BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Pelaksanaan kerja magang dalam bidang *Web Application Development* dilakukan di bawah bimbingan Yulianto selaku CEO di PT IKKAT Inovasi Teknologi. Dalam proses melakukan pekerjaan pada modul Action Process pada aplikasi Loan Origination System, setiap minggunya akan diberikan pekerjaan spesifik untuk dikerjakan dan apabila terdapat pertanyaan dapat bertanya secara langsung maupun menggunakan *WhatsApp* secara personal ataupun bertanya pada *discussion group* yang ada. Setiap Jumat diadakan pertemuan untuk membahas pekerjaan yang sudah dilakukan beserta revisi dan saran yang membangun, serta pemberian tugas berikutnya.

3.2 Tugas dan Uraian Kerja Magang

Dalam melaksanakan kerja magang pada PT IKKAT Inovasi Teknologi, terdapat beberapa tugas yang dilakukan di dalam modul Action Process pada aplikasi Loan Origination System.

3.2.1 Tugas yang Dilakukan

Modul Action Process dibuat pada aplikasi Loan Origination System berbasis *web* dengan menggunakan *framework* ReactJS. Terdapat beberapa modul dan komponen yang diperlukan di dalam modul Action Process. Modul-modul yang digunakan adalah sebagai berikut.

1. Approval Process berisikan data berstatus Underwriting, Rejected, dan Approved yang belum diselesaikan.
2. Underwriting berisikan data berstatus Underwriting yang memiliki nilai filter di garis tengah dan dapat ditentukan apakah diterima atau ditolak.
3. Rejected berisikan data berstatus Rejected yang merupakan data yang ditolak karena tidak sesuai dengan filter pada modul Filter.
4. Filter berisikan data filter untuk menguji kecocokan antara Loan Products dengan data *user*.
5. Completion berisikan data berstatus Approved yang belum diselesaikan prosesnya dan dapat ditentukan apakah data tersebut sudah selesai atau belum.
6. History berisikan catatan dari proses *loan* yang berhasil dengan status Approved dan sudah diselesaikan.

Komponen-komponen yang digunakan adalah sebagai berikut.

1. Tabel yang dapat mengambil data yang sudah dibuat melalui *form* untuk ditampilkan. Selain data utama, tabel juga memberikan *icon* untuk menampilkan data lain yang juga dibuat melalui *form*. Terdapat juga fitur *search* untuk memudahkan dalam mencari data.
2. *Speedometer* untuk menunjukkan nilai yang didapat dari kecocokan antara data dengan Loan Products yang digunakan untuk menentukan apabila data diterima atau ditolak.

3.2.2 Uraian Pelaksanaan Kerja Magang

Uraian pekerjaan yang dikerjakan selama proses kerja magang adalah sebagai berikut.

Tabel 3.1 Uraian Proses Pelaksanaan Kerja Magang

Minggu	Uraian Tugas
1	- Mempelajari penggunaan *ReactJS* dan pembuatan program sederhana - Penjelasan mengenai proyek yang dibuat
2	- Perencanaan desain aplikasi - Membuat tampilan sederhana pada *website* - Membuat tabel dengan data sederhana - Membuat *Progress Bar* dalam menghitung suatu nilai
3	- Membuat *tabs* - Membuat *dummy* JSON - Mengubah *Progress Bar* menjadi *Speedometer*
4	- Menggunakan React Bootstrap Table untuk menampilkan *dummy* JSON - Menggunakan fungsionalitas dari React Bootstrap Table
5	- Membuat dan menampilkan Filter - Membuat daftar Filter dan nilai yang sesuai
6	- Membuat Expand Data pada React Bootstrap Table
7	- Menggunakan *Accordion* pada Filter - Menentukan isi dari Underwriting
8	- Merapikan *styling* yang digunakan - Memperbaiki *error* dan *bug* yang ditemukan
9	- *Routing* data pada *navigation* - Mengambil data dari *localhost* - Mengubah data Loan dan menyimpannya
10	- Memanggil data Filter dari *localhost* - Menambah data Filter dan menyimpannya - Mengubah data Filter dan menyimpannya - Menghapus data Filter
11	- Memperbaiki *error* dan *bug* yang ditemukan - Merapikan *codes*

Tabel 3.1 merupakan uraian mengenai tugas yang dikerjakan selama 11 minggu kegiatan magang berlangsung di PT IKKAT Inovasi Teknologi dibagi menjadi tiap minggu.

Pada minggu pertama, *framework* ReactJS dipelajari secara pribadi dengan mengikuti panduan di internet untuk membuat program-program sederhana supaya terbiasa. Pada hari Jumat melalui pertemuan pertama dijelaskan mengenai proyek yang dibuat dan pembagian tugas. Proyek yang diberikan ini sebagai bentuk pembelajaran dalam pembuatan suatu program yang berkaitan dengan peminjaman dan yang sesuai dengan pengembangan serta kriteria yang sudah diberikan. Dibuatlah grup diskusi pada *WhatsApp* untuk mempermudah kontak apabila pembimbing sedang tidak berada di lokasi magang.

Pada minggu kedua, membuat tampilan aplikasi secara sederhana di Visual Studio Code berdasarkan desain dari aplikasi *Loan Originated System*. Secara perlahan, aplikasi kemudian diubah menjadi bentuk ReactJS. Bagian utama dari aplikasi adalah pada modul *All* sehingga diutamakan untuk membuatnya terlebih dahulu. Dibuat tabel untuk menampung data dan dibuat secara manual terlebih dahulu. Kemudian dibuat *Progress Bar* untuk menandai nilai dari data apakah sudah sesuai dengan filter yang berlaku atau belum.

Pada minggu ketiga, modul-modul lainnya mulai dibuat dan digunakan *tabs* untuk memisah antar modul namun masih saling terkait. *Dummy* dibuat di *file* JSON berdasarkan *fields* yang diperlukan pada data nanti sebagai contoh data. Ditemukan bentuk yang lebih sesuai untuk menandai nilai dari data sehingga *Progress Bar* diubah menjadi *Speedometer*. *Speedometer* dibuat dan didesain berdasarkan instruksi dari pembimbing.

Pada minggu keempat, menggunakan React Bootstrap Table untuk dapat menghubungkan data *dummy* JSON ke dalam tabel pada aplikasi. Membaca

dokumentasi dari React Bootstrap Table untuk dapat menggunakan dan memanfaatkan fungsionalitas yang ada. Tabel yang dibuat menggunakan React Bootstrap Table kemudian diubah tampilan dan fiturnya untuk menyesuaikan kebutuhan klien.

Pada minggu kelima, modul Filter dibuat berisikan tabel yang memuat *fields* dan nilai Filter yang sesuai berdasarkan data yang dapat diisi pada *form*. Modul Filter direncanakan bagaimana cara untuk ditampilkan sehingga klien dapat mengerti.

Pada minggu keenam, data lainnya yang diisi pada *form* yang tidak ditampilkan pada tabel di modul All dimasukkan ke dalam data tambahan pada tabel. Data tambahan ini bersifat tersembunyi dan dapat ditampilkan apabila *icon* pada baris data diklik sehingga data tambahan terlihat. Untuk mempersingkat tampilan pada data tambahan, maka data tambahan dipecah ke dalam berbagai *tabs* sesuai dengan struktur tabel. Data tambahan juga dibuat dengan tabel sederhana sehingga formatnya menjadi lebih rapi.

Pada minggu ketujuh, berdasarkan pada penggunaan Expand Data di modul All, maka tiap Loan Products pada modul Filter dapat ditampilkan dan disembunyikan menggunakan *Accordion* sehingga nantinya *user* dapat langsung memilih untuk menampilkan isi dari Loan Products yang diinginkan. Kemudian data *underwriting* pada modul Underwriting diisi dan ditentukan tingkatan berdasarkan *Speedometer*.

Pada minggu kedelapan, *codes* dirapikan dan dilakukan evaluasi singkat mengenai progres yang sudah dilakukan. Dilakukan perbaikan terhadap *error* dan

bug yang ditemukan. Pekerjaan dari minggu-minggu sebelumnya dicek ulang dan dikembangkan. *Styling* pada tiap modul diperbaiki pada CSS yang ada.

Pada minggu kesembilan, digunakan *navigation bar* untuk memudahkan dalam melakukan perpindahan antar modul dengan melakukan *routing* pada *file* config. Data pada *file* JSON kemudian dimasukkan ke *local server* pada *localhost* dan dipanggil ke modul yang terkait. Modul Underwriting dapat mengubah nilai pada status dan menyimpannya pada *file* JSON dengan menggunakan *dropdown*. Modul Completion dapat mengubah nilai pada Complete. Data pada tabel di tiap modul akan berubah apabila halaman di *refresh* karena data yang berubah.

Pada minggu kesepuluh, modul Filter juga memanggil data dari *localhost*. Selain dipanggil, data tersebut juga dapat diubah nilai nya dan disimpan pada *file* JSON nya. Terdapat juga *Insert Data* untuk menambah jumlah filter dan *Delete Data* dan menghapus jumlah filter pada Loan Products. Filter pada setiap Loan Products berbeda sehingga akan dipisah pada tempat terpisah yang bentuknya identik.

Pada minggu kesebelas, *codes* dirapikan. Dilakukan *bug testing* dengan berbagai kondisi untuk mengecek apakah terdapat *error*. Dilakukan *review* terakhir terhadap hasil dari aplikasi yang dibuat dan apa yang dapat dikembangkan oleh perusahaan.

3.3. Pelaksanaan Kerja Magang

3.3.1. Application Requirements

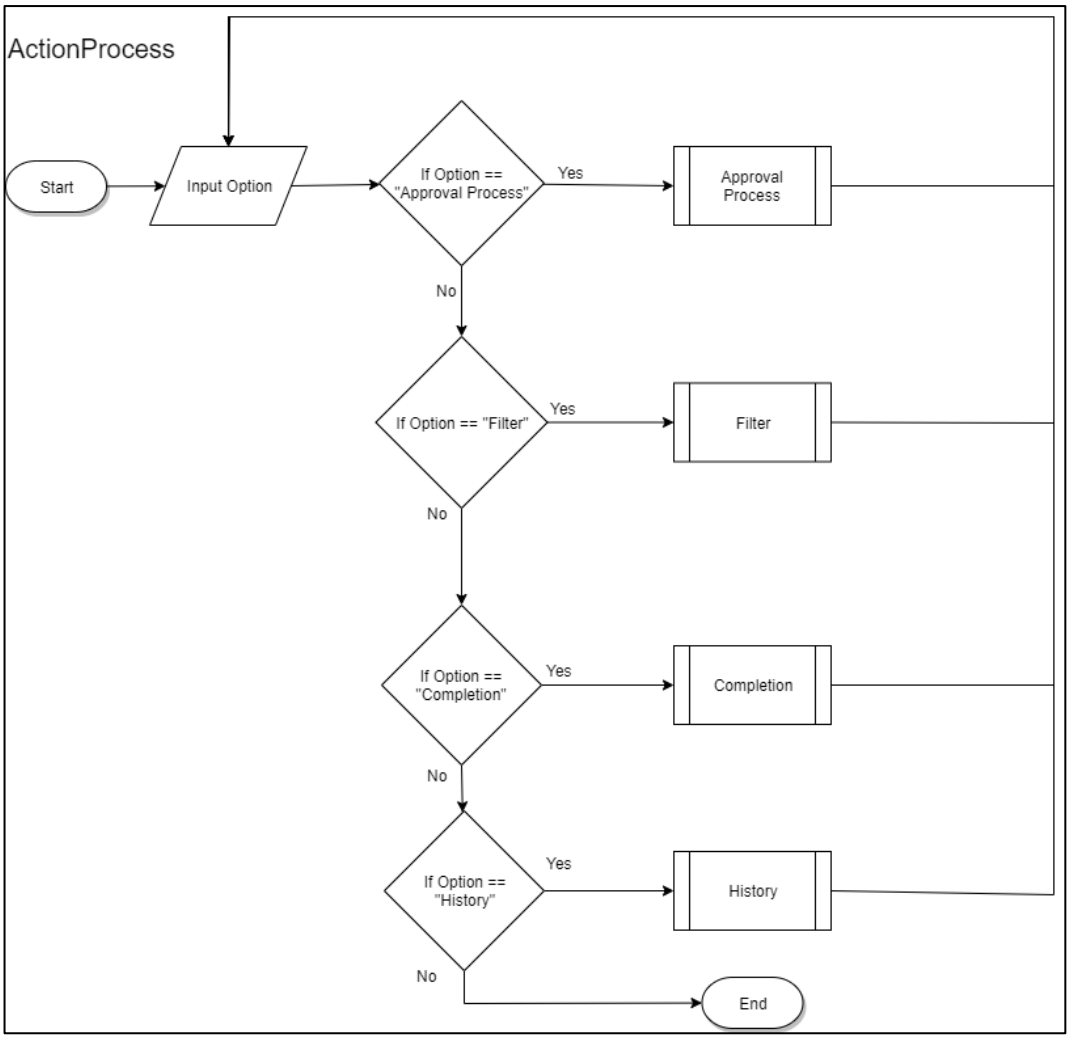
Terdapat beberapa *application requirements* dalam membangun modul Action Process.

1. Modul Action Process dapat memanggil data dari *file* JSON supaya dapat ditampilkan ke dalam tabel yang sudah dibuat.
2. Admin dapat melakukan proses Expand Data sehingga modul Action Process menampilkan data dari *form* ke dalam Expand Data. Admin dapat melihat seluruh data dari user apabila *icon* diklik. Fitur ini berguna terutama pada tabel Underwriting untuk membandingkan data dengan filter.
3. Admin dapat mengubah nilai pada *field* Status dari Underwriting menjadi Approved atau Rejected. Data yang *field* Statusnya terjadi pengubahan akan menghilang dari tabel Underwriting dan akan bertambah pada tabel Completion maupun Rejected.
4. Admin dapat mengubah nilai pada *field* Complete dari NOT DONE menjadi DONE!. Data yang *field* Complete -nya berubah akan dianggap selesai prosesnya dan akan berpindah ke tabel History.
5. Modul Action Process menyimpan pengubahan data pada modul Filter di tiap Loan Products. Admin dapat melakukan pengubahan terhadap isi dari JSON filter pada masing-masing Loan Products seperti menambah data filter, menghapus data filter, dan mengubah data filter. Data yang sudah berubah akan disimpan ke dalam file JSON.

3.3.2. Perancangan

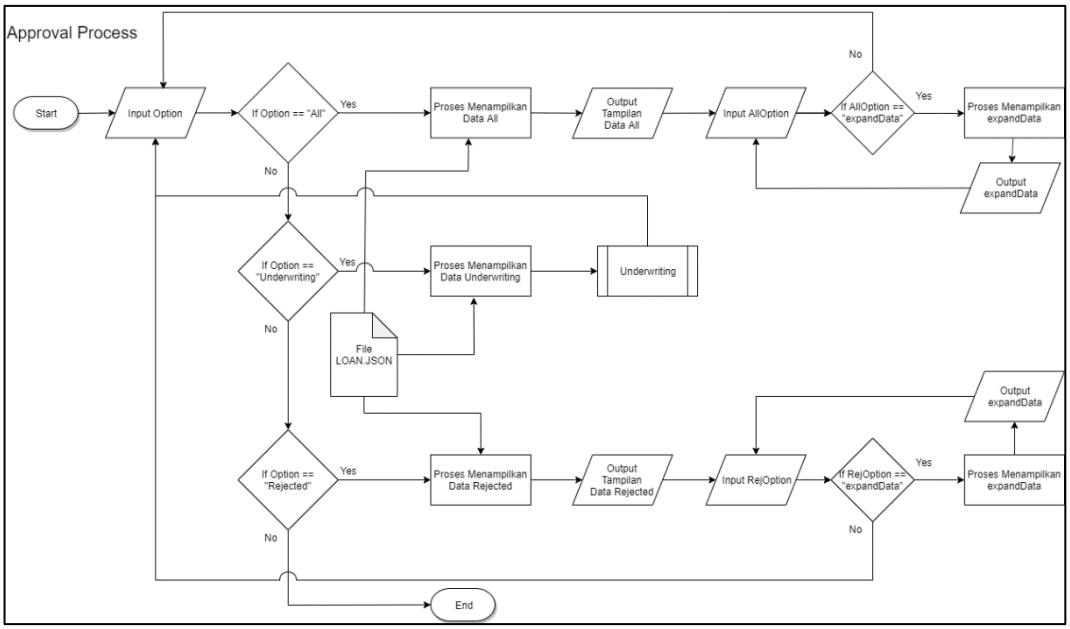
A. Flowchart

Pada bagian ini akan dijelaskan mengenai alur kerja berupa *flowchart* pada modul Action Process di aplikasi Loan Originated System.



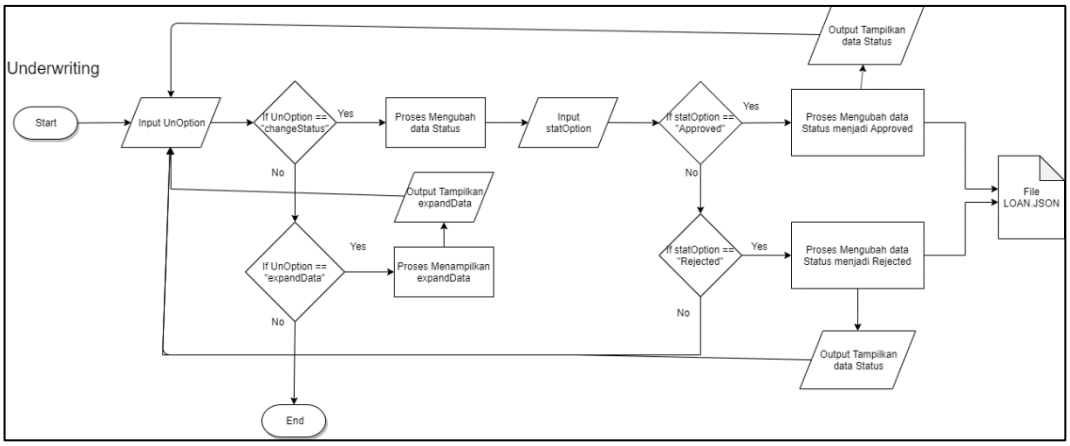
Gambar 3.1 *Flowchart* Modul Action Process

Flowchart Action Process pada Gambar 3.1 untuk menampilkan data dari *user* yang sudah diisi ke dalam *form*. Prosesnya dimulai dari admin memilih halaman yang dituju pada *navigation bar*. Apabila diklik, tampilan akan berubah sesuai dengan pilihan dari admin.



Gambar 3.2 *Flowchart* Modul Approval Process

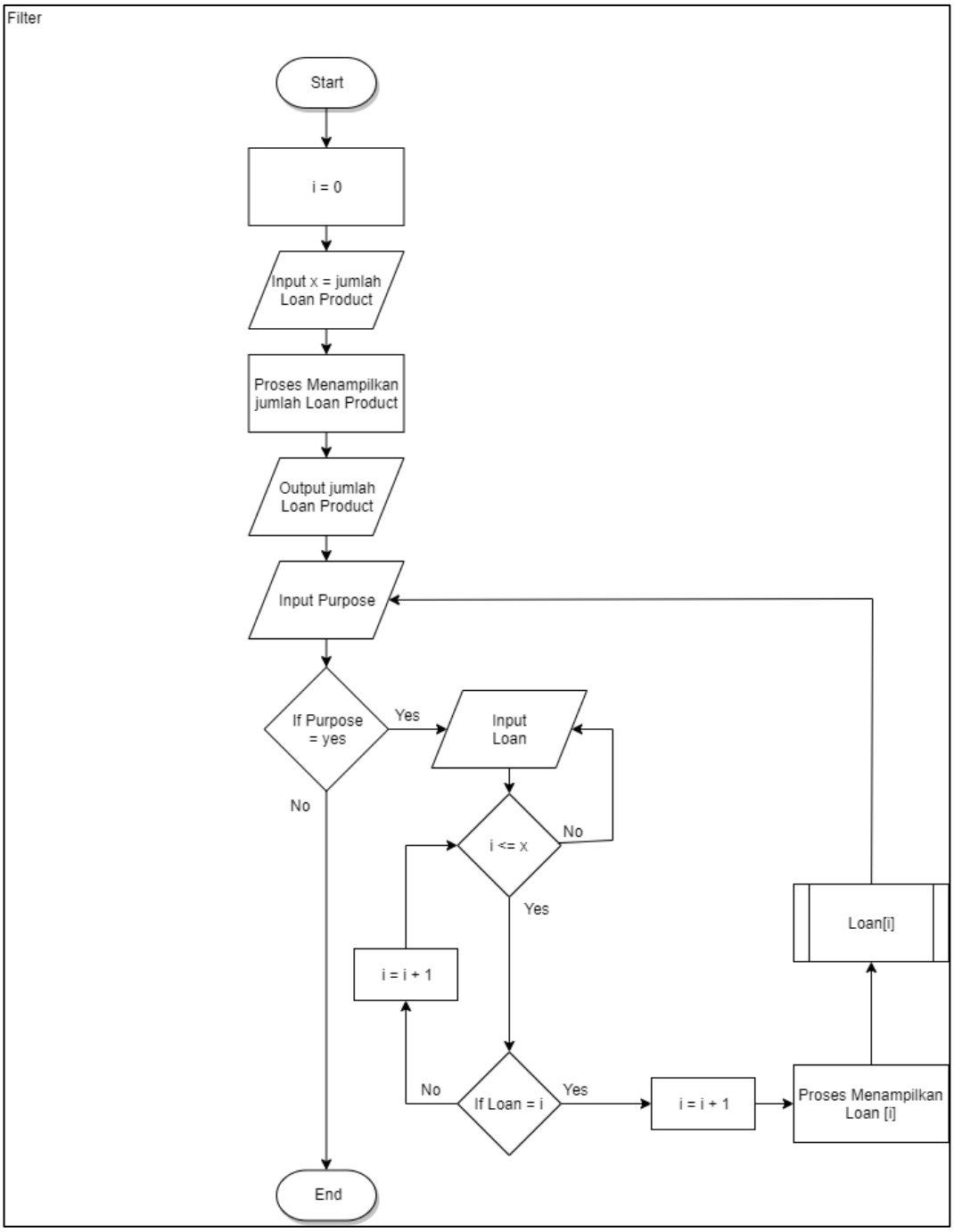
Flowchart Approval Process pada Gambar 3.2 untuk menampilkan semua data dari *user* yang sudah diisi ke dalam *form* kecuali yang prosesnya sudah selesai. Terdapat 3 modul di dalam modul Approval Process yaitu All, Underwriting, dan Reject. Prosesnya dimulai dari admin memilih halaman yang dituju pada *tabs*. Apabila diklik, tampilan akan berubah sesuai dengan pilihan dari admin. Data akan diambil dari *localhost* untuk ditampilkan pada tabel yang sudah ada. Apabila diklik, maka akan dilakukan proses Expand Data untuk memunculkan data lainnya.



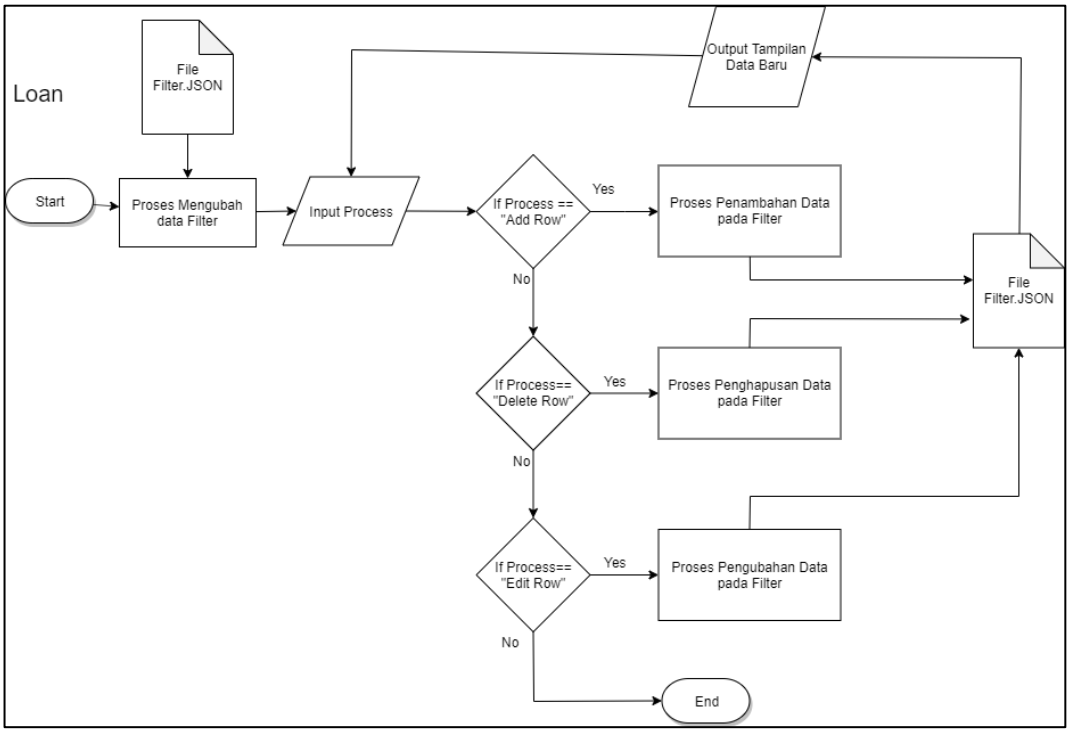
Gambar 3.3 *Flowchart* Modul Underwriting

Flowchart Underwriting pada Gambar 3.3 untuk menampilkan data dari *user* yang berstatus Underwriting. Data akan diambil dari *localhost* untuk ditampilkan pada tabel yang sudah ada. Admin dapat mengklik *icon* untuk melakukan proses Expand Data. Selain itu juga terdapat proses ketika admin dapat mengubah nilai dari status yang ada kepada pilihan yang disediakan sehingga nilainya berubah dan disimpan pada *localhost*. Admin juga memiliki pilihan untuk tidak mengubah nilai Status.

Flowchart Filter pada Gambar 3.4 untuk menampilkan filter yang dibuat oleh admin untuk kemudian dibandingkan dengan data yang ada. Terdapat beberapa Loan Products dimana masing-masing Loan Products memiliki syarat dan ketentuan yang berbeda sehingga harus ditampung pada *file* JSON yang berbeda.

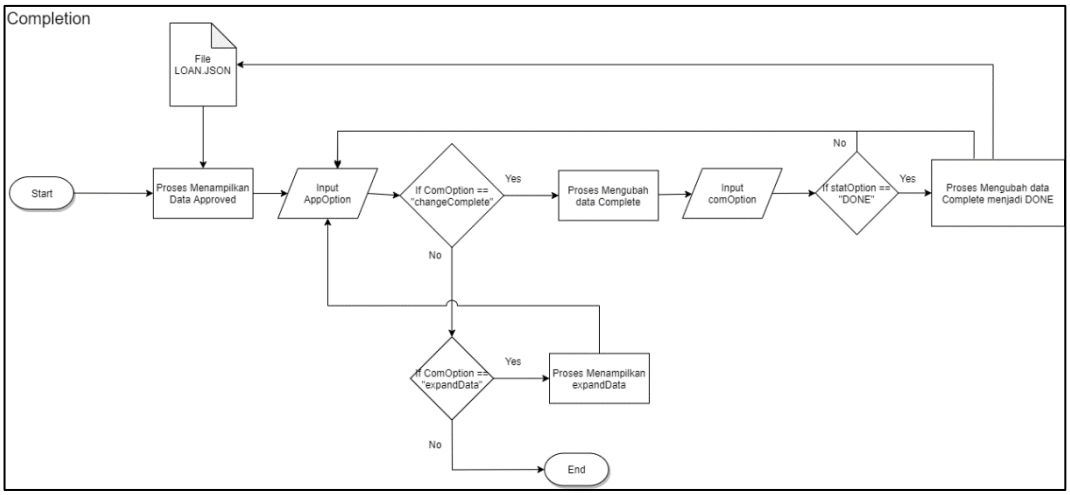


Gambar 3.4 Flowchart Modul Filter



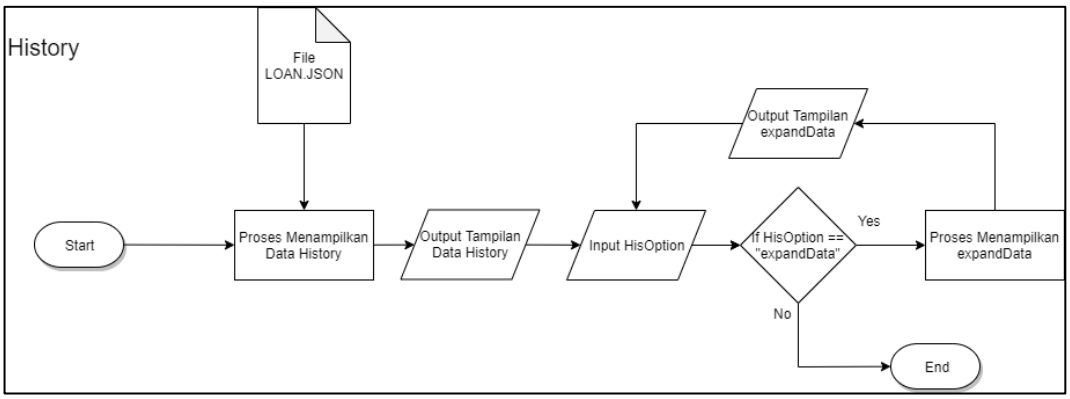
Gambar 3.5 *Flowchart* Modul Loan

Flowchart Loan pada Gambar 3.5 untuk menampilkan filter yang dibuat oleh admin untuk Loan Products. Data akan diambil dari *localhost* untuk ditampilkan pada tabel yang sudah ada. Terdapat tabel berisikan panduan atas *fields* dan *values* yang dapat diisikan. Data filter juga dapat ditambah, dihapus, dan diubah. Data akan disimpan pada *file* Loan.



Gambar 3.6 Flowchart Modul Completion

Flowchart Completion pada Gambar 3.6 untuk menampilkan semua data dari *user* yang memiliki Status Approved dan Complete bernilai NOT DONE. Admin dapat memilih untuk mengklik *icon* untuk melakukan proses Expand Data, maupun mengubah nilai dari atribut Complete. Nilai pada atribut Complete berisikan dua pilihan yaitu DONE dan NOT DONE. Apabila admin memilih DONE maka data akan tersimpan ke dalam *file* dan data akan hilang dari modul ini.



Gambar 3.7 *Flowchart* Modul History

Flowchart History pada Gambar 3.7 untuk menampilkan semua data dari *user* yang memiliki Status Approved dan Complete bernilai DONE. Admin dapat

memilih untuk mengklik *icon* untuk melakukan proses Expand Data. Data pada halaman ini bersifat final dan menjadi data terakhir.

B. Format JSON

File JSON digunakan untuk menampung data yang ditampilkan pada semua modul. Terdapat 2 format JSON yang digunakan selama proses magang berlangsung.

```
{
    "id": "",
    "fullName": "",
    "idType": "",
    "loanProduct": "",
    "loanAmount": "",
    "purposeofLoan": "",
    "tenor": "",
    "loanCreatedDate": "",
    "status": "",
    "complete": "",
    "score": "",
    "riskLevel": "",
    "gender": "",
    "mobilePhone": "",
    "placeofBirth": "",
    "dateofBirth": "",
    "education": "",
    "maritalStatus": "",
    "emailAddress": "",
    "currentAddress": "",
    "province": "",
    "city": "",
    "postal": "",
    "ownershipStatus": "",
    "hasBeen": "",
    "until": "",
    "hasLived": "",
```

Gambar 3.8 Potongan Format JSON Bagian Pertama

Potongan format JSON bagian pertama pada Gambar 3.8 memiliki *fields* yang terkait langsung dengan peminjam. *Field Id* berisikan data utama sebagai *key*

pada JSON. Data Status menjadi kondisi dari suatu data untuk dapat ditampilkan pada beberapa tabel. Data Complete menunjukkan apakah data sudah selesai diproses atau belum.


```
  "spouse_fullName": "",
  "spouse_mobilePhone": "",
  "spouse_placeofBirth": "",
  "spouse_dateofBirth": "",
  "numberofChild": "",
  "spouse_emergencyContact": "",
  "add_address": "",
  "add_province": "",
  "add_city": "",
  "add_postal": "",
  "add_ownershipStatus": "",
  "add_hasBeen": "",
  "add_Until": "",
  "add_hasLived": "",
  "company": "",
  "com_address": "",
  "com_province": "",
  "com_city": "",
  "division": "",
  "duration": "",
  "incomeType": "",
  "bank": "",
  "accountType": "",
  "accountNumber": "",
  "collateralType": ""
}
```


Gambar 3.9 Potongan Format JSON Bagian Kedua

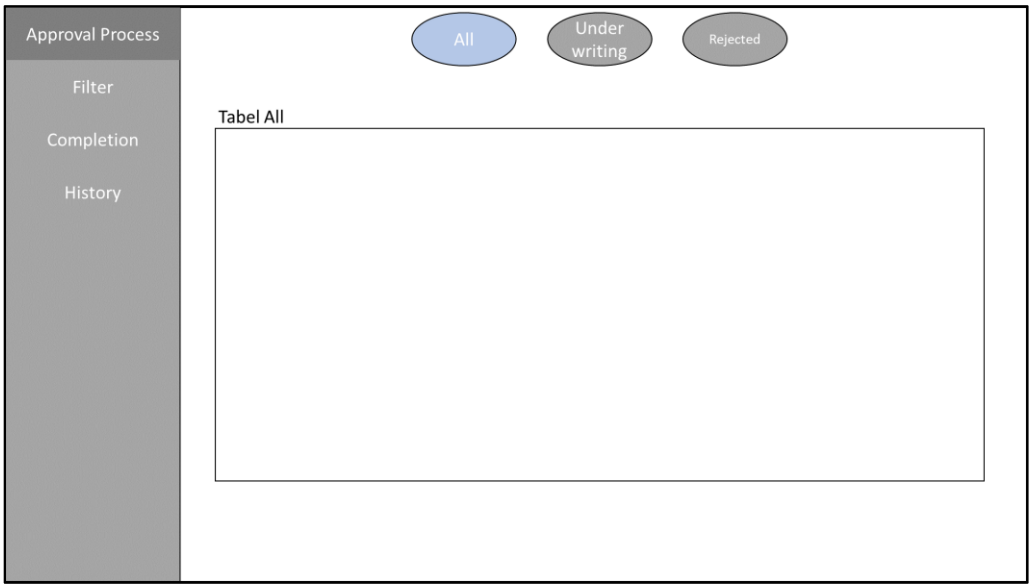
Potongan format JSON bagian kedua pada Gambar 3.9 juga memiliki *fields* yang terkait langsung dengan peminjam. *Field* Spouse berkaitan dengan kehidupan rumah tangga oleh peminjam. *Field* Add berkaitan dengan alamat tambahan bila dimiliki oleh peminjam. *Field* Com berkaitan dengan perusahaan tempat peminjam bekerja.

```
{
    "id": "",
    "field": "",
    "operand": "",
    "value": "",
    "action": "",
    "description": ""
}
```

Gambar 3.10 Format Loan JSON

Pada Gambar 3.10 terdapat *fields* yang terdapat pada Loan.JSON berisikan data yang dapat diisi dengan *dropdown* maupun *textarea*. *Id* merupakan *key* yang bernilai *unique* dan tidak dapat dikosongkan. Format digunakan untuk mengisi data pada filter.

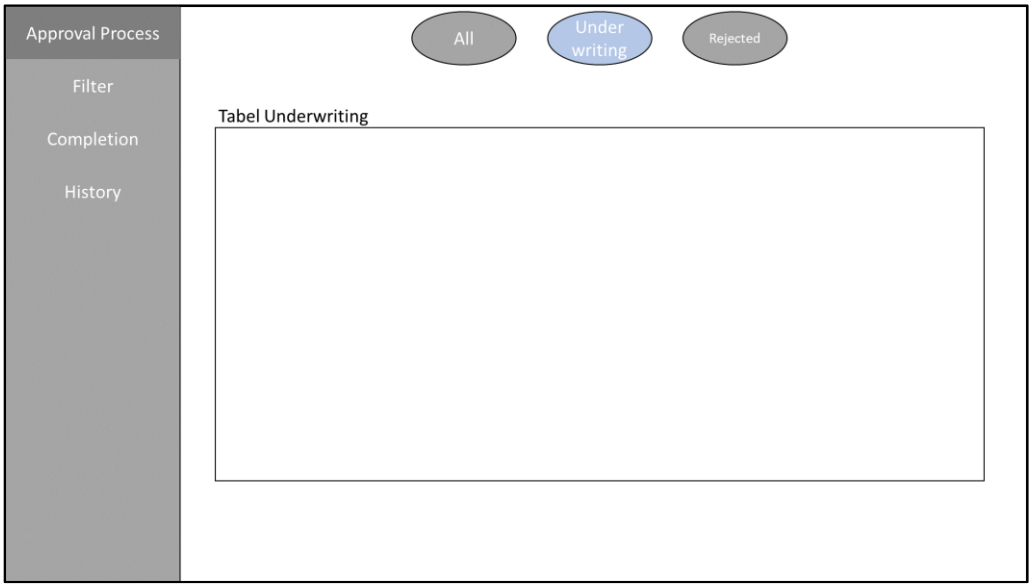
C. Rancangan Antarmuka



Gambar 3.11 Rancangan Antarmuka Modul Approval Process Tabs All

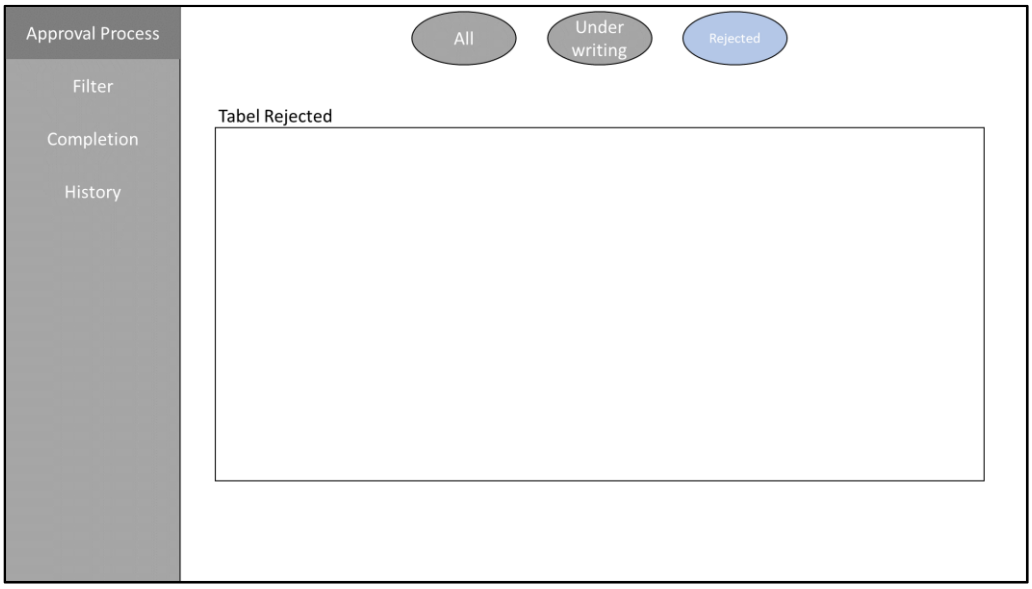
Pada rancangan antarmuka modul Approval Process Tabs All di Gambar 3.11 terdapat tabel yang akan berisikan data dari *user* yang atribut Complete

bernilai NOT DONE. Tabel akan dapat diklik untuk menampilkan data lainnya yang sudah disimpan oleh *user*.



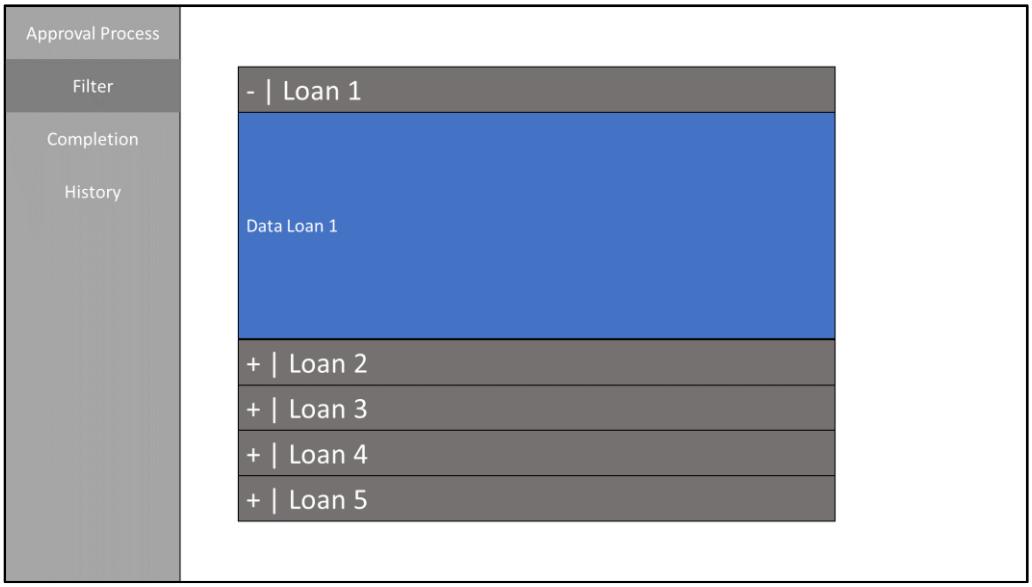
Gambar 3.12 Rancangan Antarmuka Modul Approval Process Tabs Underwriting

Pada rancangan antarmuka modul Approval Process Tabs Underwriting di Gambar 3.12 terdapat Tabel yang akan berisikan data dari *user* yang atribut Status bernilai Underwriting. Tabel akan dapat diklik untuk menampilkan sejumlah filter berdasarkan Loan Products yang dipilih dan apakah data dari *user* sudah memenuhi syarat yang ada.



Gambar 3.13 Rancangan Antarmuka Modul Approval Process Tabs Rejected

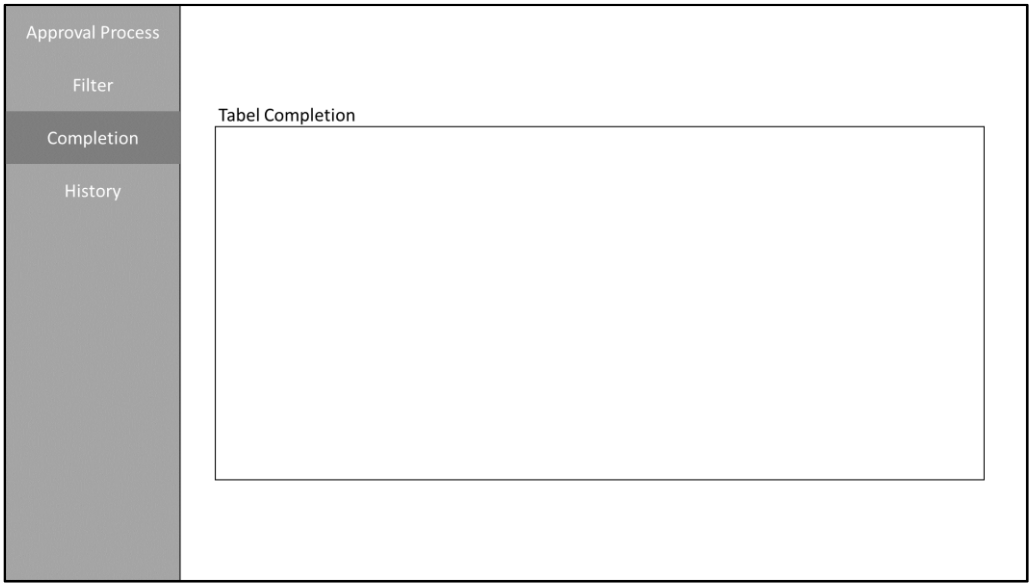
Pada rancangan antarmuka modul Approval Process Tabs Rejected di Gambar 3.13 terdapat tabel yang akan berisikan data dari *user* yang atribut Status bernilai Rejected. Tabel akan dapat diklik untuk menampilkan data lainnya yang sudah disimpan oleh *user*.



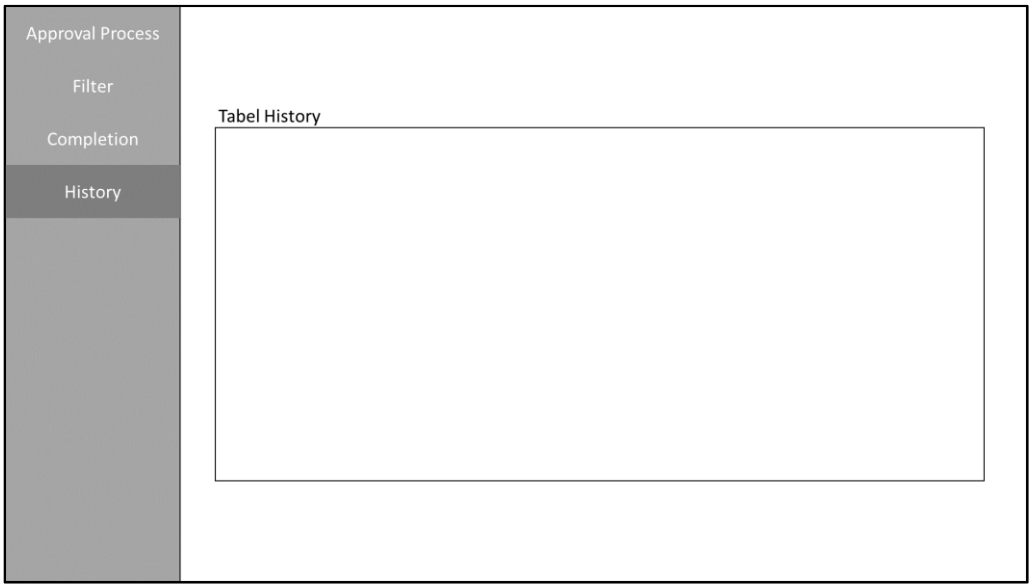
Gambar 3.14 Rancangan Antarmuka Modul Filter

Pada rancangan antarmuka modul Filter di Gambar 3.14 akan berisikan data *Loan Prodcuts* yang dibuat dalam bentuk *Accordion*. Tampilan awal dari modul ini adalah tertutup dan akan ditampilkan apabila diklik.

Pada rancangan antarmuka modul Completion di Gambar 3.15 terdapat tabel yang akan berisikan data dari *user* yang atribut Status bernilai Approved dan Complete bernilai NOT DONE. Tabel akan dapat diklik untuk menampilkan data lainnya yang sudah disimpan oleh *user*.



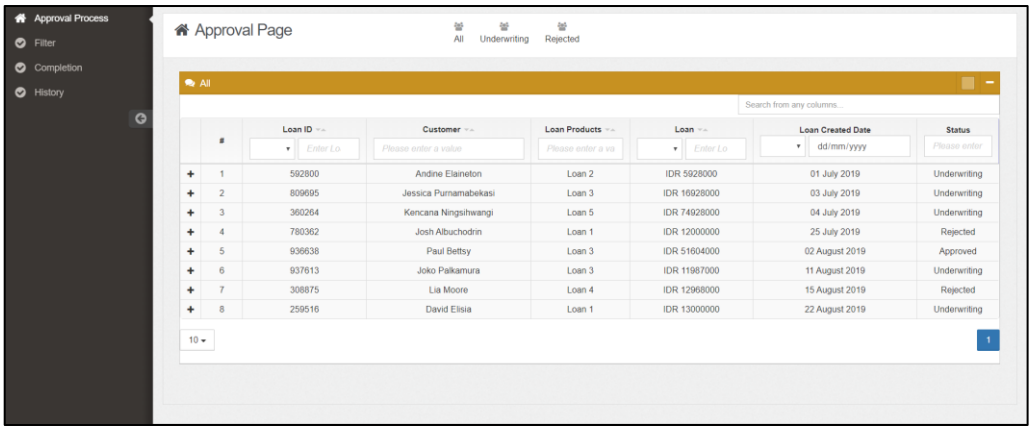
Gambar 3.15 Rancangan Antarmuka Modul Completion



Gambar 3.16 Rancangan Antarmuka Modul History

Pada rancangan antarmuka modul History di Gambar 3.16 terdapat tabel yang akan berisikan data dari *user* yang atribut Status bernilai Approved dan Complete bernilai DONE!. Tabel akan dapat diklik untuk menampilkan data lainnya yang sudah disimpan oleh *user*.

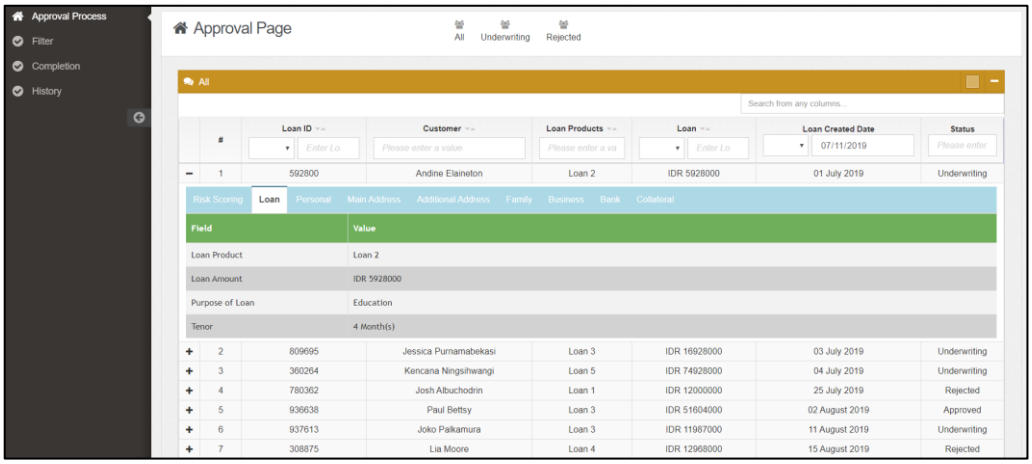
D. Implementasi



Gambar 3.17 Implementasi Modul Approval Process Tabs All

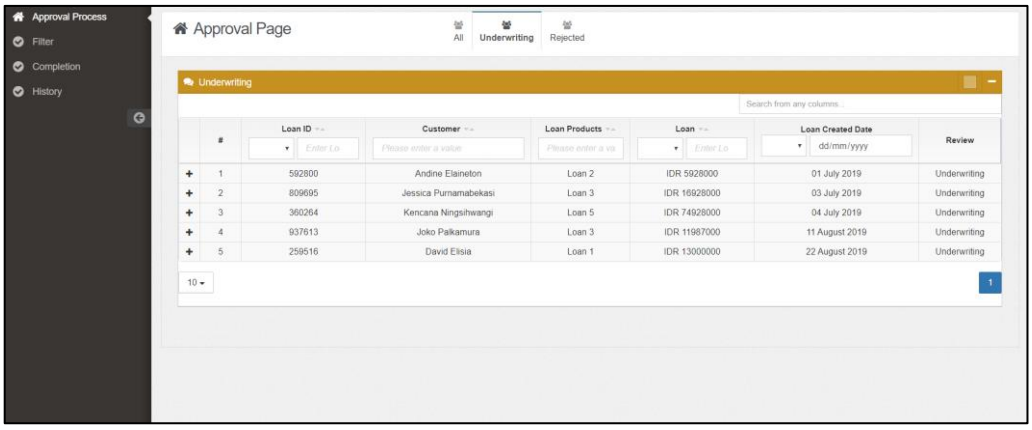
Pada implementasi modul Approval Process Tabs All, di Gambar 3.17 terdapat tabel berisikan data dari *user* yang sudah diisi sebelumnya pada modul

Input With Form yang memiliki nilai pada Status berupa Underwriting, Approved, dan Rejected dengan nilai Complete berupa NOT DONE. Tujuan dari tampilan ini adalah untuk menampilkan semua data *Loan* yang ada kecuali bagi data yang sudah selesai diproses. Data diambil dari *file* JSON menggunakan *localhost*.



Gambar 3.18 Implementasi Modul Approval Process Tabs All Expand Data

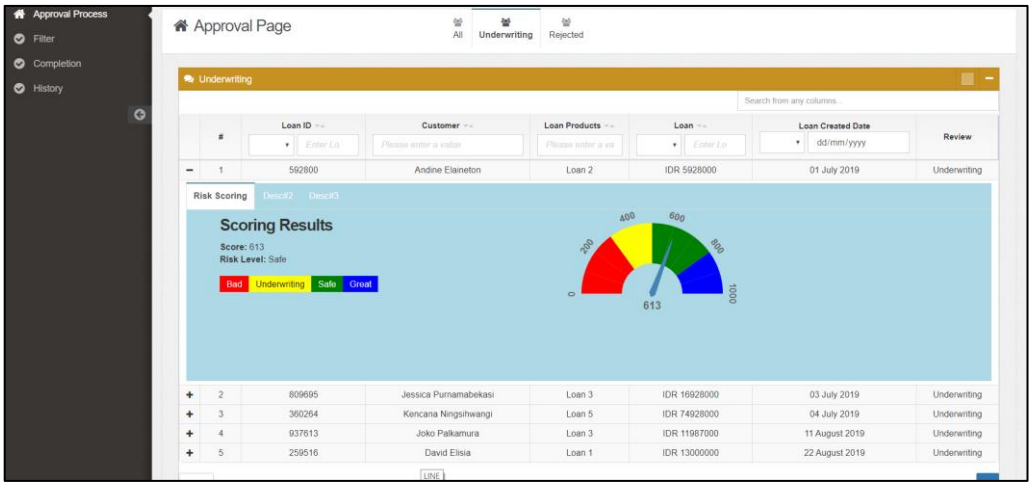
Pada implementasi modul All fitur Expand Data di Gambar 3.18 menampilkan data tambahan oleh *user* yang tidak ditampilkan pada *row* utama. Apabila diklik *icon* yang ada, maka akan muncul baris tambahan berisikan data tersebut. Digunakan *Tabs* untuk memisahkan data apabila terdapat banyak data sehingga admin tidak perlu melakukan *scroll*. Untuk *styling* dibuat tabel yang menampung data sehingga lebih rapi.



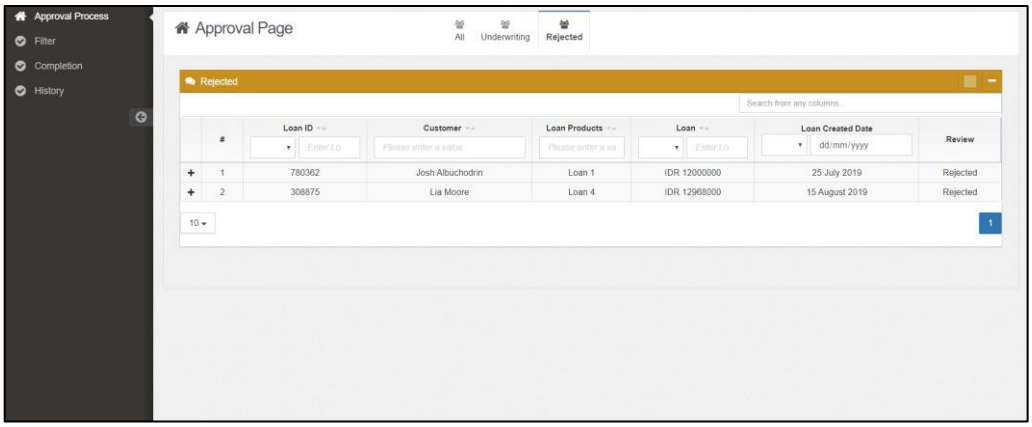
Gambar 3.19 Implementasi Modul Approval Process Tabs Underwriting

Pada implementasi modul Underwriting di Gambar 3.19 akan ditampilkan data yang memiliki nilai Underwriting pada atribut Status. Tujuan dari tampilan ini adalah untuk memilih status baru pada data sesuai dengan filter. Data pada atribut Status dapat diubah menjadi Approved ataupun Rejected sehingga data tersebut akan menghilang dari tampilan modul Underwriting.

Pada implementasi modul Underwriting Expand Data di Gambar 3.20 ditampilkan data tersembunyi apabila diklik pada *icon*. Data ini berupa deskripsi lebih lanjut mengenai kecocokan filter dengan data pada *user*. Selain itu juga terdapat *Speedometer* sebagai nilai skala dari kecocokan tersebut.



Gambar 3.20 Implementasi Modul Approval Process Tabs Underwriting Expand Data

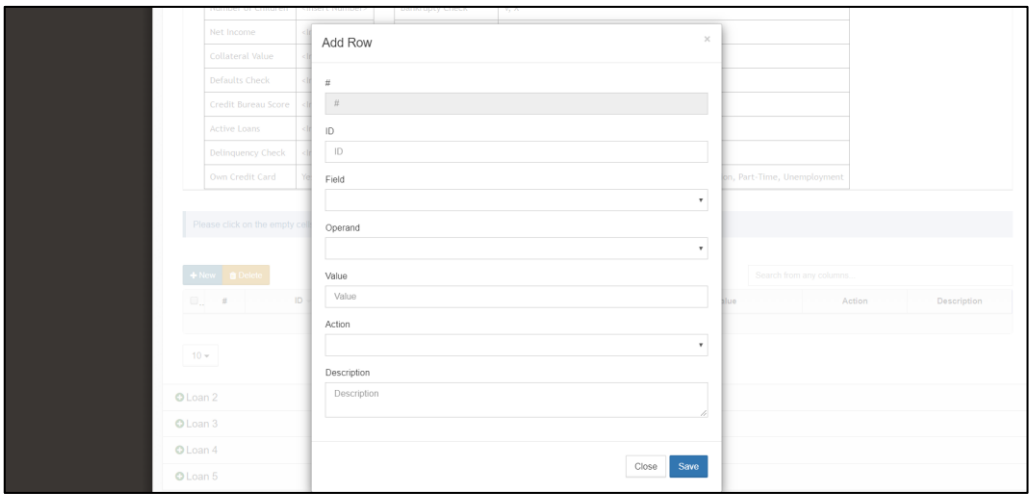


Gambar 3.21 Implementasi Modul Approval Process Tabs Rejected

Pada implementasi modul Rejected di Gambar 3.21 akan ditampilkan data yang memiliki nilai Rejected pada atribut Status. Data ini merupakan data yang ditolak dan tidak akan diproses lebih lanjut karena tidak memenuhi syarat pada filter.

Gambar 3.22 Implementasi Modul Filter Accordion

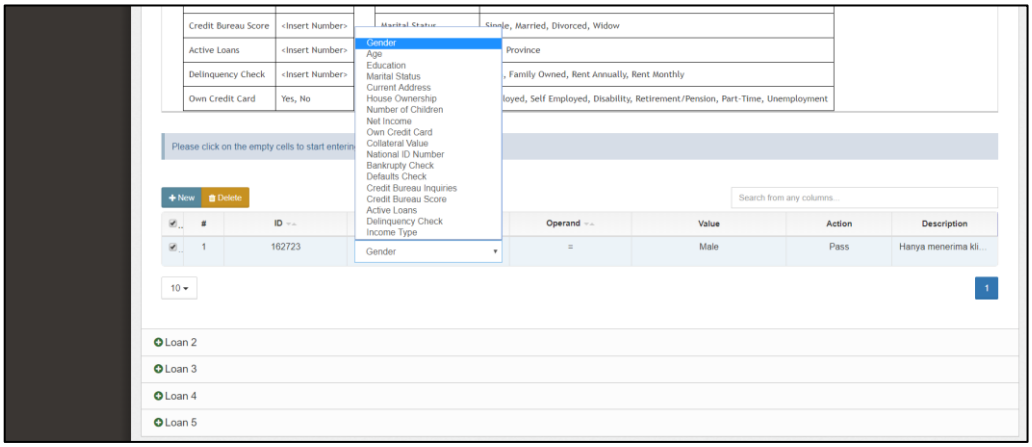
Pada implementasi modul Filter Accordion di Gambar 3.22 menampilkan semua Loan Products yang ada. Digunakan *Accordion* untuk hanya memunculkan 1 Loan Products dalam 1 waktu. Mode *default*nya adalah semua tertutup. Apabila diklik, maka akan muncul detail dari *Loan Products* yang dipilih beserta tabel dari filter yang dimiliki oleh Loan Products tersebut.



Gambar 3.23 Implementasi Modul Filter Add Row

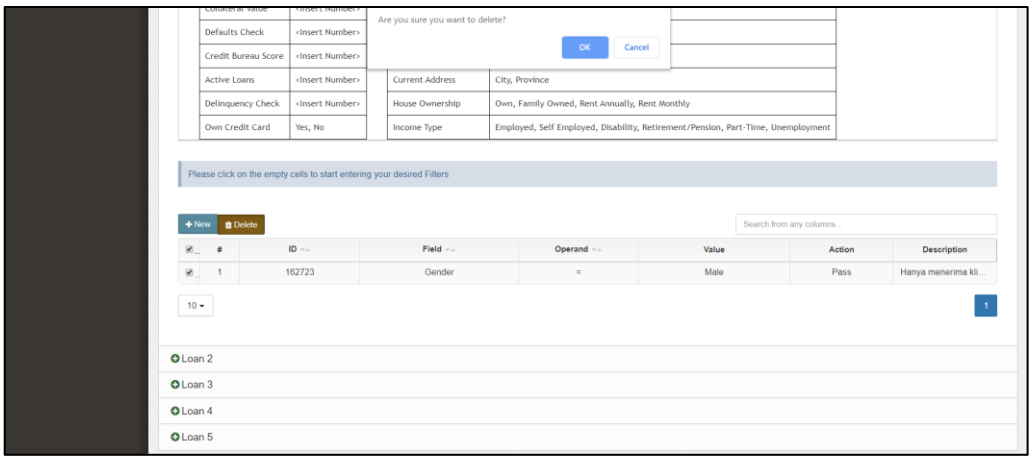
Pada implementasi modul Filter Add Row di Gambar 3.23 akan menambah data pada filter dengan mengisikan sejumlah data sesuai dengan *fields* yang ada. *Id* menjadi *key* sehingga tidak dapat *NULL* dan tidak dapat bernilai sama. Apabila data

baru dikonfirmasi, maka akan tersimpan ke dalam DB.json pada Loan Products masing-masing.



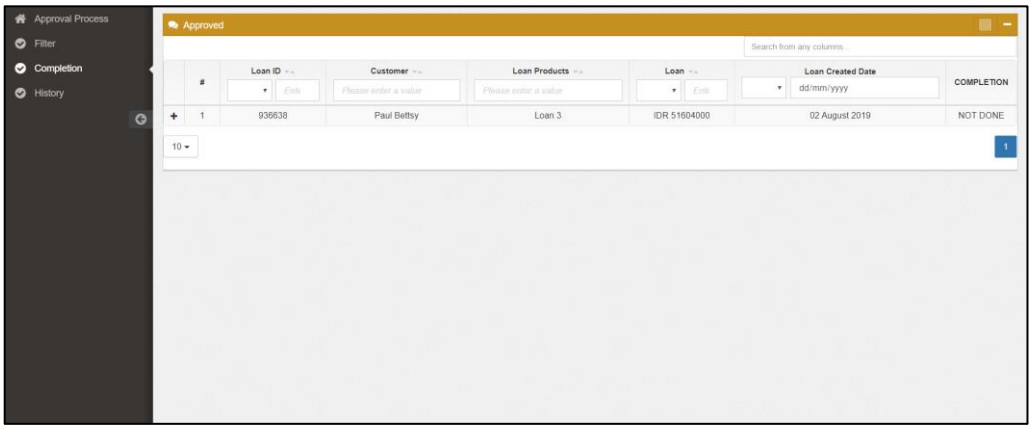
Gambar 3.24 Implementasi Modul Filter Edit Row

Pada implementasi modul Filter Edit Row di Gambar 3.24 akan mengubah data pada filter dengan mengklik *cell* yang berubah. Terdapat tabel yang berisikan panduan dalam mengisi filter. Tidak semua data pada *row* harus diubah. Penyimpanan data dilakukan dengan mode blur yaitu dengan mengklik area di sekitar *textarea*. Data yang diubah akan terupdate pada DB.json. Nilai baru akan ditampilkan ketika halaman di-*refresh*.



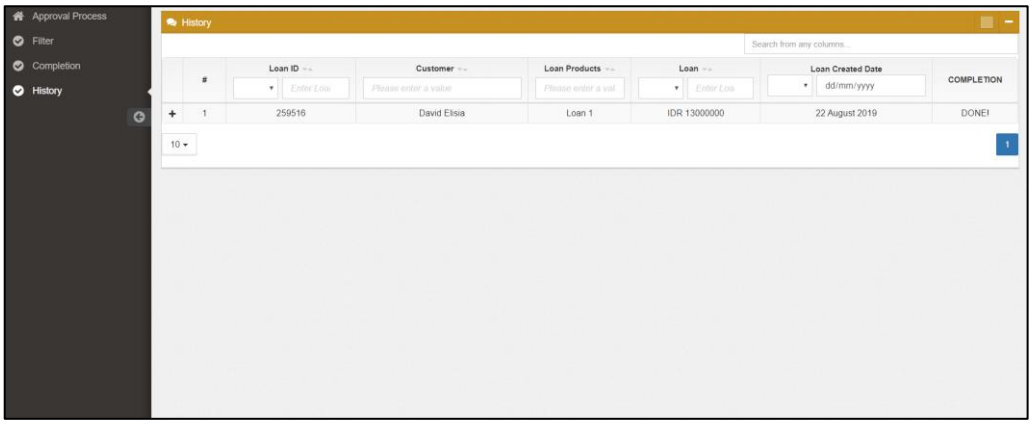
Gambar 3.25 Implementasi Modul Filter Delete Row

Pada implementasi modul Filter Delete Row di Gambar 3.25 akan menghapus data pada filter dengan memilih data mana saja yang akan di-*delete* dengan menggunakan *button checkbox*. Dapat menghapus beberapa data dalam sekali konfirmasi. Apabila dikonfirmasi, maka data dengan *id* serupa akan hilang pada DB.json.



Gambar 3.26 Implementasi Modul Completion

Pada implementasi modul Completion di Gambar 3.26 akan ditampilkan data yang memiliki nilai Approved pada atribut Status. Data yang muncul merupakan data dengan nilai NOT DONE pada atribut Complete dan kemudian dapat diubah menjadi DONE. Tujuan dari tampilan ini adalah untuk menampilkan data yang sudah di-*approved* namun belum ada proses lebih lanjut.



Gambar 3.27 Implementasi Modul History

Pada implementasi modul History di Gambar 3.27 akan ditampilkan data yang memiliki nilai Approved pada atribut Status. Data yang muncul merupakan data dengan nilai DONE pada atribut Complete dan tidak dapat diubah lagi. Tujuan dari tampilan ini adalah untuk mengecek sejarah transaksi data yang sudah di-*approved* yang seluruh prosesnya sudah selesai.

3.4 Kendala yang Ditemukan

Beberapa kendala yang ditemukan pada kerja magang ini adalah sebagai berikut.

1. Data yang diinput oleh *user* ada banyak apabila ditampilkan semuanya sekaligus. Namun apabila tidak dimuat pada tabel dari modul-modul yang terdapat pada modul Action Process maka admin tidak akan tahu mengenai nilai yang tidak ditampilkan tersebut.
2. Tampilan Loan Products yang banyak ditampilkan sekaligus sehingga admin harus *scroll*. Tidak praktis karena akan kesulitan untuk mencari Loan Products yang diinginkan apabila Karena terdapat banyak Loan Products dan pada masing-masing Loan Products memiliki isi berupa tabel dan filter, maka tampilannya akan memanjang ke bawah karena terdapat banyak data, terutama bagi yang menampilkan semuanya sekaligus.
3. Terdapat tugas yang tidak dapat dikerjakan dikarenakan tidak ditemukan penyelesaian mengenai tugas dengan cara yang telah diinstruksikan oleh pembimbing.
4. Kesulitan saat pertama mempelajari ReactJS dikarenakan materi ini sebelumnya tidak pernah dipelajari.

5. Beberapa kali dilakukan pengubahan dari rencana awal pembuatan proyek sehingga hasil yang sudah dibuat menjadi tidak sesuai, maka harus dikerjakan ulang, sehingga beberapa pengerjaan yang dilakukan sebelumnya tidak lagi digunakan.

3.5 Solusi atas Kendala yang Ditemukan

Beberapa solusi atas kendala yang ditemukan pada kerja magang ini adalah sebagai berikut.

1. Menggunakan Expand Data untuk menyembunyikan data lain yang tidak ditampilkan pada bentuk tabel utama dan menampilkannya apabila diklik.
2. Menggunakan *Accordion* untuk hanya memunculkan 1 Loan Products pada suatu waktu sehingga admin dapat langsung memilih melihat Loan Products tertentu.
3. Mencari dan menemukan cara lain dengan hasil serupa dan mendiskusikan kepada pembimbing mengenai cara baru sebagai alternative dari cara lama.
4. Mempelajari ReactJS secara mandiri dengan mencari bahan-bahan yang terkait dengan ReactJS.
5. Selalu berdiskusi dengan pembimbing untuk memastikan arah dan gambaran besar proyek sehingga tidak perlu mengerjakan ulang materi yang sama.