

Penerapan *Financial Technology* pada PT JATEK Servis Indonesia menggunakan *React Native*

Yonatan Abiezer^{#1}, Tjatur Kandaga^{*2}

[#] Program Studi SI Teknik Informatika, Universitas Kristen Maranatha Jl.
Prof. Drg. Surya Sumantri 65 Bandung

¹yoyokaroi22@gmail.com

²tjatur.k@gmail.com

Abstract — *In this era right now, technology is not something that weird. Activities can now be done by using technology because of the simpleness of technology. The biggest technology own by people is smartphone. Because of that loan lending process want to use technology too and want to do it on smartphone too. On the application that we want to made, we want to create an application that can be use to lending money and follow the rules that's been given by otoritas jasa keuangan. Otoritas jasa keuangan is institution that rule money flow in our country. Because of that, every institution that want to set up money centered business, they have to had otoritas jasa keuangan confirm status first. OJK rules that have been fulfilled is the amount that loaned, and then the borrower must be in Indonesia, then the money that been loan must be from the loaner.*

Keywords: *financial, technology, loan, money*

I. PENDAHULUAN

Pada kehidupan jaman sekarang ini, hampir semua aspek kehidupan kita sudah menggunakan teknologi, baik dari penjualan, pendaftaran, pembelian dan lain-lain. Hal ini dikarenakan dimudahkannya pengguna dengan adanya teknologi sehingga tidak perlu terbebani dengan proses-proses yang harus dilalui sebelum adanya teknologi. Semakin berkembangnya teknologi pun, proses-proses bisnis dilakukan setiap saat melalui barang yang ada di genggamannya kita, yaitu *smartphone*. Dengan adanya hal ini, pembayaran dilakukan dengan sangat mudah, hanya dengan menekan satu tombol di *smartphone* kita, kita bisa membayar suatu transaksi.

PT JATEK Service Indonesia menginginkan sebuah aplikasi peminjaman uang yang bebasiskan *smartphone*. Hal ini bisa dilakukan dengan adanya penggunaan *financial technology*. Proses peminjaman ini bisa dilakukan dengan baik hanya melalui aplikasi *smartphone* dan pengguna bisa menggunakan jasanya dengan mudah.

Berdasarkan rumusan masalah yang telah dideskripsikan di atas, tujuan pembuatan aplikasi adalah: Mengimplementasikan *financial technology* pada aplikasi berbasis *mobile* untuk mempermudah proses peminjaman uang sesuai dengan ketentuan dari Otoritas Jasa Keuangan.

II. KAJIAN TEORI

Di dalam penelitian ini, terdapat beberapa teori yang dijadikan landasan dalam penelitian ini. Berikut ini adalah teori-teori yang digunakan:

A. *Financial Technology*

Financial technology, atau teknologi finansial adalah sebuah cara yang digunakan untuk memudahkan dalam mengakses produk-produk yang berhubungan dengan keuangan. Contohnya adalah dalam hal penjualan dan pembelian. Dalam penjualan dan pembelian, dapat menggunakan teknologi yang bisa membuat pengguna membayar tidak dengan uang secara langsung tetapi dengan uang digital. Dengan begitu pengguna bisa dimudahkan dengan adanya teknologi ini. [1]

B. *React*

React adalah *library* yang populer untuk membuat *user interface*. *React* bertugas untuk mengubah pembuatan *user interface* menjadi lebih mudah, lebih mudah digunakan kembali, lebih mudah diubah, lebih mudah dipelihara. [5]

C. React Native

React native adalah *framework* dari *javascript* yang digunakan untuk membuat aplikasi *mobile* pada *platform IOS* dan *android*. *React native* ini dibuat berdasarkan *React*, tetapi tidak mengacu kepada *browser*, melainkan ke *platform mobile*. Sama seperti *react*, *react native* ditulis dari perpaduan *javascript* dan *XML-esque* yang disebut *JSX*. *React native* ini yang menjembatani *native rendering APIs* pada *Objective-C* (untuk *IOS*) dan *Java* (untuk *Android*) sehingga dapat digunakan pada *IOS* dan *Android* sekaligus. *React native* juga bisa digunakan untuk mengakses fungsi dari *platform* contohnya seperti kamera dan *GPS*. [6]

D. Otoritas Jasa Keuangan

Otoritas jasa keuangan atau OJK adalah Lembaga yang independen dan bebas dari campur tangan pihak lain, yang mempunyai fungsi, tugas, dan wewenang pengaturan, pengawasan, pemeriksaan, serta penyidikan sector jasa keuangan di Indonesia.

Tugas OJK adalah mengatur dan mengawasi kegiatan jasa keuangan di sector perbankan, kegiatan jasa keuangan di sector pasar modal, kegiatan jasa keuangan di sector perasuransian, dana pensiun, lembaga pembiayaan, dan lembaga jasa keuangan lainnya. [8]

E. React Redux

React Redux adalah sebuah *library* yang mengandung alat yang bisa memudahkan kita untuk memberikan *store* atau variabel. Dengan adanya *react redux*, pengambilan variabel dari view lain menjadi dipermudah. Variabel yang ada bisa dipanggil seperti berada pada satu halaman yang sana. [5]

III. ANALISIS DAN RANCANGAN SISTEM

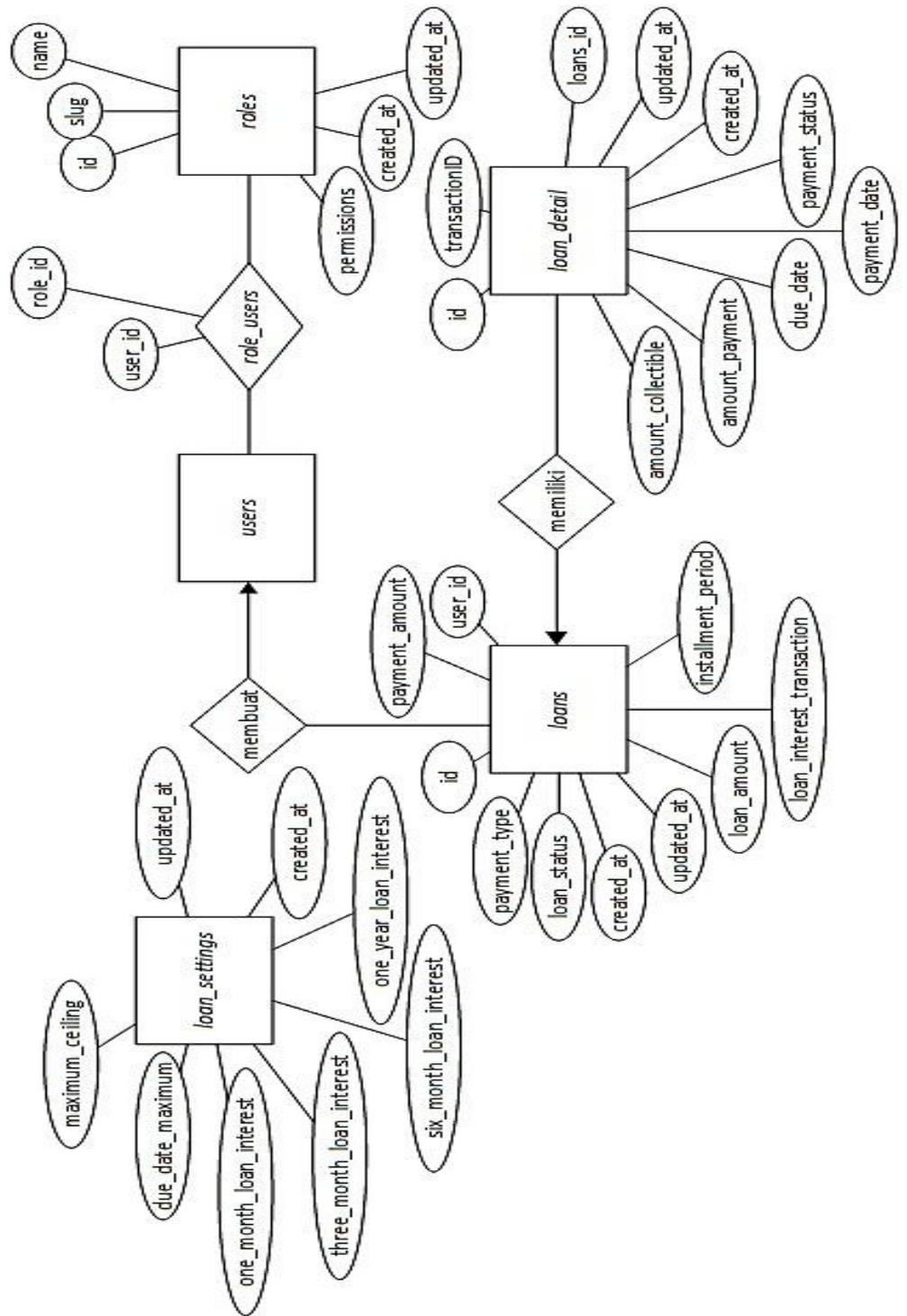
Setelah melakukan wawancara dengan pihak PT. Jatek Servis Indonesia, maka berikut ini adalah informasi yang berhubungan dengan aplikasi peminjaman yang akan dibuat.

Akan ada satu buah *role* yang digunakan untuk aplikasi *mobile* yaitu *customer*. *Customer* akan bisa melakukan peminjaman, pengecekan status pinjaman, pembayaran serta registrasi.

A. Rancangan Basis Data

Berikut adalah rancangan basis data untuk aplikasi *mobile* untuk PT. Jatek Service Indonesia:

1. *Entity relationship diagram*: Gambar 1 adalah rancangan *Entity relationship diagram* (ERD) yang akan digunakan pada aplikasi yang akan dibuat. Entitas *users* dan *roles* memiliki hubungan *many to many* sehingga membuat sebuah entitas baru *role_users* yang berisi *id user* dan juga *id roles*. Lalu entitas *users* juga memiliki hubungan dengan entitas *loans* dimana satu *loan* hanya dimiliki oleh satu *user* sedangkan satu *user* bisa membuat banyak *loan*. Lalu setiap *loan* akan memiliki *loan_detail* yang di dalamnya berisi tanggal cicilan dan jumlah bayar.



Gambar 1 Entity Relationship Diagram

□ *Tabel Users*

Berikut adalah atribut dari table *users*:

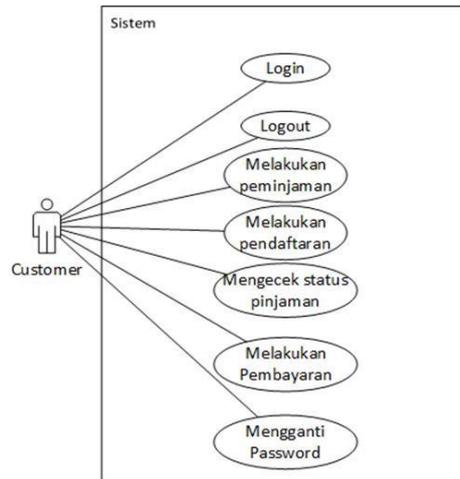
customers	
123 id	
ABC full_name	ABC bank_account_number
ABC birth_place	ABC bank_account_owner_name
🕒 birth_date	ABC office_telephone_number
ABC religion	ABC office_address
ABC email	ABC office_address_province
ABC phone_number	ABC office_address_city
ABC ktp_id_number	ABC office_address_subDistricts
ABC last_education	ABC office_address_urban_village
ABC tribe	ABC office_address_postal_code
ABC dependents	ABC job
ABC telephone_number	ABC job_position
ABC home_address	ABC job_length
ABC home_address_province	ABC job_monthly_salary
ABC home_address_subDistricts	ABC monthly_expenses
ABC home_address_city	ABC KPR_installment
ABC home_address_urban_village	ABC ktp_photo
ABC home_address_postal_code	ABC pay_slip
ABC home_status	ABC photo
ABC close_family_name	ABC loan_type
ABC close_family_phone_number	123 Start_plafon
ABC close_family_address	🕒 created_at
ABC close_family_address_province	🕒 updated_at
ABC close_family_address_city	ABC CreatedBy
ABC close_family_address_subDistricts	ABC UpdatedBy
ABC close_family_address_urban_village	🕒 deleted_at
ABC close_family_postal_code	✓ isSuspend
ABC bank_account_name	123 latest_loan
	123 user_id
	✓ gold_membership
	123 last_paid_off_amount

Gambar 2 Tabel Users

Gambar 2 adalah rancangan atribut dari tabel users. Tabel ini akan berisi semua data dari user yang nantinya akan dimasukkan pada bagian pendaftaran.

B. Use Case

Gambar 3 merupakan rancangan *use case* untuk sistem yang digunakan. Aplikasi akan mempunyai fitur *login*, *logout*, melakukan pinjaman, melakukan pendaftaran, mengecek status pinjaman, melakukan pembayaran, dan mengganti password.

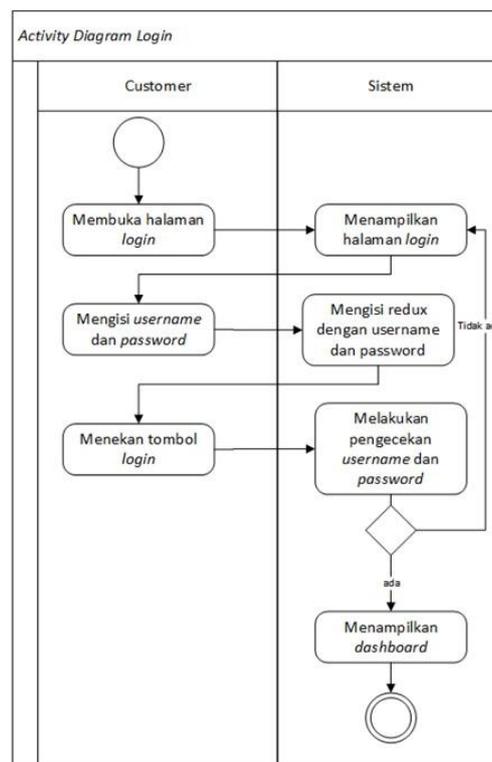


Gambar 3 Use Case Diagram

C. Activity Diagram

Berikut adalah *activity diagram* yang akan digunakan dalam pembuatan aplikasi ini:

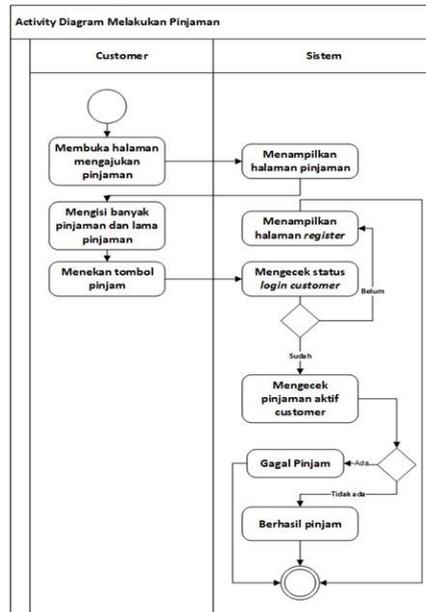
1. *Activity Diagram Login*: Gambar 4 adalah gambar *activity diagram* dari *usecase login*. Proses yang akan terjadi pertama adalah *customer* membuka halaman *login*. Lalu sistem akan menampilkan halaman *login*. Setelah itu, *customer* akan mengisi *username* dan *password* lalu menekan tombol *login*. Sistem akan melakukan pengecekan terhadap *username* dan *password* yang digunakan untuk *login*. Jika *username* dan *password* ditemukan, maka sistem akan menampilkan halaman *dashboard*. Jika tidak ditemukan, maka sistem akan mengeluarkan validasi.



Gambar 4 Activity Diagram Login

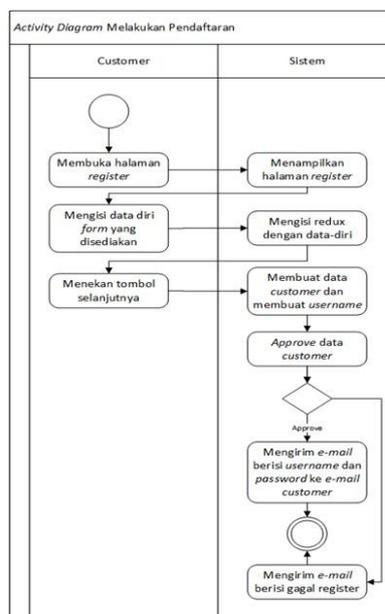
2. *Activity Melakukan Pinjaman*: Gambar 5 adalah gambar *activity diagram* dari *use case* melakukan pinjaman. Saat *customer* memilih pengajuan pinjaman, sistem akan menampilkan halaman pengajuan pinjaman. Dan setelah itu, *customer* bisa memilih banyak pinjaman yang dia ingin pinjam dan lama waktu pengembalian lalu menekan tombol pinjam. Setelah itu sistem akan melakukan pengecekan terhadap *status login customer*, jika sudah *login* peminjaman

akan langsung diajukan untuk diproses lebih lanjut. Jika belum *login customer* akan dilemparkan ke halaman *register* dan melakukan *register* terlebih dahulu. Setelah itu baru peminjaman akan diproses.



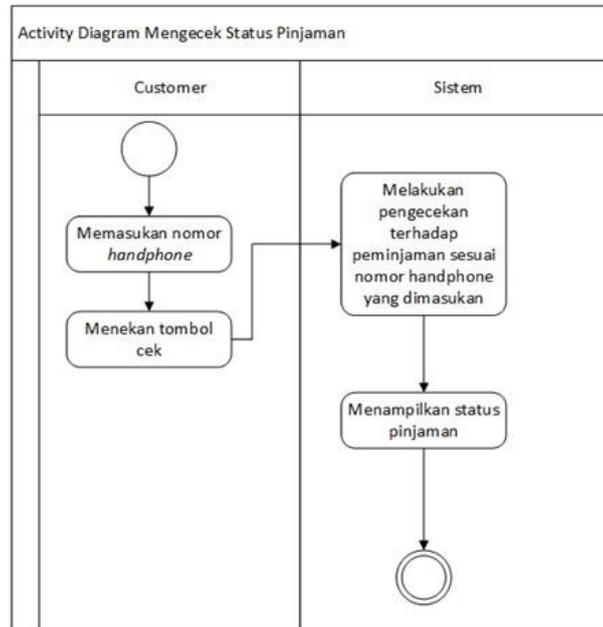
Gambar 5 Activity Melakukan Pinjaman

3. *Activity Melakukan Pendaftaran*: Gambar 6 adalah gambar *activity diagram* dari *use case* melakukan pendaftaran. Halaman pendaftaran hanya akan terbuka jika akan meminjam saat *customer* belum melakukan login. Saat *customer* sudah mengisi semua data dengan benar, maka sistem akan menambahkan data *customer* dan membuat user untuk *customer* tersebut. Jika *register* sudah berhasil, *customer* bisa melihat status pinjaman pada halaman *home*, dan jika sudah disetujui, maka *customer* akan mendapat *email* berupa *username* dan *password* untuk melakukan *login*.



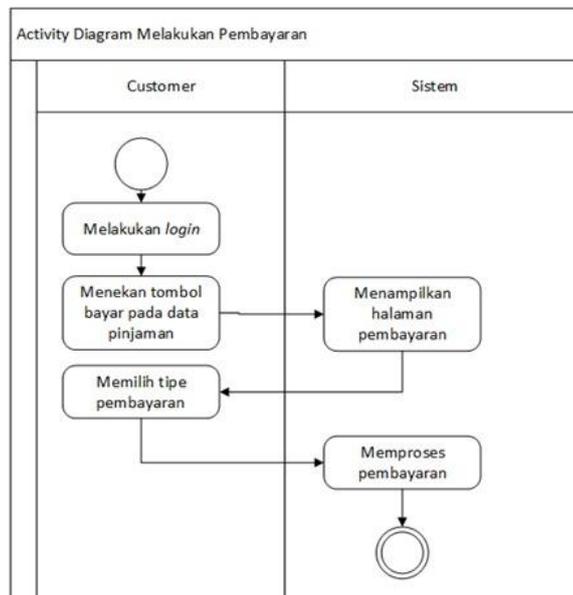
Gambar 6 Activity Melakukan Pendaftaran

4. *Activity Mengecek Status Pinjaman*: Gambar 7 adalah gambar *activity diagram* dari *usecase* mengecek status pinjaman. Saat *customer* memasukkan nomor *handphone* atau *email* dan menekan tombol cek, maka sistem akan mencari dan menampilkan status dari nomor *handphone* atau *email* yang dicari.



Gambar 7 Activity Mengecek Status Pinjaman

5. *Activity Melakukan Pembayaran*: Gambar 8 adalah *activity diagram* dari *usecase* melakukan pembayaran. Saat *customer* sudah melakukan login dan memiliki pinjaman aktif, maka akan ada tombol bayar. Saat tombol bayar ditekan, sistem akan menampilkan pilihan pembayaran. *Customer* lalu memilih jenis pembayaran dan melakukan pembayaran. Setelah itu sistem akan memproses pembayaran tersebut.

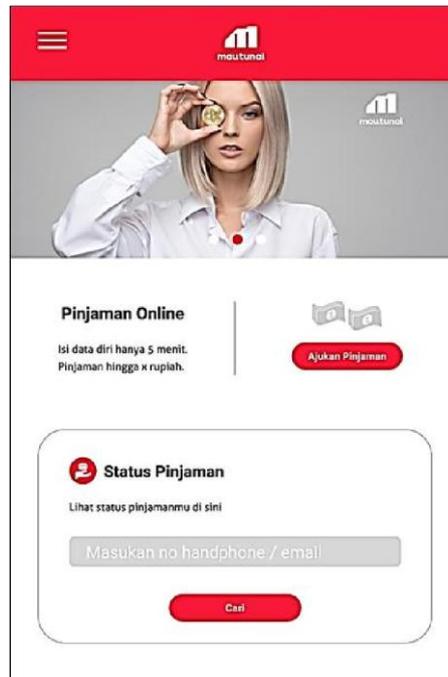


Gambar 8 Activity Melakukan Pembayaran

IV. IMPLEMENTASI

A. Implementasi Tampilan Home

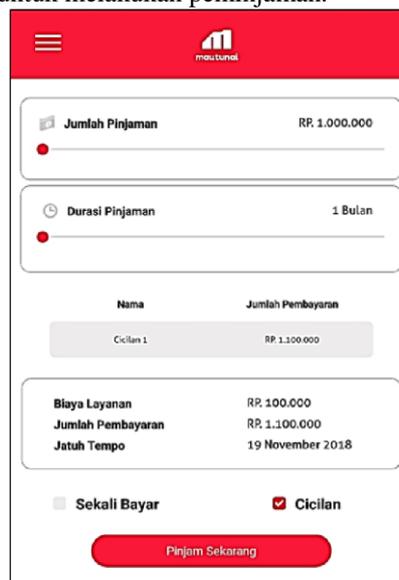
Gambar 9 adalah implementasi dari tampilan *home* yang sudah dirancang. Ada tombol ajukan pinjaman untuk melakukan pinjaman. Di halaman ini ada pula ada *form* untuk melihat status pinjaman berdasarkan nomor *handphone* atau email.



Gambar 9 Implementasi Tampilan *Home*

B. Implementasi Tampilan Ajukan Pinjaman

Gambar 10 adalah gambar dari hasil implementasi tampilan ajukan pinjaman yang sudah dibuat. Pada halaman ini akan ada *scrollbar* untuk memilih jumlah pinjaman dan juga lama pinjaman. Lalu di bawahnya akan ada tabel untuk memperlihatkan cicilan tiap bulannya dan juga tanggal-tanggal penting untuk pembayaran. Ada juga *checkbox* untuk pilihan sekali bayar dan cicilan. Dan ada tombol untuk melakukan peminjaman.



Gambar 10 Implementasi Tampilan Ajukan Pinjaman

```
var data = new FormData();
data.append('payment_type', this.state.bayar);
data.append('loan_status', "New");
data.append('loan_amount', this.state.loanValue);
data.append('customer_id', this.props.user[1]);
data.append('installment_period', this.state.loanDuration);
return fetch('http://mautunai.javacreativ.com/api/v1/loan/register'
,
{method: 'POST',

headers: {
'accept': "application/json",
'content-type': "multipart/form-data",
'Authorization': 'Bearer '+this.props.token,
},
body: data
}
).then( ( response ) => response.json() ).then( ( responseJson ) => {
// Perform success response.
console.log(responseJson)
if(responseJson==="409: Conflict")
{
this.setState({cekLoan:true})
this.refs_scrollView.scrollTo({animated: true}, 0);
}
else
{
this.props.navigation.navigate('Dashboard')
}
}
}
```

Gambar 11 Implementasi Ajukan Pinjaman

Gambar 11 adalah gambar dari implementasi kode untuk ajukan pinjaman. Kode di gambar menunjukkan saat *customer* mengajukan peminjaman, sistem akan melakukan pengecekan menurut *api* pada *server*. Jika pengembalian berupa *conflict* maka peminjaman akan gagal dan akan muncul validasi. Jika pengembaliannya adalah sukses, maka *customer* akan dipindahkan ke halaman *dashboard*.

C. Implementasi Tampilan Login

Gambar 12 adalah gambar dari implementasi halaman *login*. Pada halaman ini ada *form* untuk mengisi *email* dan *password*. Lalu ada tombol *login* untuk melakukan proses *login*.



Gambar 12 Implementasi Tampilan Login

D. Implementasi Tampilan Dashboard

Gambar 13 adalah implementasi dari tampilan dashboard. Pada halaman ini ada tombol untuk membuat pinjaman baru. Lalu ada status untuk pinjaman terbaru, dan jika masih ada pinjaman yang sedang berjalan, akan dikeluarkan tanggal jatuh tempo dan jumlah yang harus dibayarkan. Lalu pada halaman ini ada tombol untuk melakukan pembayaran.



Gambar 13 Implementasi Tampilan Dashboard

E. Implementasi Tampilan History Peminjaman

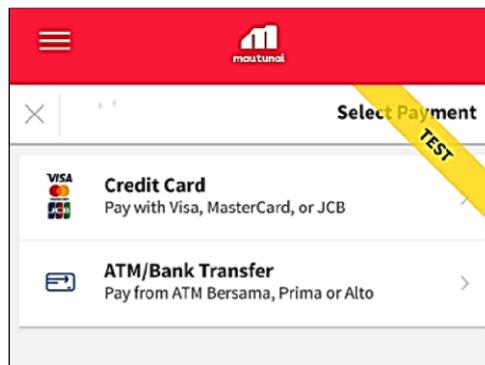
Gambar 14 adalah implementasi dari tampilan *history* peminjaman. Pada halaman ini kita bisa melihat pinjaman-pinjaman yang pernah kita lakukan beserta detail-detailnya.



Gambar 14 Implementasi Tampilan *History* Peminjaman

F. Implementasi Tampilan Pembayaran

Gambar 15 adalah implementasi dari tampilan pembayaran. Pada halaman ini akan ada pilihan antara pembayaran dengan kartu kredit atau pembayaran dengan *transfer* ATM.



Gambar 15 Implementasi Tampilan Pembayaran

G. Implementasi Tampilan Pembayaran Kartu Kredit

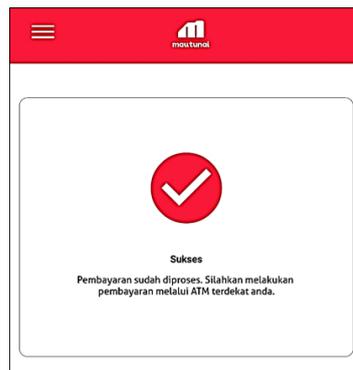
Gambar 16 adalah implementasi dari tampilan pembayaran kartu kredit. Halaman ini hanya akan menampilkan terima kasih jika sudah melakukan pembayaran melalui cara menggunakan kartu kredit.



Gambar 16 Implementasi Tampilan Pembayaran Kartu Kredit

H. Implementasi Tampilan Pembayaran Transfer ATM

Gambar 17 adalah implementasi dari tampilan pembayaran *transfer* ATM. Halaman ini hanya akan menampilkan terima kasih jika sudah melakukan pembayaran melalui cara menggunakan *transfer* ATM.



Gambar 17 Implementasi Tampilan Pembayaran *Transfer* ATM

I. Implementasi Tampilan Register

Gambar 18 adalah implementasi dari tampilan halaman pertama dari *register*. Pada halaman ini *customer* bisa mengisi *email*, nomor telepon, dan nomor KTP. Lalu ada tombol selanjutnya untuk melanjutkan ke halaman kedua *register*. Jika salah satu dari *email*, nomor telepon, atau nomor KTP sudah digunakan, saat menekan selanjutnya akan keluar notifikasi *error* sudah terpakai.

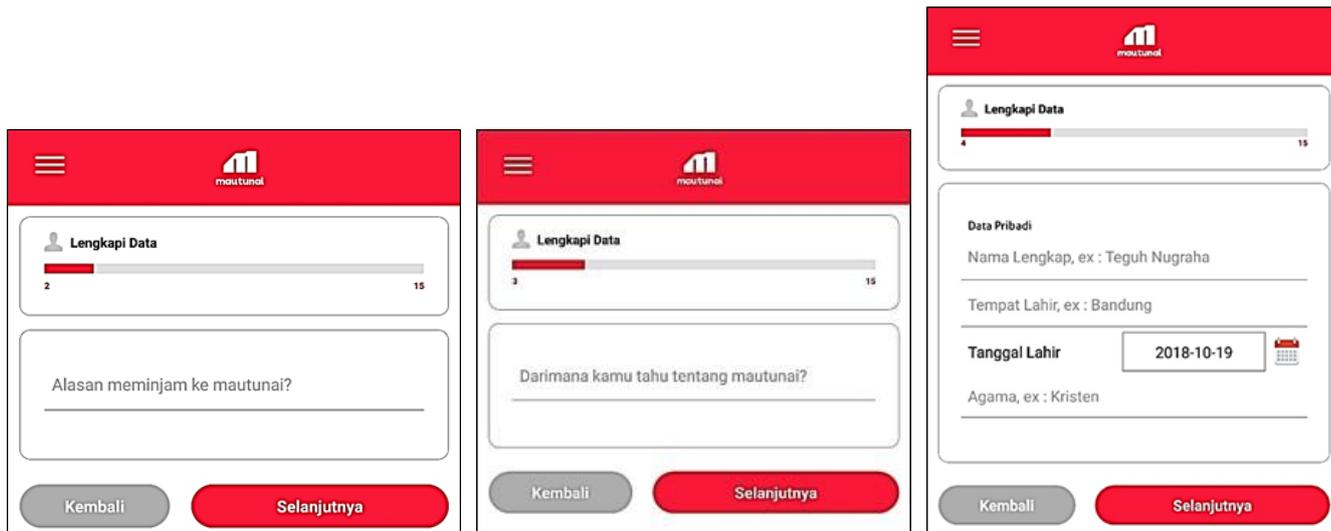


Gambar 18 Implementasi Tampilan *Register*

Gambar 19 adalah implementasi dari tampilan halaman kedua dari *register*. Pada halaman ini *customer* bisa mengisi alasan kenapa ingin meminjam uang. Lalu akan ada tombol selanjutnya untuk melanjutkan ke halaman *register* ketiga. Akan ada tombol kembali juga untuk kembali ke halaman *register* pertama.

Gambar 20 adalah implementasi dari tampilan halaman ketiga dari *register*. Pada halaman ini *customer* bisa mengisi darimana tahu tentang aplikasi ini. Lalu akan ada tombol selanjutnya untuk melanjutkan ke halaman *register* keempat. Akan ada tombol kembali juga untuk kembali ke halaman *register* kedua.

Gambar 21 adalah implementasi dari tampilan halaman keempat dari *register*. Pada halaman ini *customer* bisa mengisi nama lengkap, tempat lahir, tanggal lahir, dan juga agama. Lalu akan ada tombol selanjutnya untuk melanjutkan ke halaman *register* kelima. Akan ada tombol kembali juga untuk kembali ke halaman *register* ketiga.

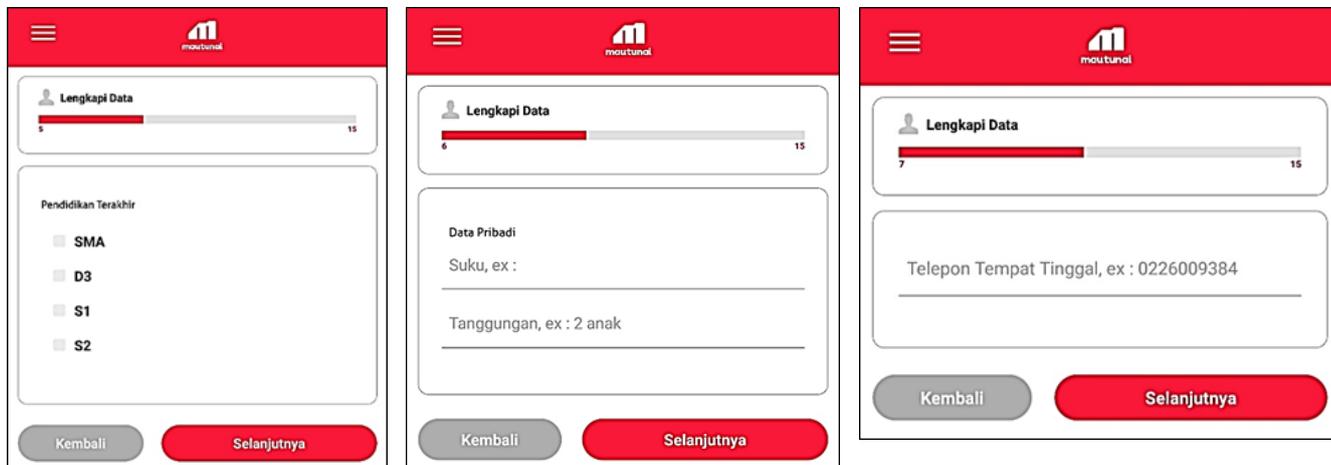


Gambar 19, 20, dan 21 Implementasi Tampilan Register

Gambar 22 adalah implementasi dari tampilan halaman kelima dari *register*. Pada halaman ini *customer* bisa mengisi Pendidikan terakhir yang diterima oleh *customer*. Lalu akan ada tombol selanjutnya untuk melanjutkan ke halaman *register* keenam. Akan ada tombol kembali juga untuk kembali ke halaman *register* keempat.

Gambar 23 adalah implementasi dari tampilan halaman keenam dari *register*. Pada halaman ini *customer* bisa mengisi data suku dan juga tanggungan yang dimiliki *customer*. Lalu akan ada tombol selanjutnya untuk melanjutkan ke halaman *register* ketujuh. Akan ada tombol kembali juga untuk kembali ke halaman *register* kelima.

Gambar 24 adalah implementasi dari tampilan halaman ketujuh dari *register*. Pada halaman ini *customer* bisa mengisi nomor telepon rumah yang ditinggali saat ini. Lalu akan ada tombol selanjutnya untuk melanjutkan ke halaman *register* kedelapan. Akan ada tombol kembali juga untuk kembali ke halaman *register* keenam.

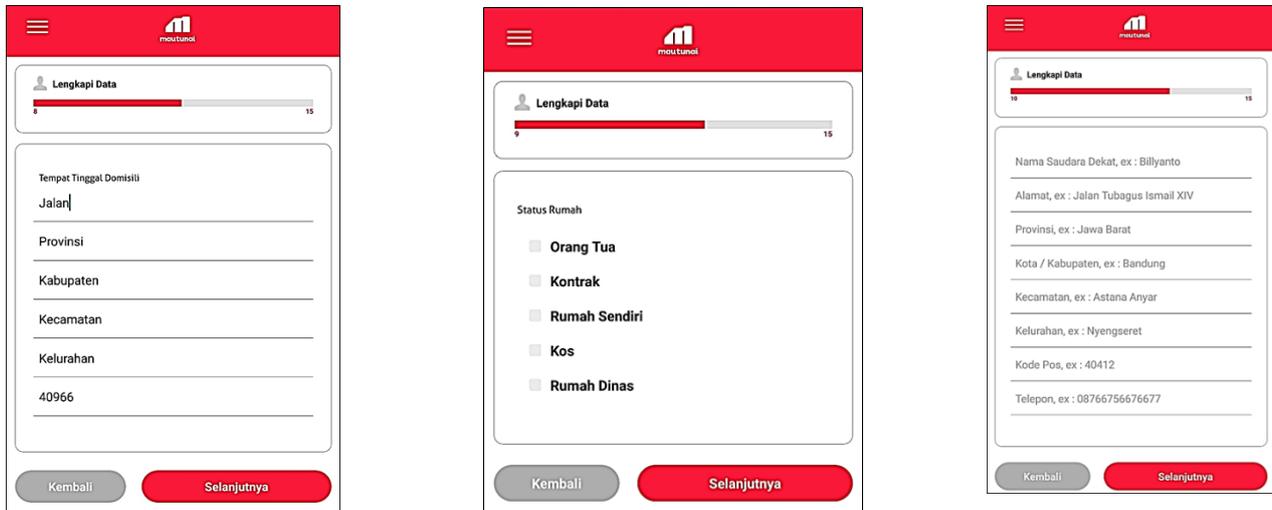


Gambar 22, 23, dan 24 Implementasi Tampilan Register

Gambar 25 adalah implementasi dari tampilan halaman kedelapan dari *register*. Pada halaman ini *customer* bisa mengisi tempat tinggal, mulai dari alamat, provinsi, kabupaten, kecamatan, kelurahan, dan kode pos. Lalu akan ada tombol selanjutnya untuk melanjutkan ke halaman *register* kesembilan. Akan ada tombol kembali juga untuk kembali ke halaman *register* ketujuh.

Gambar 26 adalah implementasi dari tampilan halaman kesembilan dari *register*. Pada halaman ini *customer* bisa mengisi status rumah yang ditinggali sekarang. Lalu akan ada tombol selanjutnya untuk melanjutkan ke halaman *register* kesepuluh. Akan ada tombol kembali juga untuk kembali ke halaman *register* kedelapan.

Gambar 27 adalah implementasi dari tampilan halaman kesepuluh dari *register*. Pada halaman ini *customer* bisa mengisi data nama saudara dekat beserta alamat lengkap dari saudara tersebut. Lalu akan ada tombol selanjutnya untuk melanjutkan ke halaman *register* kesebelas. Akan ada tombol kembali juga untuk kembali ke halaman *register* kesembilan.

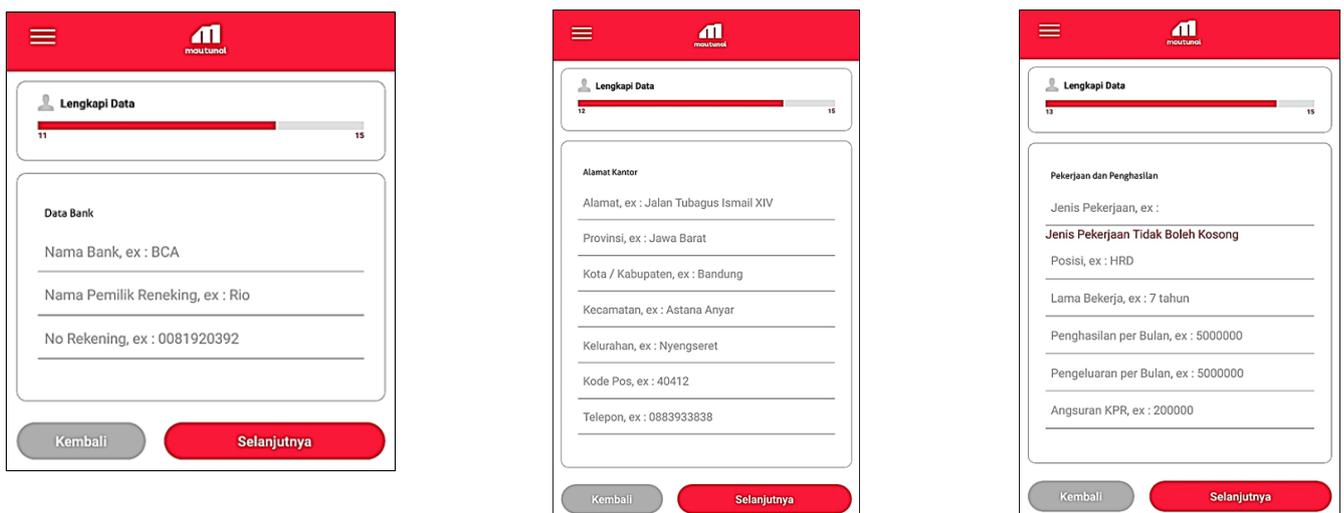


Gambar 25, 26, dan 27 Implementasi Tampilan *Register*

Gambar 28 adalah implementasi dari tampilan halaman kesebelas dari *register*. Pada halaman ini *customer* bisa mengisi data dari nomor rekening *customer*, baik dari nama bank yang digunakan, nama pemilik rekening, dan juga nomor rekening. Lalu akan ada tombol selanjutnya untuk melanjutkan ke halaman *register* keduabelas. Akan ada tombol kembali juga untuk kembali ke halaman *register* kesepuluh.

Gambar 29 adalah implementasi dari tampilan halaman keduabelas dari *register*. Pada halaman ini *customer* bisa mengisi data alamat lengkap dari kantor tempat bekerja. Lalu akan ada tombol selanjutnya untuk melanjutkan ke halaman *register* ketigabelas. Akan ada tombol kembali juga untuk kembali ke halaman *register* kesebelas

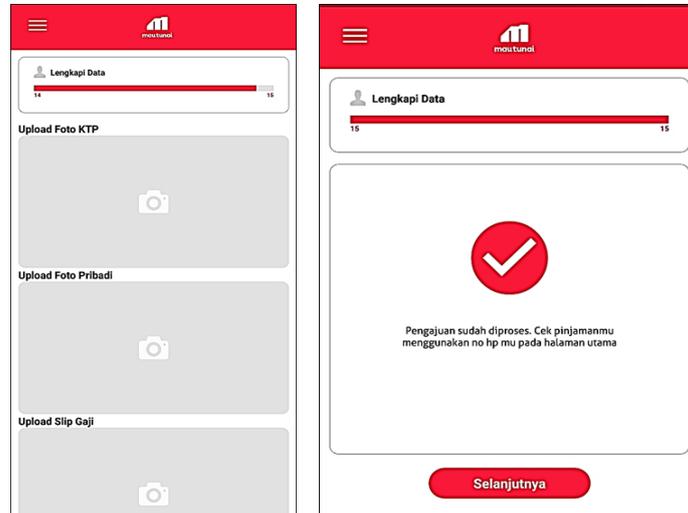
Gambar 30 adalah implementasi dari tampilan halaman ketigabelas dari *register*. Pada halaman ini *customer* bisa mengisi data-data pekerjaan yang sedang dijalani sekarang yaitu jenis pekerjaan, posisi, lama bekerja, penghasilan per bulan, pengeluaran per bulan, lalu angsuran KPR. Lalu akan ada tombol selanjutnya untuk melanjutkan ke halaman *register* keempatbelas. Akan ada tombol kembali juga untuk kembali ke halaman *register* keduabelas.



Gambar 28, 29, dan 30 Implementasi Tampilan *Register*

Gambar 31 adalah implementasi dari tampilan halaman keempatbelas dari *register*. Pada halaman ini *customer* bisa mengisi foto dari KTP, foto probadi, dan foto slip gaji. Lalu akan ada tombol selanjutnya untuk melanjutkan ke halaman *register* kelimabelas. Akan ada tombol kembali juga untuk kembali ke halaman *register* ketigabelas.

Gambar 32 adalah implementasi dari tampilan halaman kesebelas dari *register*. Pada halaman ini hanya menampilkan bahwa registrasi sudah selesai. Akan ada tombol selanjutnya untuk kembali ke halaman *home*.



Gambar 31 dan 32 Implementasi Tampilan *Register*

V. SIMPULAN DAN SARAN

A. Simpulan

Berdasarkan hasil implementasi yang dilakukan, dapat disimpulkan bahwa proses peminjaman uang sesuai dengan ketentuan dari Otoritas Jasa Keuangan dapat dipermudah dengan mengimplementasikan *financial technology* pada aplikasi berbasis mobile. Aplikasi meliputi proses *authentication* (Gambar 12), pengajuan pinjaman (Gambar 10), dan melihat status pinjaman (Gambar 9). Adapun proses penerapan aplikasi menggunakan teknologi *react native*. Dari hasil pengujian dapat disimpulkan aplikasi berjalan dengan baik dan dapat melakukan pembayaran dengan baik.

B. Saran

Saran yang dapat diberikan untuk pengembangan lebih lanjut adalah lebih memperkaya penggunaan api untuk pembayaran, dan juga untuk memperketat proses verifikasi yang sudah berjalan.

DAFTAR PUSTAKA

- [1] S. Chishti, *The FINTECH Book: The Financial Technology Handbook for Investors, Entrepreneurs and Visionaries*, New Delhi: TJ International Ltd, 2016.
- [2] Oriell Incorporated, *Flowcharts: Plain & Simple*, Madison, 2002.
- [3] K. Bittner, *Use Case Modeling*, Boston, 2003.
- [4] S. Bagui, *Database Design Using Entity-Relationship Diagrams*, New York, 2012.
- [5] A. Banks, *Learning React Functional Web Development with React and Redux*, Sebastopol: O'Reilly Media, 2017.
- [6] B. Eisenman, *Learning React Native Building Native Mobile Apps With Javascript*, Sebastopol: O'Reilly Media, 2018.
- [7] "React Native," Facebook Inc., 2018. [Online]. Available: <https://facebook.github.io/react-native/docs/getting-started>.
- [8] O. J. Keuangan, *Mengenal Otoritas Jasa Keuangan dan Industri Jasa Keuangan*, Jakarta: Elex Media, 2010.
- [9] J. William, *Financial Technology: This Book Bundle Includes Fintech and Blockchain*, 2016.
- [10] R. S. Freedman, *Introduction to Financial Technology*, San Diego California: Elsevier Inc, 2006.
- [11] B. Nicoletti, *The Future of FinTech: Integrating Finance and Technology in Financial Services*, Cham Switzerland, 2017.