

## **BAB III**

### **PELAKSANAAN KERJA MAGANG**

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

Pada saat pelaksanaan kegiatan kerja magang dimulai, penulis ditempatkan di departemen IT dari PT Emporia Digital Raya.

Pada minggu pertama pelaksanaan kegiatan kerja magang, penulis diberikan tanggungjawab untuk membantu developer dari produk PT Emporia Digital Raya yaitu IKI Mitra. IKI Mitra merupakan sebuah *platform* elektronik dan solusi bisnis bagi warung, BPR, koperasi, BUMDES, dan komunitas yang disebut dengan sebutan “IKI Mitra”. Aplikasi IKI Mitra memfasilitasi penggunaanya untuk berbisnis dan mendapatkan penghasilan yang menjanjikan dengan 4 pilar, yaitu :

1. Solusi pembelian dan pembayaran PLN, PDAM, pulsa, dan berbagai produk lainnya.
2. Solusi layanan transfer dana ke seluruh bank di Indonesia
3. Solusi pencairan kiriman uang dari luar negeri
4. Solusi pendanaan *peer to peer* (P2P)

Tidak hanya 4 pilar bisnis seperti yang telah disebutkan diatas, IKI Mitra juga memberikan layanan nilai tambah untuk menjamin keberlangsungan dan pengembangan bisnis bagi Mitra IKI, antara lain :

1. Tim perwakilan yang tersebar di setiap wilayah cakupan dan bertugas untuk menyelesaikan permasalahan Mitra IKI
2. Solusi pengembangan jaringan bisnis Mitra. Misalnya, Mitra IKI dapat merekrut *downline* untuk mendapatkan penghasilan tambahan
3. Solusi permodalan bagi Mitra yang ingin mengembangkan bisnisnya
4. Solusi pengajuan pinjaman bagi klien Mitra untuk membantu meningkatkan perekonomian sebagai Mitra.

Pada saat penulis ditempatkan di IKI Mitra, penulis secara langsung bekerja sama dengan salah seorang *software developer* IKI Mitra yaitu Jonathan Atmaja, dibawah pengawasan *group head of information technology* yaitu Bapak Andri Marengga.

Pada minggu kedua pelaksanaan kegiatan kerja magang, penulis mendapatkan perintah dari Bapak Andri Marengga untuk membantu tim *software developer* dari produk PT Emporia Digital Raya lainnya yaitu “IKI Modal”. IKI Modal merupakan layanan *peer-to-peer* lending (Pinjaman Online) berbasis *web* dan *mobile* yang dikembangkan untuk membantu

*channeling* pendanaan mitra serta untuk memberikan penghasilan tambahan untuk mitra sebagai agen pinjaman IKI Modal.

### **3.2 Tugas yang dilakukan**

Tugas yang dilakukan sebagai *software developer* di PT Emporia Digital Raya, antara lain :

1. Membuat fitur baru menggunakan PHP Native & ReactJS.
2. Menyempurnakan fitur yang telah ada pada aplikasi menggunakan PHP Native & ReactJS.
3. Memperbaiki fitur yang telah ada pada aplikasi menggunakan PHP Native & ReactJS.
4. Membuat REST API untuk mendukung aktivitas operasional bisnis menggunakan Lumen PHP Framework dengan metode *Model View Controller (MVC)*.
5. Membantu proses *deployment* dari server *Unit Acceptance Testing (UAT)* ke server *live / production*.

### 3.3 Uraian Pelaksanaan Kerja Magang

**Tabel 3.3.1 Uraian Aktivitas Pelaksanaan Kerja Magang**

| Minggu Ke | Aktivitas   | Tanggal Mulai | Tanggal Selesai |
|-----------|---|---------------|-----------------|
| 1         | Pembuatan fitur layanan bantuan ( <i>customer care</i> )  | 13/07/2020    | 21/07/2020      |
| 2         | Pembuatan REST API untuk <i>import bulk data</i> peminjam V1, V2, V3, V4, dan V5                                      | 22/07/2020    | 30/07/2020      |
| 3         | Pembuatan halaman <i>disburse bulk</i> , perbaikan <i>fitur partial payment</i> dan pembuatan fitur perubahan cicilan | 03/08/2020    | 11/08/2020      |
| 4         | Pembuatan fitur <i>generate va</i> via BNI dan pembuatan fitur <i>callback BNI</i>                                    | 12/08/2020    | 25/08/2020      |
| 5         | Pembuatan fitur <i>auto generate va</i> via Nicepay dan penyempurnaan fitur <i>auto generate va</i> via Nicepay       | 26/08/2020    | 03/09/2020      |
| 6         | Pembuatan REST API “ <i>Get Loan List</i> ” dan “ <i>Pay Selected Installment</i> ”                                   | 04/09/2020    | 14/09/2020      |

| Minggu Ke | Aktivitas  | Tanggal Mulai | Tanggal Selesai |
|-----------|--|---------------|-----------------|
| 7         | Pembuatan fitur <i>generate all VA</i> pada halaman admin IKI Modal dan pembuatan fitur <i>repayment bulk data</i>   | 15/09/2020    | 25/09/2020      |
| 8         | Pembuatan REST API " <i>Privy Register</i> ", REST API " <i>Privy Check Status</i> ", penambahan fitur token pada aplikasi IKI Modal <i>Microsite</i> , dan penambahan fitur daftar PrivyID pada aplikasi IKI Modal Ajukan | 28/09/2020    | 02/10/2020      |

Hari pertama pelaksanaan kegiatan kerja magang yaitu pada tanggal 13 Juli 2020, penulis bertemu dengan *human capital staff* dari PT Anabatic Technologies yang bernama Ibu Amelia Rebecca. Penulis diminta untuk menandatangani kontrak pelaksanaan kerja magang dan diantar ke ruangan kerja penulis yang berada di lantai 11 gedung Graha Anabatic untuk bertemu dengan kepala divisi IT dari PT Emporia Digital Raya yaitu Bapak Andri Marengga. Kemudian penulis dijelaskan mengenai produk-produk bisnis yang dimiliki oleh PT Emporia Digital Raya serta penjelasan detail mengenai produk bisnis IKI Mitra.

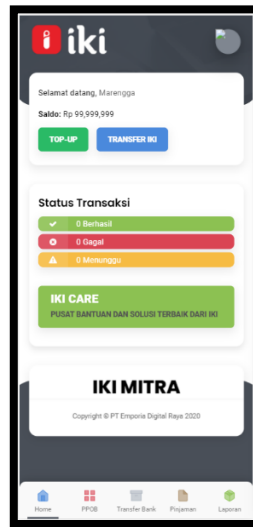
### 3.3.1 Minggu Pertama (Pembuatan Fitur Layanan Bantuan)

Pada minggu pertama, penulis mendapatkan tugas dari Bapak Kenas Revanda selaku *product owner* pada produk IKI Mitra untuk membuat sistem & menu *customer care* yang diberi nama “IKI Care”. Bahasa pemrograman yang digunakan dalam pengembangan fitur *customer care* ini adalah PHP native, HTML, CSS, Javascript, dan juga Whatsapp API.

Bapak Kenas Revanda menjelaskan bahwa fitur pelaporan memiliki 6 topik utama pelaporan masalah antara lain top up, transaksi, login & password, pinjaman, perhitungan laba, dan lainnya. Setelah dijelaskan, penulis dibuatkan change request agar dapat menjadi acuan serta panduan dalam pengerjaan pembuatan fitur / sistem baru.

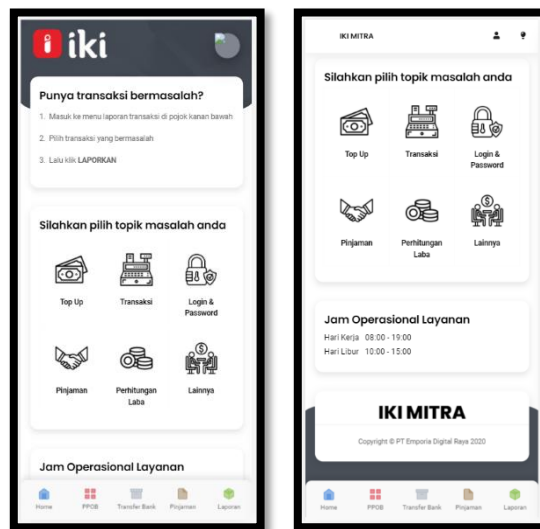
Berikut merupakan hasil pengembangan sistem yang dilakukan oleh penulis :

1. Penambahan fitur IKI Care pada menu utama aplikasi IKI Mitra.



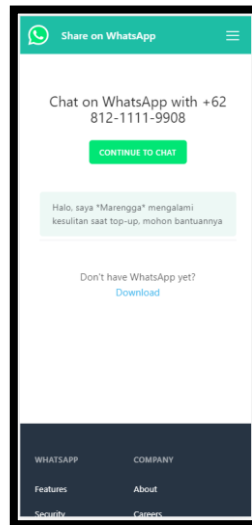
**Gambar 3. 1 Halaman Home Aplikasi IKI Mitra**

2. Apabila *user* melakukan tap pada menu “IKI Care” maka *user* akan di redirect ke page layanan pelaporan masalah seperti gambar dibawah ini :



**Gambar 3. 2 Halaman Menu IKI Care**

3. Apabila *user* melakukan tap pada menu “Top Up”, maka secara otomatis aplikasi akan memanggil Whatsapp API ke nomor *customer care* dari IKI Mitra & secara otomatis pesan Whatsapp tertulis (sesuai dengan pilihan permasalahannya) dalam hal ini adalah “Top Up”. Pesan Whatsapp otomatis yang tertulis yaitu “Hallo, saya \*nama *user*\* mengalami kesulitan saat top-up, mohon bantuannya.”

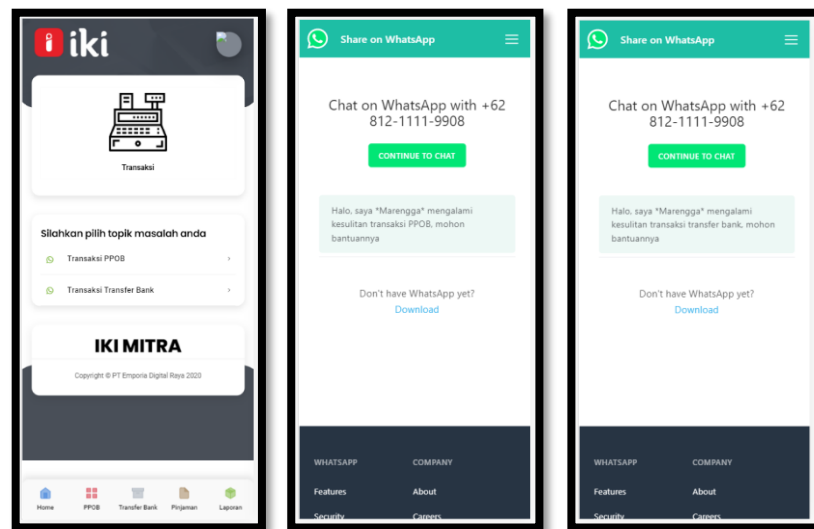


***Gambar 3. 3 Hasil Redirect Menu Top Up***

Pada proses ini, nama *user* diambil dari SESSION yang sedang berlangsung, sehingga secara otomatis aplikasi mengambil dari SESSION yang sedang berlangsung dan menggunakan nama *user* sebagai pelapor.



4. Apabila *user* melakukan tap pada menu “Transaksi” maka secara otomatis aplikasi akan melakukan page redirect ke antarmuka (UI) pemilihan topik masalah yang selanjutnya. Di dalam transaksi terdapat 2 menu, antara lain : transaksi PPOB dan transaksi transfer bank. Berikut merupakan antarmuka yang dimaksud :

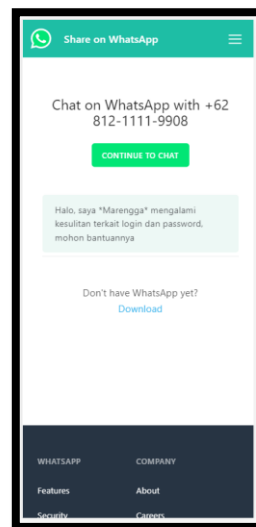


**Gambar 3. 4 Hasil Redirect Menu Transaksi PPOB & Transfer Bank**

Jika *user* melakukan tap pada menu “Transaksi PPOB” maka *user* secara otomatis dinavigasi ke halaman Whatsapp API dengan pesan menyesuaikan menu yang di tap oleh *user*. Isi pesan dari menu “Transaksi PPOB” yaitu “Halo, saya \*nama user\* mengalami kesulitan transaksi PPOB, mohon bantuannya”. Sementara, pesan yang dihasilkan apabila *user* melakukan tap pada menu “Transaksi

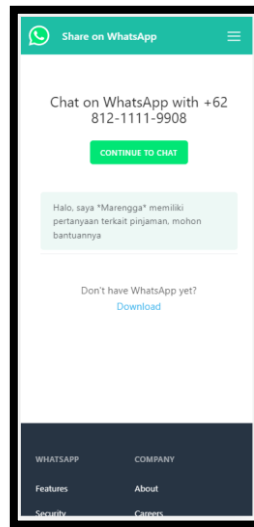
Transfer Bank” yaitu “Halo, nama saya \*nama user\* mengalami kesulitan transaksi transfer bank, mohon bantuannya”.

5. Apabila *user* melakukan tap pada menu “Login & Password” di menu utama dari IKI Care, maka *user* akan langsung dinavigasi ke halaman Whatsapp API dan terdapat auto text yang berisi “Halo, saya \*nama user\* mengalami kesulitan terkait login dan password, mohon bantuannya”. Sebagai informasi, nama dari *user* harus diberikan tanda “ \* “ agar pada saat pesan terkirim ke layanan *customer service* IKI Mitra, nama *user* memiliki bold font style sehingga lebih dapat dilihat dengan jelas oleh admin yang menangani pelaporan dari *user* terkait.



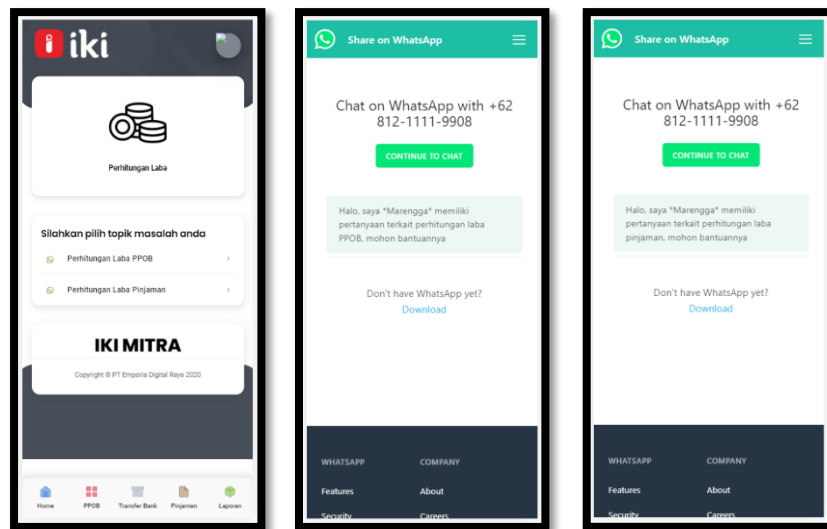
**Gambar 3. 5 Hasil Redirect Menu Login & Password**

6. Jika *user* melakukan tap pada menu “Pinjaman” di menu utama IKI Care, maka *user* akan secara otomatis dinavigasi ke halaman Whatsapp API serta terdapat pesan otomatis “Halo, saya \*nama *user*\* memiliki pertanyaan terkait pinjaman, mohon bantuannya”.



**Gambar 3. 6 Hasil *Redirect* Menu Pinjaman**

7. Apabila *user* melakukan tap pada menu “Perhitungan Laba” di menu IKI Care, maka *user* akan secara otomatis dinavigasi ke halaman lanjutan untuk menu “Perhitungan Laba” adalah sebagai berikut :



**Gambar 3. 7 Hasil Redirect Menu Perhitungan Laba PPOB & Laba Pinjaman**

Apabila *user* melakukan tap pada menu “Perhitungan Laba PPOB” maka *user* akan langsung dinavigasi ke halaman Whatsapp API dan mendapatkan pesan otomatis “Halo, saya \*nama *user*\* memiliki pertanyaan terkait perhitungan laba PPOB, mohon bantuannya”. Begitu juga sebaliknya, *user* akan mendapatkan pesan otomatis yaitu “Halo, saya \*nama *user*\* memiliki pertanyaan terkait perhitungan laba pinjaman, mohon bantuannya”.

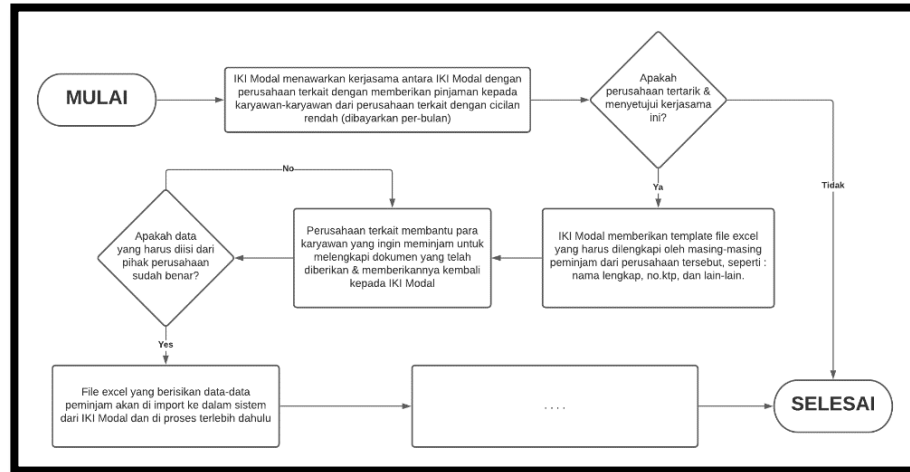
8. Apabila *user* melakukan tap pada menu “Lainnya”. Maka *user* akan langsung dinavigasi ke Whatsapp API dan mendapatkan pesan otomatis yaitu “Halo, saya \*nama *user*\* mohon bantuannya untuk penggunaan IKI Mitra”.

### **3.3.2 Minggu Kedua (Pembuatan REST API untuk *Import Bulk Data* peminjam V1, V2, V3, V4, dan V5)**

Pada minggu kedua pelaksanaan program kerja magang, penulis diminta untuk membantu IKI Modal membuat sebuah sistem baru pada aplikasi IKI Modal yaitu “*Import Bulk Borrower Data*”.

Sistem *import bulk borrower* data ini dibuat berdasarkan alasan yang jelas dari *product owner* & tim IKI Modal. Pada mulanya, IKI Modal ingin mengupgrade status produk dari terdaftar menjadi berizin di Otoritas Jasa Keuangan (OJK). Syarat agar IKI Modal menjadi berizin pada OJK yaitu IKI Modal harus memiliki transaksi sukses sebesar 5.000 transaksi, sementara pada saat itu IKI Modal belum mencapai 5.000 transaksi. Maka dari itu, IKI Modal bekerja sama dengan beberapa perusahaan di pedesaan & non pedesaan untuk memberikan pinjaman dengan cicilan yang meringankan.

Berikut adalah diagram *flowchart* prosedur peminjaman antara *platform* IKI Modal dengan perusahaan yang bekerja sama dengan IKI Modal untuk mengadakan pinjaman dengan cicilan ringan :



**Gambar 3. 8 Flowchart Prosedur Peminjaman di IKI Modal**

Apabila menggunakan sistem peminjaman normal, setiap peminjam harus membuat akun *borrower* di IKI Modal. Namun dengan sistem baru yang dibuat saat ini, para peminjam dimungkinkan untuk meminjam tanpa melewati proses pembuatan akun pribadi namun meminjam menggunakan nama perusahaan tempat mereka bekerja pada sistem IKI Modal.

Pada proses pembuatan sistem baru ini, penulis dipercayakan untuk membuat sistem *import* data dari file .xls, xlsx, dan .csv agar secara otomatis masuk ke dalam *database* IKI Modal. API *import bulk data* yang dibuat oleh penulis terdiri dari 3 bagian utama, yaitu:



**Gambar 3. 9 3 Bagian Utama API Import *Bulk***

Berikut adalah penjelasan detail mengenai 3 bagian utama dari API *import bulk data* :

1. *Import Data*

Pada proses import data, API membaca beberapa variable dari *user* antara lain : file excel, nama admin yang memproses *import data*, nama perusahaan, dan lain-lain.

2. *Data Processing*

Pada proses data processing, API akan mengolah terlebih dahulu file excel yang di import oleh *user*. Pengolahan yang dilakukan di dalam API *import bulk* antara lain yaitu apakah ada data yang tidak bernilai (null values), apakah ada data KTP yang tidak valid, apakah ada data no.rekening yang tidak valid, dan lain-lain. Semua data akan diolah serta diverifikasi kebenarannya di dalam API *import bulk data* ini sehingga apabila data sudah lolos proses, dapat dipastikan data yang masuk ke dalam *database* IKI Modal adalah data bersih.

### 3. *Response*

Pada proses yang terakhir dalam API import *bulk* yaitu “*response*”, API akan memberikan pesan kembalian apabila data sudah selesai di proses / di eksekusi. *Response* kembali dari API digunakan agar *user* mengetahui apa hasil akhir dari proses *import data* terkait. Contoh *response* yang ada di dalam API import *bulk* data, antara lain : pesan *error* karena terdapat cell yang kosong di dalam file, pesan *error* karena KTP tidak valid, pesan *error* karena no.rekening tidak valid, pesan berhasil apabila semua data lolos uji, dan lain-lain.

Setelah proses import *bulk* selesai, sistem selanjutnya dikembangkan oleh *senior software developer* dari IKI Modal yaitu Brinardi Leonardo dan bukan merupakan kewenangan penulis untuk menjelaskan (singkat/detail) dari sistem atau proses selanjutnya.

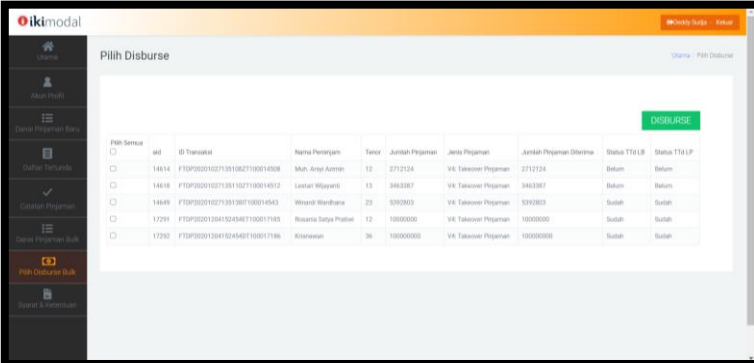
Pembuatan *API import bulk borrower data* ini dibuat menggunakan *Lumen PHP Framework* dengan metode *Model View Controller (MVC)*.



### 3.3.3 Minggu Ketiga

#### 3.3.3.1 Pembuatan Halaman *Disburse bulk*

Halaman *disburse bulk* merupakan halaman dimana pihak pemberi dana (*lender*) dapat melakukan pencairan dana ke pihak peminjam (*borrower*) dimana para peminjam berasal dari perusahaan yang bekerja sama dengan IKI Modal bukan perorangan. Berikut merupakan halaman yang penulis buat guna untuk memenuhi kebutuhan bisnis dari IKI Modal :



| Pilih Service            | id    | ID Transaksi                 | Nama Peminjam       | Tenor | Jumlah Pinjaman | Jenis Pinjaman        | Jumlah Pinjaman Dibayar | Status TGLB | Status TGLP |
|--------------------------|-------|------------------------------|---------------------|-------|-----------------|-----------------------|-------------------------|-------------|-------------|
| <input type="checkbox"/> | 14614 | PTDP202010271331027100014008 | Mah. Anang Arman    | 12    | 2712124         | VA. Tabungan Peminjam | 2712124                 | Belum       | Belum       |
| <input type="checkbox"/> | 14616 | PTDP202010271331027100014012 | Lestari Widyanti    | 18    | 8463367         | VA. Tabungan Peminjam | 8463367                 | Belum       | Belum       |
| <input type="checkbox"/> | 14649 | PTDP202010271331027100014043 | Riswan Wibisono     | 24    | 8292803         | VA. Tabungan Peminjam | 8292803                 | Subah       | Subah       |
| <input type="checkbox"/> | 17291 | PTDP202010410245467020017185 | Riswan Surya Pratno | 12    | 1000000         | VA. Tabungan Peminjam | 1000000                 | Subah       | Subah       |
| <input type="checkbox"/> | 17292 | PTDP202010410245467020017186 | Krisnawan           | 36    | 10000000        | VA. Tabungan Peminjam | 10000000                | Subah       | Subah       |

**Gambar 3. 10 Halaman *Disburse bulk***

Data-data yang ada pada halaman seperti gambar diatas merupakan data yang telah diperiksa dahulu untuk diverifikasi, seperti : apakah *borrower* yang bersangkutan memiliki pinjaman yang macet / tidak lancar? Apabila ya, maka data tidak akan bisa masuk halaman terkait. Pada *disburse bulk* ini, terdapat 3 fitur utama, antara lain :

1. Memilih sebagian pinjaman yang ada pada tabel

Fitur ini dapat dilakukan dengan melakukan *checkbox* pada setiap *checkbox* yang ada pada tabel.

2. Memilih seluruh pinjaman yang ada pada tabel

Fitur ini dapat dilakukan dengan cara melakukan klik pada bagian “Pilih Semua” sehingga semua *checkbox* akan bernilai *true*.

3. Mencairkan dana kepada *borrower* sesuai dengan jumlah pinjamannya masing-masing.

Fitur ini dapat dilakukan dengan cara melakukan *checkbox* dahulu dan kemudian menekan tombol “*Disburse*” yang terletak pada pojok kanan atas.

### **3.3.3.2 Perbaikan Fitur *Partial payment***

Pada tugas ini, penulis dipercayakan untuk memperbaiki fitur *partial payment* sehingga fitur *partial payment* bisa digunakan kembali dengan perhitungan yang benar & membantu berjalannya proses bisnis dari IKI Modal. Fitur *partial payment* pada awalnya dibuat oleh mahasiswa magang yang melaksanakan program kerja magang di PT Emporia Digital Raya sebelumnya. Namun, karena terdapat

ketidaktepatan dari fitur tersebut, penulis diminta untuk menyempurnakan dan memperbaiki kesalahan perhitungan dari fitur *partial payment*. Secara singkat, *partial payment* digunakan untuk melakukan perubahan data cicilan peminjam tertentu. Fitur ini diperuntukkan bagi para peminjam yang dalam perjalanan cicilannya tidak mampu membayar secara keseluruhan sesuai dengan yang harus ia bayarkan. Sebagai contoh, Andi meminjam uang untuk modal usaha Rp 6.000.000 dan dicicil selama 12 bulan. Cicilan pokok yang harus dibayarkan Andi setiap bulannya adalah Rp 500.000. Disamping itu, Andi harus membayar biaya admin sebesar Rp 15.000 dan juga bunga untuk pemberi pinjaman yaitu Rp 15.000. Sehingga, total yang harus dibayarkan Andi setiap bulannya adalah Rp 530.000. Apabila Andi tidak mampu membayar cicilan sebesar Rp 530.000, Andi dapat menghubungi layanan admin dari IKI Modal dan menginformasikan berapa cicilan yang Andi mampu bayarkan agar denda jatuh tempo tidak berjalan khusus untuk cicilan Andi. Apabila Andi menghubungi admin, admin dapat menggunakan fitur *partial payment* untuk membantu Andi melakukan pembayaran tidak sesuai dengan cicilan yang seharusnya ia bayarkan. Syarat pembayaran *partial payment* adalah minimal pembayaran yang disetorkan yaitu biaya admin

ditambahkan dengan bunga pinjaman. Dalam hal ini, pembayaran terkecil yang bisa Andi bayarkan untuk menggunakan fitur *partial payment* adalah sebesar Rp 30.000. Apabila Andi memiliki uang sebesar Rp 50.000, maka Andi akan melunasi biaya admin, bunga pinjaman, serta cicilan pokok sebesar Rp 20.000. Maka, sisa dari cicilan pokok yang Andi belum bayarkan tadi ditambahkan ke cicilan berikutnya.

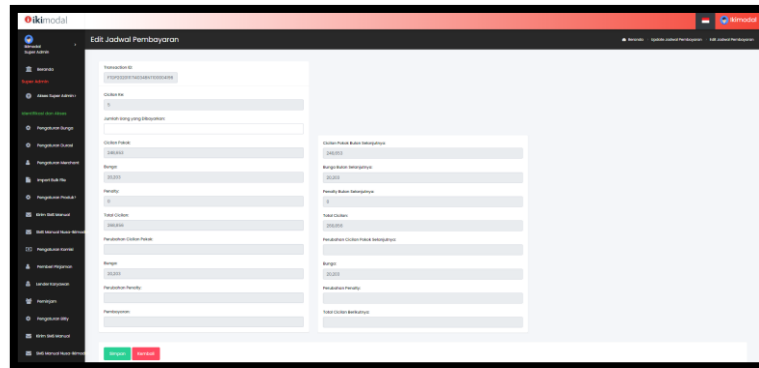
Berikut merupakan *screenshot* halaman *partial payment* :

| ID   | Account ID | Payment ID | Month | Percent | Pay Amount | Origin  | Interest | Admin Fee | Lender Fee | Lender Fee Due | Penalty | Left Amount | Status | Disburse Status | Paid Date           | Lender Receive Date |
|------|------------|------------|-------|---------|------------|---------|----------|-----------|------------|----------------|---------|-------------|--------|-----------------|---------------------|---------------------|
| edit | 00000000   | 0000000001 | 1     | 0       | 200.000    | 200.000 | 0.0000   | 0.0000    | 0.0000     | 0.0000         | 0       | 0           | Paid   | Success         | 2020-10-02 00:00:00 | 2020-10-02 00:00:00 |
| edit | 00000000   | 0000000002 | 2     | 0       | 200.000    | 200.000 | 0.0000   | 0.0000    | 0.0000     | 0.0000         | 0       | 0           | Paid   | Success         | 2020-10-02 00:00:00 | 2020-10-02 00:00:00 |
| edit | 00000000   | 0000000003 | 3     | 0       | 200.000    | 200.000 | 0.0000   | 0.0000    | 0.0000     | 0.0000         | 0       | 0           | Paid   | Success         | 2020-10-02 00:00:00 | 2020-10-02 00:00:00 |
| edit | 00000000   | 0000000004 | 4     | 0       | 200.000    | 200.000 | 0.0000   | 0.0000    | 0.0000     | 0.0000         | 0       | 0           | Paid   | Success         | 2020-10-02 00:00:00 | 2020-10-02 00:00:00 |
| edit | 00000000   | 0000000005 | 5     | 0       | 200.000    | 200.000 | 0.0000   | 0.0000    | 0.0000     | 0.0000         | 0       | 0           | Unpaid | Success         | 2020-10-02 00:00:00 | 2020-10-02 00:00:00 |
| edit | 00000000   | 0000000006 | 6     | 0       | 200.000    | 200.000 | 0.0000   | 0.0000    | 0.0000     | 0.0000         | 0       | 0           | Unpaid | Success         | 2020-10-02 00:00:00 | 2020-10-02 00:00:00 |
| edit | 00000000   | 0000000007 | 7     | 0       | 200.000    | 200.000 | 0.0000   | 0.0000    | 0.0000     | 0.0000         | 0       | 0           | Unpaid | Success         | 2020-10-02 00:00:00 | 2020-10-02 00:00:00 |
| edit | 00000000   | 0000000008 | 8     | 0       | 200.000    | 200.000 | 0.0000   | 0.0000    | 0.0000     | 0.0000         | 0       | 0           | Unpaid | Success         | 2020-10-02 00:00:00 | 2020-10-02 00:00:00 |
| edit | 00000000   | 0000000009 | 9     | 0       | 200.000    | 200.000 | 0.0000   | 0.0000    | 0.0000     | 0.0000         | 0       | 0           | Unpaid | Success         | 2020-10-02 00:00:00 | 2020-10-02 00:00:00 |
| edit | 00000000   | 0000000010 | 10    | 0       | 200.000    | 200.000 | 0.0000   | 0.0000    | 0.0000     | 0.0000         | 0       | 0           | Unpaid | Success         | 2020-10-02 00:00:00 | 2020-10-02 00:00:00 |

**Gambar 3. 11 Halaman *Partial payment***

Pada halaman tersebut, admin dapat melihat seluruh data cicilan peminjam. Apabila admin ingin melakukan *partial payment*, admin dapat klik tombol “edit” di kolom paling kiri pada baris tabel yang ada. Berikut merupakan halaman yang akan muncul

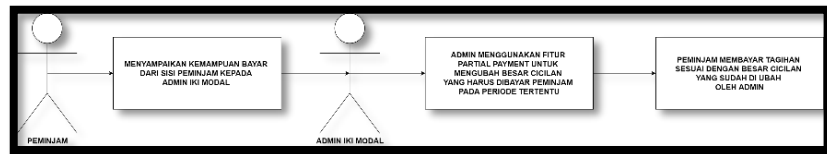
apabila admin melakukan klik pada tombol “*edit*” di kolom paling kiri pada tabel :



**Gambar 3. 12 Halaman *Edit Selected Installment***

Admin hanya dapat melakukan edit data pada input field “Jumlah Uang yang Dibayarkan”. Apabila admin melakukan input jumlah uang dibawah dari biaya admin dan bunga pinjaman, maka secara otomatis halaman tidak akan merespon perubahan tersebut karena syarat melakukan *partial payment* yaitu pembayaran paling kecil adalah biaya admin ditambahkan dengan bunga pinjaman. Apabila admin menginput jumlah uang lebih dari biaya admin dan bunga pinjaman, maka seluruh field akan berubah secara otomatis menyesuaikan dengan perhitungan yang telah ditentukan oleh sistem. Berikut

merupakan *flow* dari prosedur pembayaran cicilan melalui fitur *partial payment* :



**Gambar 3. 13 Flow Prosedur Cicilan *Partial payment***

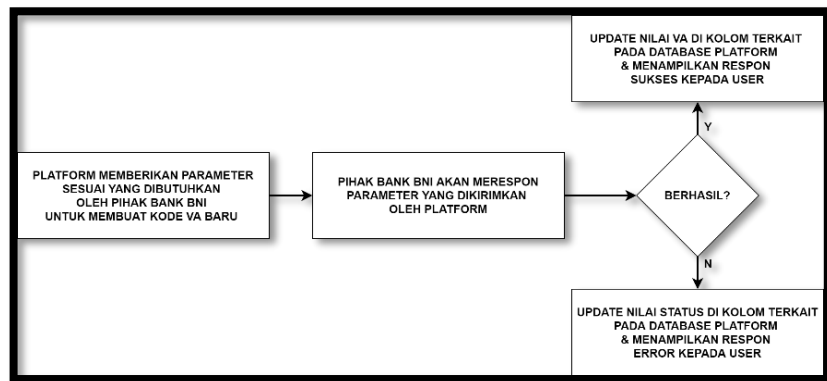
### 3.3.4 Minggu Keempat

#### 3.3.4.1 Pembuatan Fitur *Generate va* via BNI

Pada minggu keempat, penulis dipercayakan untuk membuat fitur pembayaran cicilan peminjam dengan menggunakan *Virtual Account* (VA) dari Bank BNI. Pada sistem sebelumnya, IKI Modal menggunakan NicePay sebagai jasa pembayaran online. Apabila peminjam ingin membayar pinjaman, maka peminjam harus masuk ke dalam akun IKI Modal kemudian memilih bank apa yang ingin dibuatkan kode VA nya. Namun, untuk memenuhi keperluan bisnis, peminjam tidak perlu melakukan pemilihan terhadap bank tertentu lagi, namun hanya ada tombol *generate va* yang akan langsung mengarah ke pembuatan VA Bank BNI. Berikut ini merupakan

screenshot halaman *generate va* via BNI di dalam halaman peminjam (*borrower*) :

Flow secara singkat mengenai cara teknis *generate va* via BNI adalah sebagai berikut :



**Gambar 3. 14 Flow Singkat *Generate va* via BNI**

### 3.3.4.2 Pembuatan Fitur *Callback VA* BNI

Selain pembuatan fitur *generate va* via BNI, penulis juga dipercayakan untuk membuat fitur callback apabila terdapat *borrower* yang membayar cicilan via *Virtual Account (VA)* BNI. Dalam tahap ini, tidak ada tampilan *frontend* karena callback bersifat *backend*.

Secara singkat, flow dari callback VA via BNI adalah sebagai berikut :



**Gambar 3. 15 Flow *Callback VA BNI***

1. Bank BNI memberikan informasi bahwa ada nasabah dari IKI Modal yang membayar cicilan menggunakan Virtual Account BNI. Informasi yang dikirimkan seperti jumlah pembayaran, nomer VA, dan lain-lain.
2. Pihak IKI Modal menerima informasi tersebut dan mencari apakah ada nomer VA tersebut dan status belum terbayar? Apabila ada, maka status akan di update menjadi “Lunas” untuk row data yang memiliki nomer VA tersebut serta berstatus “Belum Lunas”.

### **3.3.5 Minggu Kelima**

#### **3.3.5.1 Pembuatan Fitur *Auto generate va via Nicepay***

Fitur *auto generate* ini sebenarnya hampir sama dengan fitur *generate va* via BNI. Namun, pada server production masih menggunakan *generate va* via Nicepay. Direktur IKI Modal merasa fitur *generate va* belum dapat membantu *customer* mendapatkan *user experience* yang mudah pada saat menggunakan aplikasi IKI Modal. Maka, direktur mengusulkan



untuk dibuatkan *auto generate va* dan tombol “*generate va*” pada halaman *borrower* di hapus. Fitur *auto generate va* akan dijalankan oleh cronjob secara otomatis setiap jam 12 malam.

### **3.3.5.2 Perbaikan Fitur *Auto generate va* via Nicepay**

Perbaikan fitur *auto generate va* via Nicepay dilakukan karena terdapat masalah yang cukup fatal pada saat VA di generate secara otomatis. Alasan perbaikan fitur *auto generate va* tidak dapat disebutkan pada penelitian ini untuk menghargai privasi perusahaan.

## **3.3.6 Minggu Keenam**

### **3.3.6.1 Pembuatan REST API “*Get Loan List*”**

Pembuatan API *Get Loan List* dibuat karena terdapat kebutuhan integrasi sistem antara IKI Modal dengan IKI Mitra. Strategi bisnis dari IKI Modal dan IKI Mitra ini memungkinkan *customer* IKI Mitra melakukan checking terhadap cicilan yang sedang berjalan pada saat ini di *platform* IKI Modal. Pembuatan API ini menggunakan Lumen PHP framework. API ini dilengkapi dengan keamanan HMAC token SHA-256 yang membandingkan token pada saat *customer* melakukan hit pada API *Get Loan List*. Selain itu, API ini dilengkapi dengan sistem

*logging* yang jelas runtutan prosesnya sehingga apabila terjadi sesuatu, semua informasi dapat ditelusuri dengan mudah.

Prosedur teknis tidak dapat digambarkan di dalam *flowchart* karena alasan keamanan layanan pada *platform* IKI Modal.

### **3.3.6.2 Pembuatan REST API “*Pay Selected Installment*”**

Pembuatan API *PaySelectedInstallment* digunakan apabila *customer* IKI Mitra ingin membayar tagihan *platform* IKI Modal menggunakan *platform* IKI Mitra. Secara teknis keamanan dan sistem *logging*, API ini sama dengan API *Get Loan List*. Namun, yang membedakan hanyalah kegunaan fungsionalitas API saja. Prosedur teknis tidak dapat digambarkan di dalam *flowchart* karena alasan keamanan layanan pada *platform* IKI Modal.

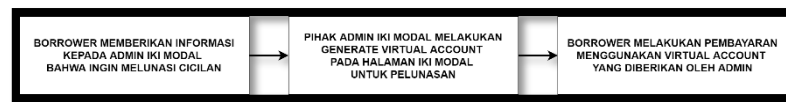
## **3.3.7 Minggu Ketujuh**

### **3.3.7.1 Pembuatan *Fitur Generate All VA* pada Halaman Admin**

Pembuatan fitur *generate all va* pada halaman admin ditujukan apabila ada *customer* yang ingin membayar tagihan secara keseluruhan / melunaskan cicilan. Pada sistem yang ada saat ini, tidak terdapat tombol *generate va all* dikarenakan VA

yang ada pada halaman *borrower* merupakan VA *single* yang hanya berlaku untuk masing-masing cicilan yang akan jatuh tempo.

Prosedur pembayaran pelunasan pada *platform* IKI Modal adalah sebagai berikut :



**Gambar 3. 16** Prosedur Pelunasan Cicilan IKI Modal

1. *Borrower* menghubungi layanan admin IKI Modal apabila ingin melunasi cicilan di IKI Modal.
2. Admin IKI Modal akan melakukan *generate va* dengan total pembayaran sesuai dengan jumlah cicilan yang belum dibayarkan oleh *borrower* dan mengirimkan VA kepada *borrower* saat itu juga.
3. *Borrower* dapat membayar lewat virtual account yang telah diberikan oleh admin IKI Modal.

### 3.3.7.2 Pembuatan Fitur *Repayment bulk data*

Pembuatan fitur *repayment bulk data* ditujukan untuk melakukan proses pembayaran khusus untuk fitur *insert bulk*

data. Isi data merupakan kumpulan data pembayaran yang sudah dilakukan oleh masing-masing *borrower* untuk cicilannya di *platform* IKI Modal.

Penjelasan detail dari fitur *repayment bulk data* tidak dapat dipaparkan secara terperinci karena alasan privasi perusahaan / *platform* yang bersangkutan.

### **3.3.8 Minggu Kedelapan**

#### **3.3.8.1 Pembuatan REST API “*Privy Register*”**

Pada pembuatan REST API registrasi akun *privyID*, penulis menggunakan Lumen PHP Framework untuk mengembangkan REST API. Secara fungsionalitas, REST API ini dibuat dengan basis *microservices* sehingga produk-produk di PT Emporia Digital Raya selain IKI Modal sewaktu-waktu dapat menggunakan *services* yang sama apabila dibutuhkan. Registrasi akun *PrivyID* sendiri digunakan untuk kebutuhan *digital signature*. *Digital signature* biasa digunakan untuk transaksi P2P lending dalam hal persetujuan / kesepakatan antara pihak *lender* dan juga *borrower* terkait regulasi pinjaman, contohnya : bunga, denda, dan lain-lain.

Detail dari REST API tidak dapat dijelaskan pada penelitian ini karena alasan privasi dari PrivyID dan juga *platform* yang bersangkutan.

### **3.3.8.2 Pembuatan REST API “*Privy Check Status*”**

Pembuatan REST API cek status registrasi akun PrivyID ditujukan untuk melakukan checking secara berkala terkait status registrasi akun PrivyID yang dilakukan oleh *borrower*. Secara teknis, apabila registrasi telah berhasil, maka IKI Modal perlu melakukan update data kedalam data *user* di dalam database IKI Modal. Selain itu, REST API cek status registrasi akun PrivyID dibuat dikarenakan belum adanya callback features dari *platform* IKI Modal, maka harus dilakukan cek secara berkala dengan API ini.

Detail dari REST API tidak dapat dijelaskan pada penelitian ini karena alasan privasi dari PrivyID dan juga *platform* yang bersangkutan.

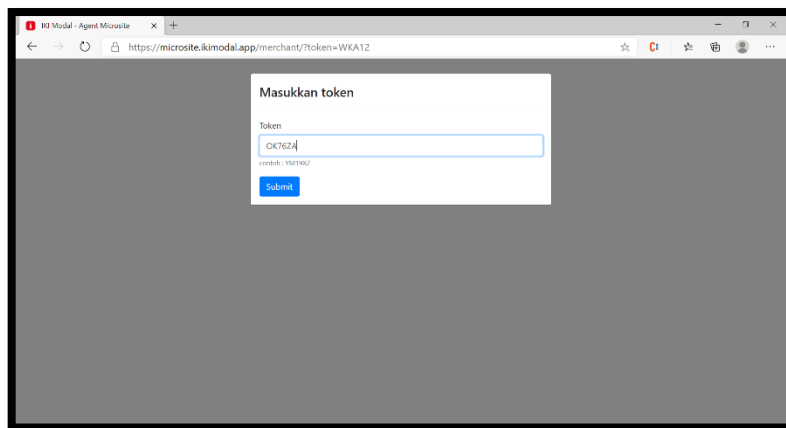
### **3.3.8.3 Penambahan Fitur Cek Token pada Aplikasi IKI Modal**

#### **Microsite**

Aplikasi microsite disempurnakan dengan fitur token pada saat aplikasi akan digunakan karena terdapat perubahan

regulasi dari sistem integrasi IKI Modal dan IKI Mitra. Pada awalnya aplikasi IKI Modal dapat dibuka secara langsung di dalam aplikasi IKI Mitra. Namun, seiring berkembangnya bisnis dari IKI Mitra, perubahan regulasi dilakukan agar tidak mengganggu sistem dari IKI Mitra secara menyeluruh.

Berikut merupakan *screenshot* dari halaman yang penulis kembangkan pada aplikasi microsite IKI Modal :



**Gambar 3. 17 Halaman Microsite**

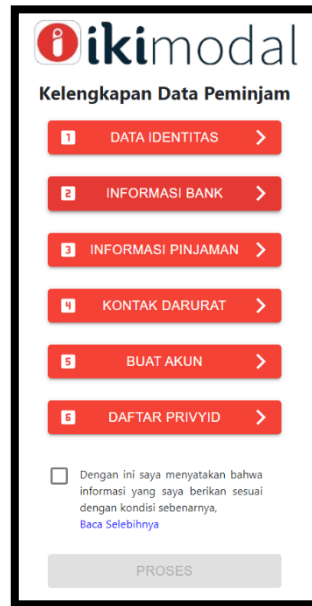
*User* diminta untuk melakukan input token sebelum masuk kedalam aplikasi. Token akan didapatkan *user* dari aplikasi IKI Mitra. Sebelumnya, aplikasi IKI Modal dapat berjalan secara *iframe* di sistem IKI Mitra. Namun sekarang, apabila *borrower* ingin mengajukan pinjaman pada IKI Modal, maka IKI Mitra hanya melakukan generate token dan memunculkan URL yang

dapat diakses oleh *borrower*. Tentunya, *borrower* akan mengakses dari browser secara terpisah. Pengembangan sistem ini dikembangkan dengan ReactJS *Framework*.

#### **3.3.8.4 Penambahan Fitur Daftar PrivyID pada Aplikasi IKI**

##### **Modal Ajukan**

Aplikasi ajukan merupakan aplikasi lanjutan apabila *borrower* sudah melewati tahap pengajuan pinjaman melalui *microsite*. Aplikasi ajukan akan digunakan oleh *borrower* untuk melengkapi data dan juga kebutuhan lainnya yang berhubungan dengan proses administrasi pengajuan pinjaman pada *platform* IKI Modal. Berikut merupakan tampilan yang penulis kembangkan dengan ReactJS *framework* untuk fitur daftar PrivyID :



**Gambar 3. 18 Halaman Ajukan**

Semua data-data yang ada diatas harus diisi dan juga terdapat fitur tambahan yaitu “Daftar PrivyID”. Apabila semua sudah dilengkapi, maka tombol yang berwarna merah akan berubah menjadi hijau dan tombol “Proses” dapat bersifat “*enable*” / dapat ditekan.

### **3.4 Kendala**

Pada saat kegiatan kerja magang berlangsung, satu-satunya kendala yang dialami penulis yaitu kurangnya pengetahuan mengenai *tools* pengembangan sistem terbaru seperti ReactJS, ExpressJS, NodeJS, Golang, dan ReactNative. Pengembangan sistem dengan *tools* tersebut diyakini dapat



mempercepat kinerja dari aplikasi dan juga memberikan efisiensi waktu serta tenaga karena dapat berjalan di OS yang berbeda, contohnya React Native dapat berjalan di iOS dan juga android.

### **3.5 Solusi**

Solusi dari kendala yang dialami oleh penulis pada saat kegiatan kerja magang berlangsung yaitu belajar dengan cepat bahasa pemrograman baru / *development framework* baru dengan cepat sehingga dapat mencapai target perusahaan dalam waktu yang telah ditentukan.