ditunjukkan pada Kode 3.15, yang mengatur pemisahan konten menjadi beberapa bagian agar setiap informasi dapat ditampilkan secara terstruktur dan mudah diakses pengguna.

```

1 <ul class="nav nav-tabs tabpad">
2   <li class="active">
3     <a data-toggle="tab" href="#tab-approval">
4       Approval
5     </a>
6   </li>
7   <li class="">
8     <a data-toggle="tab" href="#tab-data-pokok">
9       Data Pokok
10    </a>
11  </li>
12  <li class="">
13    <a data-toggle="tab" href="#tab-data-pembiayaan">
14      Data Pembiayaan
15    </a>
16  </li>
17</ul>
18
19<div class="tab-content">
20  <div class="tab-pane infoPadding active"
21    id="tab-approval">
22    <form id="form-approval" ...>
23  </div>
24  <div class="tab-pane infoPadding" id="tab-data-pokok">
25    <div class="row">
26      <jsp:include page="data_pokok_detail_view.jsp"/>
27    </div>
28  </div>
29  <div class="tab-pane infoPadding"
30    id="tab-data-pembiayaan">
31    <div class="row">
32      <jsp:include page="data_pembiayaan_detail_view.
33      jsp"/>
34    </div>
35  </div>
36</div>

```

Kode 3.15: Struktur tab pada halaman Approval Pembiayaan Usaha

Inisialisasi data dilakukan secara asinkron melalui fungsi `loadData()` yang dipanggil saat halaman dimuat. Proses ini ditunjukkan pada Kode 3.16, di mana setiap fungsi pemanggilan data (API *request*) seperti `getDDLBank()`, `loadProvince()`, dan `runSimulation()` dipanggil secara berurutan menggunakan mekanisme *async/await* untuk memastikan urutan eksekusi yang konsisten.

```
1 $(document).ready(function() {  
2     getDDLBank();  
3     loadData();  
4     async function loadData() {  
5         try {  
6             showLoadingProg(true, "");  
7             const ddlProv = await loadProvince();  
8             const ddl = await getDDL();  
9             const data = await getData();  
10            const simulasi = await runSimulation();  
11        } catch (error) {  
12            console.error("Error:", error);  
13        }  
14    }  
15 });  
16
```

Kode 3.16: Proses inisialisasi data pada tab Approval

Bagian ini membantu memastikan konsistensi data antar-tahap proses dan memudahkan manajemen dalam melakukan verifikasi sebelum memberikan keputusan akhir. Setiap *tab* dirancang agar informasi dapat diakses dengan cepat tanpa mengubah atau mengedit data yang telah tersimpan. Tampilan hasil implementasi *tab* Data Pokok ditunjukkan pada Gambar 3.9.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Gambar 3.9. Tab Data Pokok halaman Approval Pembiayaan Usaha

Untuk menjaga agar halaman bersifat *read-only*, fungsi `disabledInput()` digunakan untuk menonaktifkan seluruh elemen input dan tombol interaktif pada halaman. Kode 3.17 menunjukkan implementasi fungsi tersebut.

```

1 function disabledInput(type){
2   let item = type
3   item.find("input").attr("readonly", "readonly");
4   item.find("select").attr("readonly", "readonly");
5   item.find("select.selectpicker").closest(".bootstrap-select").addClass("readonly");
6   item.find("select.selectpicker").selectpicker("refresh");
7   item.find(".file-upload").hide();
8   item.find(".file-delete").hide();
9   item.find(".file-upload-owner").hide();
10  item.find(".file-delete-owner").hide();
11  item.find(".wrapper-save-submit").remove();
12  item.find('.barang-agunan-group-add').remove();
13  item.find('.group-delete').remove();
14  item.find('.delete-barang-inner').remove();
15  item.find('.add-button').remove();
16 }
17

```

Kode 3.17: Fungsi untuk menonaktifkan input pada tab Approval

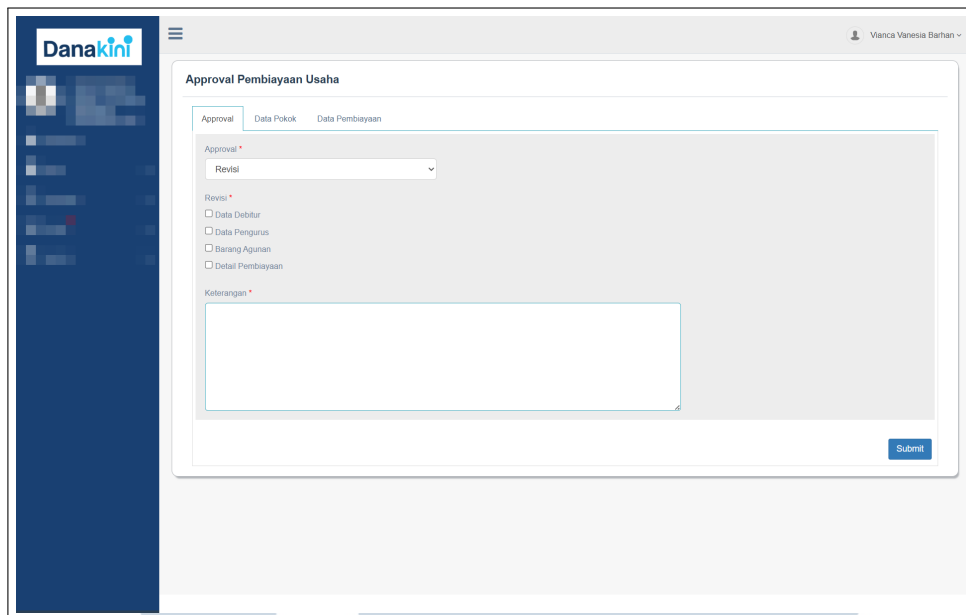
Sementara itu, *tab* Approval merupakan satu-satunya bagian yang bersifat interaktif. Pada bagian ini, pihak manajemen dapat memberikan keputusan akhir berupa persetujuan atau permintaan revisi terhadap pengajuan pembiayaan. Jika diperlukan revisi, sistem akan menampilkan bagian tambahan untuk memasukkan keterangan revisi seperti ditunjukkan pada Kode 3.18.

```
1 $(document).ready(function(e){
2     $("#approval").on("change",function(){
3         let curVal = $(this).val();
4
5         if (curVal == "Approved"){
6             $(".group-revisi").remove();
7         }else{
8             $("#revisi-section").append(revisiSection);
9         }
10    })
11 });
12
```

Kode 3.18: Logika interaktif pada tab Approval

Bagian ini dirancang untuk memberikan fleksibilitas bagi manajemen dalam menentukan hasil evaluasi secara langsung melalui antarmuka sistem. Selain menampilkan opsi keputusan, halaman juga menyesuaikan tampilan secara otomatis ketika status *approval* berubah, sehingga pengguna dapat segera menginput catatan revisi tanpa perlu memuat ulang halaman. Gambar 3.10 menunjukkan tampilan hasil implementasi *tab* Approval.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.10. Tab Approval halaman Approval Pembiayaan Usaha

Dari sisi implementasi, halaman ini dikembangkan dengan cara mengintegrasikan ulang komponen dari halaman Data Pokok dan Data Pembiayaan, serta menambahkan satu *tab* baru khusus untuk *approval*. Pendekatan ini memungkinkan efisiensi dalam pengembangan karena sebagian besar elemen antarmuka dapat digunakan kembali tanpa perlu penulisan kode dari awal. Selain itu, pendekatan tersebut juga mempermudah proses pemeliharaan sistem di kemudian hari karena struktur kode menjadi lebih konsisten dan modular.

Untuk menampilkan data pembiayaan usaha secara dinamis, sistem juga dilengkapi dengan integrasi API yang memungkinkan pertukaran data antara antarmuka pengguna dan lapisan *backend* secara langsung. Melalui mekanisme ini, informasi dari basis data dapat diakses dan diperbarui secara efisien tanpa memerlukan proses pemuatan ulang halaman.

3.4 Kendala dan Solusi yang Ditemukan

Dalam proses pengembangan, ditemukan beberapa kendala teknis dan non-teknis yang memengaruhi pelaksanaan sistem, beserta solusi yang diterapkan untuk mengatasinya.

3.4.1 Kendala

Beberapa kendala ditemukan dalam pengembangan *website*, antara lain:

1. Kesulitan dalam menyesuaikan kode program saat melakukan *upgrade libraries* dan *plugin*, karena terdapat perbedaan sintaks dan struktur fungsi dari versi sebelumnya.
2. Kesulitan dalam memahami alur sistem yang telah dikembangkan sebelumnya, karena ketiadaan dokumentasi teknis yang cukup lengkap, sehingga proses penyesuaian terhadap sistem menjadi lebih memakan waktu.
3. Kurangnya sinkronisasi antara pengembangan *Front End* dan *Back End*, yang menyebabkan proses integrasi data tidak selalu berjalan dengan optimal.

3.4.2 Solusi

Adapun solusi yang telah berhasil disusun dan dijalankan untuk mengatasi kendala yang dihadapi meliputi:

1. Melakukan pembelajaran mandiri terhadap dokumentasi resmi *libraries* dan *plugin* serta membandingkan perbedaan sintaks antarversi guna menyesuaikan implementasi kode yang digunakan pada proyek, sehingga fungsionalitas sistem tetap berjalan sesuai dengan versi sebelumnya.
2. Melakukan pembacaan dan penelusuran alur sistem secara bertahap, disertai diskusi dengan mentor atau anggota tim untuk mendapatkan gambaran yang lengkap mengenai mekanisme sistem serta keterkaitan antar komponennya.
3. Melakukan koordinasi rutin antara tim *Front End* dan *Back End* melalui komunikasi yang jelas serta penyepakatan format data untuk memastikan kesesuaian proses integrasi dan komunikasi antar sistem.