



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB III

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Pelaksanaan kerja magang dilakukan di *Developer Team* sebagai *Software Developer* yang terdiri dari 6 orang. Supervisi dilakukan oleh Bapak Ainul Labib yang menjabat sebagai *IT Analyst* di *Developer Team*. Tugas-tugas diberikan dari supervisi untuk mengembangkan beberapa *web service* dan *website project*, termasuk web *Reconcile Admin*.

#### 3.2 Tugas yang dilakukan

Tugas yang diberikan supervisi berupa pengembangan *project*. Detail tugas yang dikerjakan dapat dilihat pada Tabel 3.1.

Tabel 3.1 Tugas yang dilakukan

Minggu	Deskripsi
1	Pengenalan <i>web service &amp; tools</i> yang digunakan
2	Pengenalan dengan <i>project web service Recharge</i>
3	Pengembangan modul <i>Create Order</i> di <i>web service Recharge</i>
4	Pengembangan modul <i>Notif Payment</i> di <i>web service Recharge</i>
5	Revisi tugas yang dilakukan di <i>project web service Recharge</i>
6	Pengembangan modul web Konfirmasi Pembayaran untuk <i>project web service Recharge</i>
7	Pengembangan <i>project web service Tcash Bill Payment</i>
8	Pengenalan dengan <i>project web Reconcile Admin</i>

Tabel 3.1 Tugas yang dilakukan (Lanjutan)

9	Pengembangan modul <i>Report Rekonsiliasi Virtual Account</i> di <i>project Reconcile Admin</i>
10	Pengembangan modul <i>Report Rekonsiliasi detail</i> di <i>project Reconcile Admin</i>
11	Pengembangan modul <i>Report Rekonsiliasi History</i> di <i>project Reconcile Admin</i>
12	Revisi tugas yang dilakukan di <i>project web Reconcile Admin</i>
13	Revisi tugas yang dilakukan di <i>project web Reconcile Admin</i>
14	Revisi tugas yang dilakukan di <i>project web Reconcile Admin</i>

Penjelasan mengenai *project* yang dikerjakan adalah sebagai berikut.

- 1) *Web service Recharge* merupakan *service* yang menyediakan *function* untuk mengontrol transaksi seperti *Create Order*, *Payment Notification*, dan konfirmasi pembayaran yang terjadi pada *project Recharge*.
- 2) *Web service Tcash Bill Payment* merupakan *service* yang menyediakan *function* untuk mengontrol transaksi yang terjadi pada *project Tcash Bill Payment*.
- 3) *Web Reconcile Admin* merupakan aplikasi web yang digunakan untuk mempermudah kegiatan rekonsiliasi laporan keuangan di PT Inovasi Daya Solusi.

### 3.3 Uraian Pelaksanaan Kerja Magang

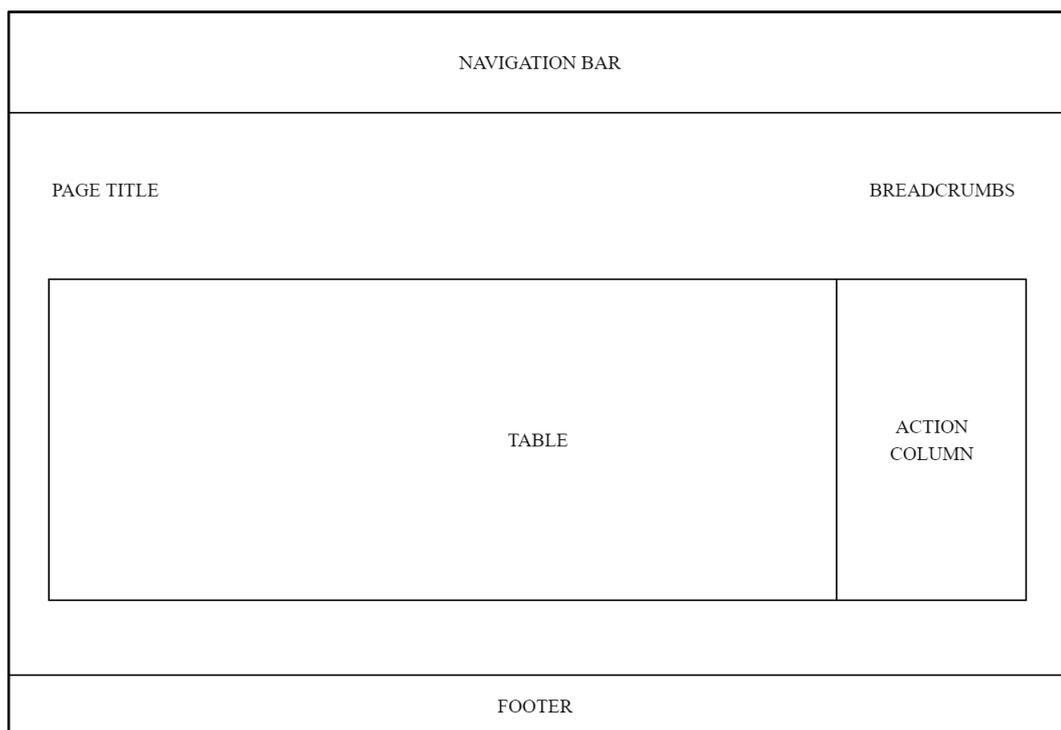
Pelaksanaan kerja magang berupa pengembangan modul *Report Rekonsiliasi Virtual Account* dan *Report History* pada web *Reconcile Admin* akan dijelaskan pada bagian ini.

### 3.3.1 Proses Pelaksanaan

*Reconcile Admin* adalah *project* yang dibangun sebagai aplikasi berbasis web dengan menggunakan *framework Laravel* yang memakai bahasa pemrograman *PHP*. *Database* yang digunakan dalam *project* adalah *MySQL*. *Supervisor* memberikan tugas untuk mengembangkan modul *Report Rekonsiliasi Virtual Account* dan *Report History*. Modul-modul tersebut akan digunakan oleh *user* untuk melakukan kegiatan rekonsiliasi laporan keuangan di web *Reconcile Admin*. *User* adalah semua orang yang ada di divisi *Finance & Acc*.

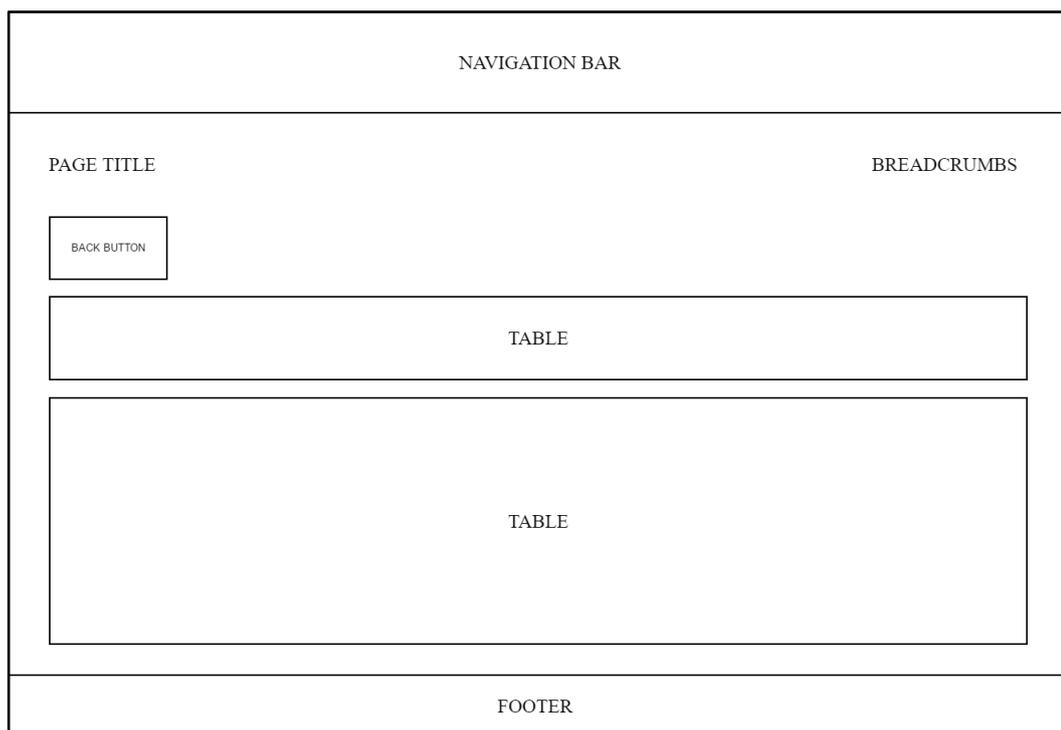
#### 1) Rancangan antarmuka

Rancangan antarmuka dibuat untuk memberikan gambaran hasil akhir dari *user interface* modul yang dikembangkan. Gambar 3.1 menunjukkan rancangan antarmuka halaman *Report Rekonsiliasi Virtual Account* dan *Report History*. Pada bagian *header* terdapat *navigation bar* untuk mengakses fitur lainnya di web *Reconcile Admin*. Judul dari halaman dan *breadcrumbs* terdapat di bawah *header*. Di baris ketiga, terdapat tabel untuk menampilkan data yang berkaitan dengan halaman. Pada tabel tersebut, ada satu kolom *action* yang berisi *button* untuk berinteraksi dengan data pada tabel dan pada bawah halaman terdapat *footer*.



Gambar 3.1 Rancangan Antarmuka Halaman Report Rekonsiliasi Virtual Account dan Report History

Gambar 3.2 menunjukkan rancangan antarmuka halaman Report Rekonsiliasi Virtual Account Detail. Pada paling atas halaman terdapat *navigation bar*, judul halaman dan *breadcrumbs* dibawah navigation bar, serta footer di bawah halaman. Halaman Report Rekonsiliasi Virtual Account Detail memiliki 2 tabel. Tabel pertama berisikan *summary* dari isi tabel kedua, sedangkan tabel kedua berisi data detail dari halaman Report Rekonsiliasi Virtual Account atau halaman Report History. Tombol kembali juga disediakan untuk kembali ke halaman sebelumnya.

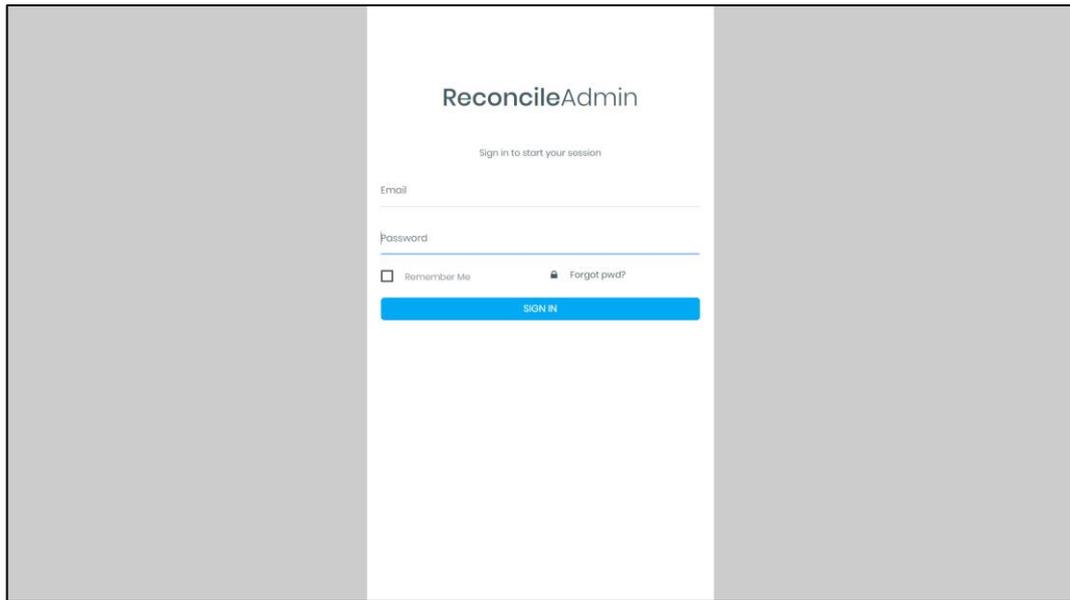


Gambar 3.2 Rancangan Antarmuka Halaman Report Rekonsiliasi Virtual Account

### Detail

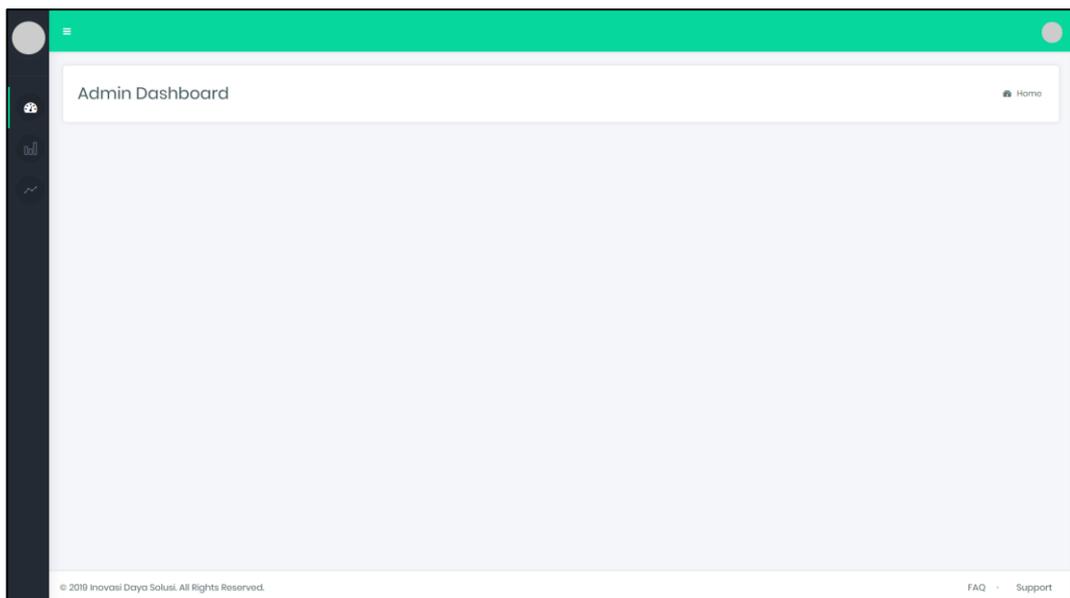
#### 2) *User interface*

*User interface* dibuat dengan menggunakan *framework Bootstrap* agar tampilan dari modul yang dibuat menjadi *responsive*. Saat *user* membuka *Reconcile Admin* pertama kali, akan muncul tampilan halaman Login. *User* harus mengisi *email* dan *password* dengan format yang benar dan tidak boleh mengosongkan salah satu atau kedua *field*. Apabila *email* atau *password* yang diisikan oleh *user* salah, maka halaman Login akan muncul *error message*. Tampilan halaman Login dapat dilihat pada Gambar 3.3.



Gambar 3.3 Halaman Login

Jika *user* sudah berhasil login, maka akan muncul halaman Beranda web *Reconcile Admin* seperti pada Gambar 3.4. Pada bagian kiri terdapat *navigation bar* yang dapat dipakai oleh user untuk memakai fitur yang ada di aplikasi dengan mudah. Halaman Report Rekonsiliasi Virtual Account dan halaman Report History dapat dituju dengan menggunakan *navigation bar*.

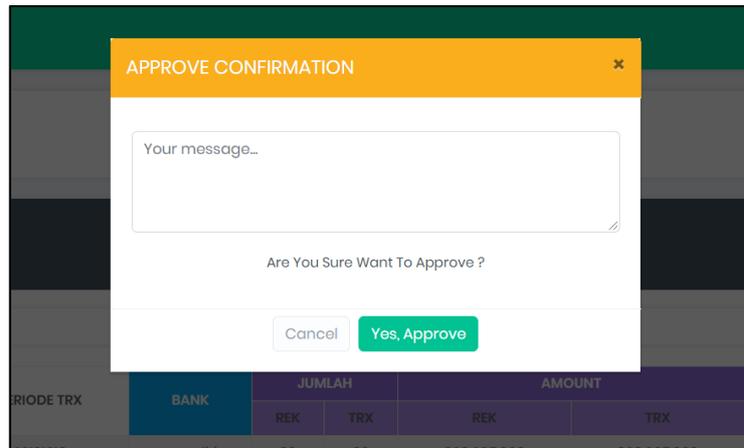


Gambar 3.4 Beranda

Tampilan halaman Report Rekonsiliasi Virtual Account dapat dilihat pada Gambar 3.5. Halaman ini berisi data total transaksi dalam satu periode. Pada halaman ini *user* hanya dapat melihat data yang termasuk *privilege user*, yaitu data yang belum di *approve* dan memerlukan *approve* dari *user* dengan *role* dan *level* yang sesuai dengan *user* yang sedang *login*. *User* dapat melihat data yang sudah *balance* maupun *unbalance* dan melakukan *approval* terhadap data tersebut dengan memilih *button* Approve yang kemudian akan menampilkan *modal Approve* seperti pada Gambar 3.6. Data yang dibidang *balance* adalah data yang memiliki jumlah yang sama antara transaksi yang tercatat (kolom TRX) dan jumlah *transfer* ke bank yang dilakukan (kolom REK). Terdapat juga *button* Detail untuk menuju halaman Report Rekonsiliasi Virtual Account Detail untuk melihat seluruh transaksi pada satu periode tertentu. Tampilan Report Rekonsiliasi Virtual Account Detail dapat dilihat pada Gambar 3.7.

PERIODE SETTLEMENT	PERIODE TRX	BANK	JUMLAH		AMOUNT		STATUS	ACTION	
			REK	TRX	REK	TRX		Detail	Approve
20191021	20191018	mandiri	36	36	308,065,968	308,065,968	BALANCE	Detail	Approve
20191021	20191020	mandiri	2	2	26,944,000	26,944,000	BALANCE	Detail	Approve
20191022	20191020	mandiri	1	1	12,481,000	12,481,000	BALANCE	Detail	Approve
20191022	20191021	mandiri	67	67	575,415,710	575,415,710	BALANCE	Detail	Approve
20191023	20191021	mandiri	1	1	12,591,000	12,591,000	BALANCE	Detail	Approve
20191023	20191022	mandiri	30	30	208,492,824	208,492,824	BALANCE	Detail	Approve

Gambar 3.5 Report Rekonsiliasi Virtual Account



Gambar 3.6 Modal Approve

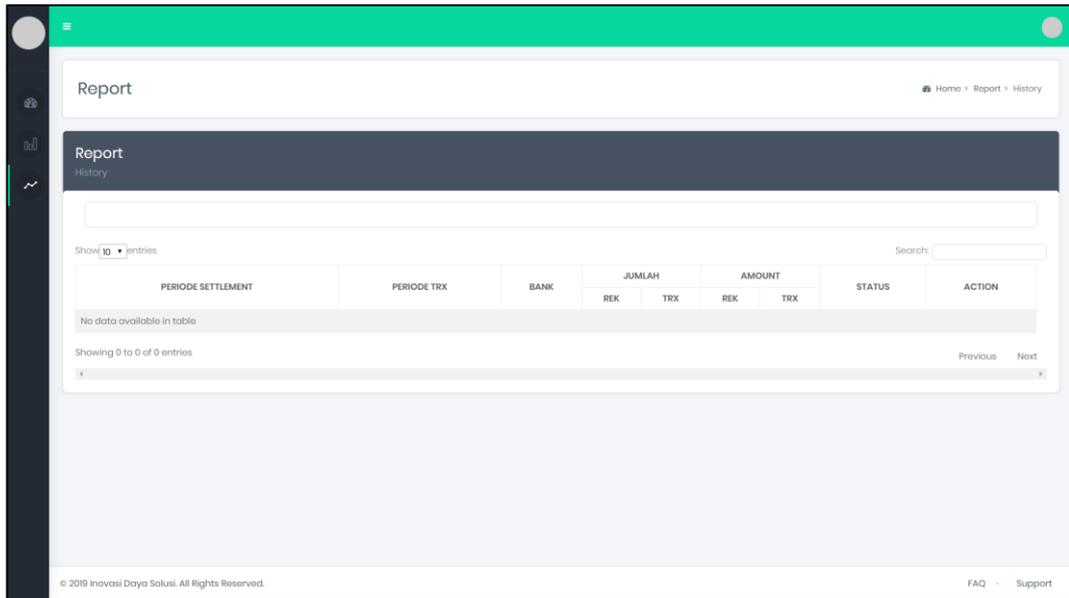
PRODUCT NAME	JUMLAH TRX	TOTAL REKON AMOUNT	TOTAL TRANSACTION AMOUNT
MAF	35	303,082,788	303,082,788
SMPVA	1	4,983,200	4,983,200

PRODUCT NAME	PERIODE SETTLEMENT	PERIODE TRX	NO VA	AMOUNT		TRXDATE	STATUS BILLER
				REK	TRX		
MAF	20191021	20191018	14198200101900734	228,000	228,000	2019-10-18 09:21:18	SUCCESS
MAF	20191021	20191018	141982001791900412	802,000	802,000	2019-10-18 17:08:01	SUCCESS
MAF	20191021	20191018	141982002371902377	606,000	606,000	2019-10-18 12:18:05	SUCCESS
MAF	20191021	20191018	141982003221900309	2,720,000	2,720,000	2019-10-18 17:34:18	SUCCESS
MAF	20191021	20191018	141982005071901137	2,050,000	2,050,000	2019-10-18 08:05:49	SUCCESS
MAF	20191021	20191018	141982006581900569	8,245,000	8,245,000	2019-10-18 17:25:51	SUCCESS
MAF	20191021	20191018	141982006581900812	15,141,000	15,141,000	2019-10-18 12:27:33	SUCCESS

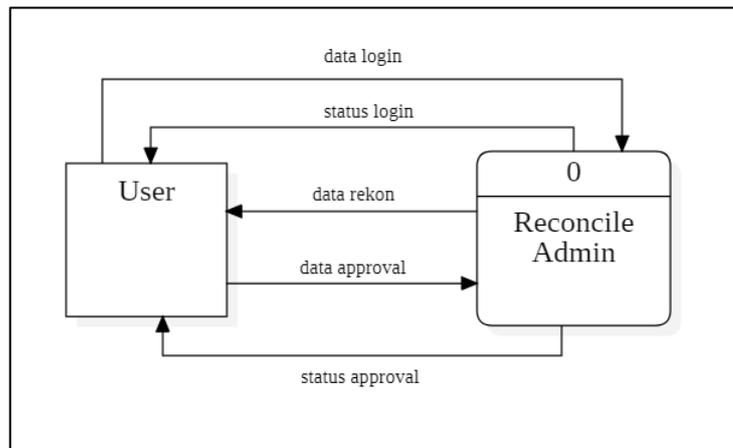
Gambar 3.7 Report Rekonsiliasi Virtual Account Detail

Tampilan halaman Report History dapat dilihat pada Gambar 3.8. Halaman ini berisikan data yang sudah di *approve* oleh user *Reconcile Admin*. Pada halaman ini terdapat juga button Detail sama seperti dengan di halaman Report Rekonsiliasi Virtual Account.



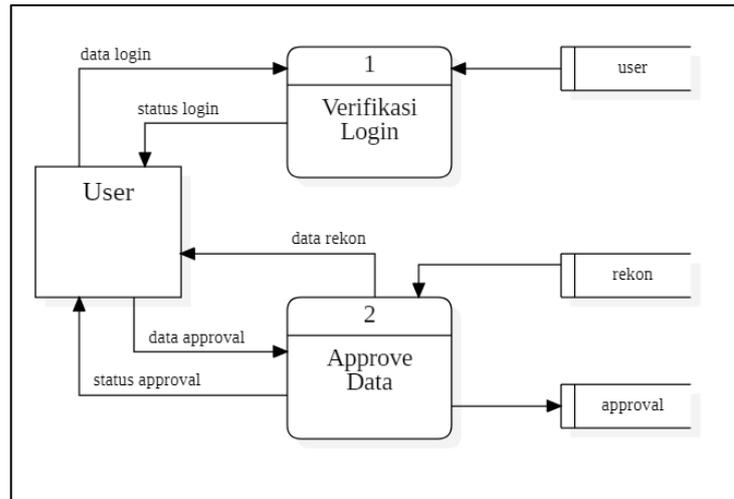
Gambar 3.8 Report History

### 3) Data Flow Diagram



Gambar 3.9 Diagram Konteks Reconcile Admin

Gambar 3.9 menunjukkan diagram konteks dari *Reconcile Admin*. Terdapat satu entitas saja yang berhubungan dengan aplikasi web, yaitu *user*. *User* mengirim data ke web *Reconcile Admin* berupa data *login* dan data *approval*. Lalu, data yang dikirimkan dari aplikasi web berupa status *login*, data rekonsiliasi, dan status *approval*.



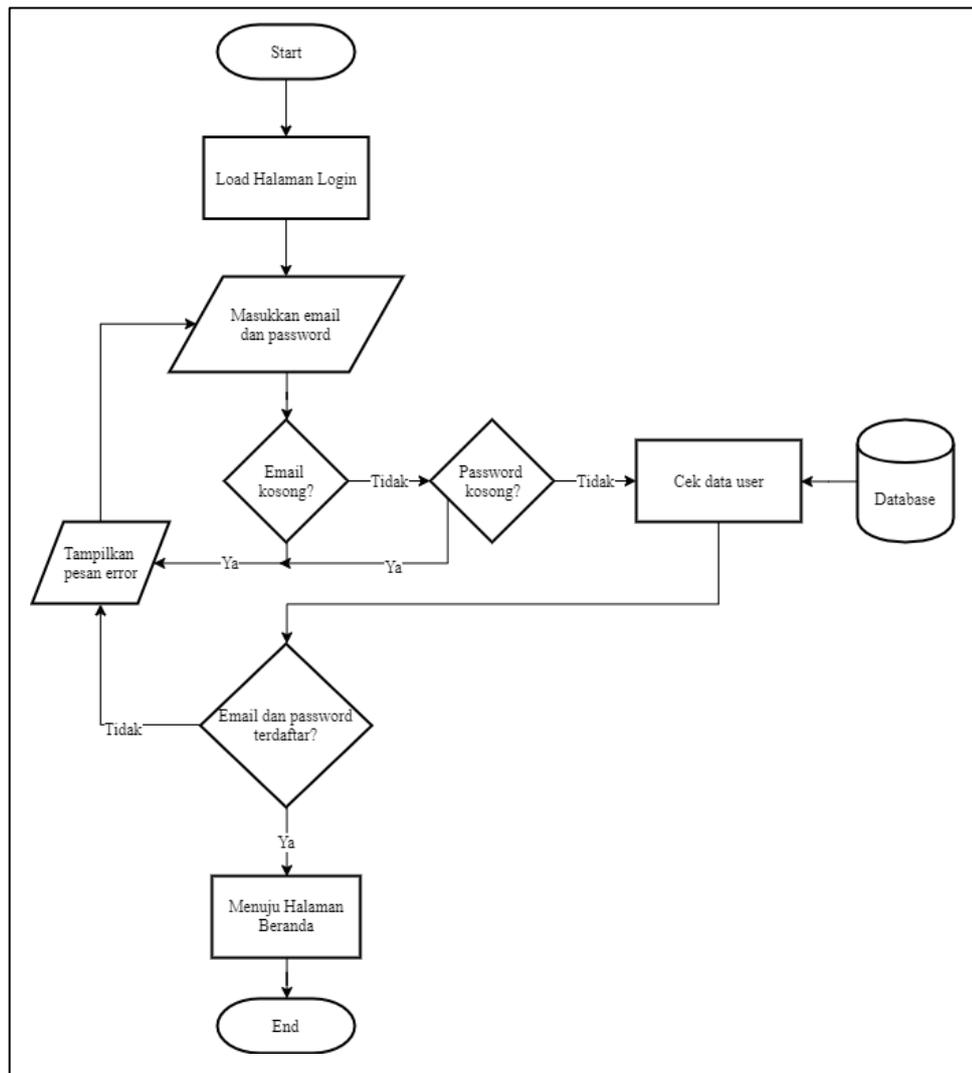
Gambar 3.10 Diagram Level 1 Reconcile Admin

Pada Gambar 3.10 terdapat Diagram Level 1 yang memiliki 2 proses utama. Proses utama yang pertama adalah Verifikasi Login yang memproses apakah *user* sudah terdaftar di *database*. Proses Verifikasi Login menerima data *login* dari *user* kemudian akan melakukan pengecekan ke *database* apakah data *login* sudah terdaftar. Apabila pengecekan sudah selesai, proses akan mengirim *response* berupa status *login* ke entitas *user*.

Proses kedua adalah *Approve Data*, yang akan memproses *approval* dari *user*. Proses akan mengirim data rekonsiliasi yang didapat dari *database* ke *user*, kemudian *user* dapat melakukan *approval* dengan mengirimkan data *approval*. Proses akan melakukan pengecekan apakah data yang ingin di-*approve* sudah *balance* ke *database*. Setelah selesai proses akan mengirimkan *response* berupa status *approval*. Jika data yang di-*approve* sudah *balance* dan berhasil di-*approve* maka proses akan menyimpan data *approval* ke *database*.

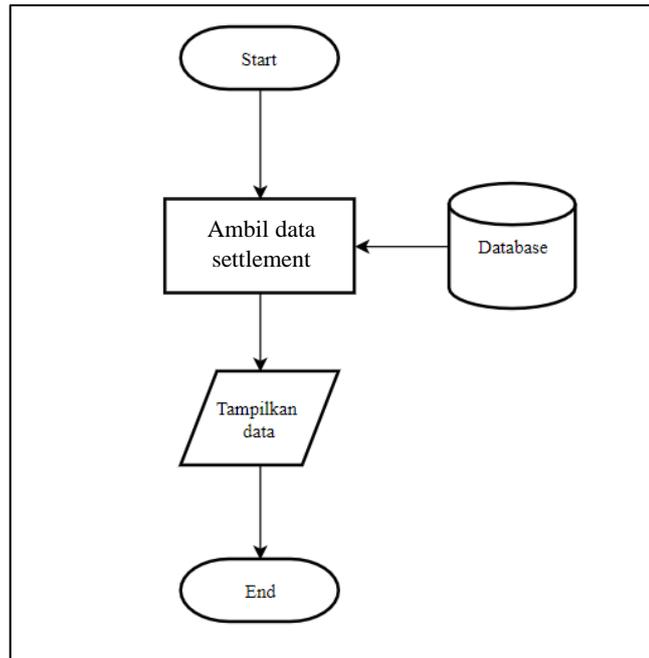
#### 4) Flowchart

Sebelum mengembangkan modul, dilakukan perancangan menggunakan *flowchart*. *Flowchart* yang dibuat berupa perancangan proses *login*, *approval*, dan mengambil data yang akan ditampilkan di halaman web.



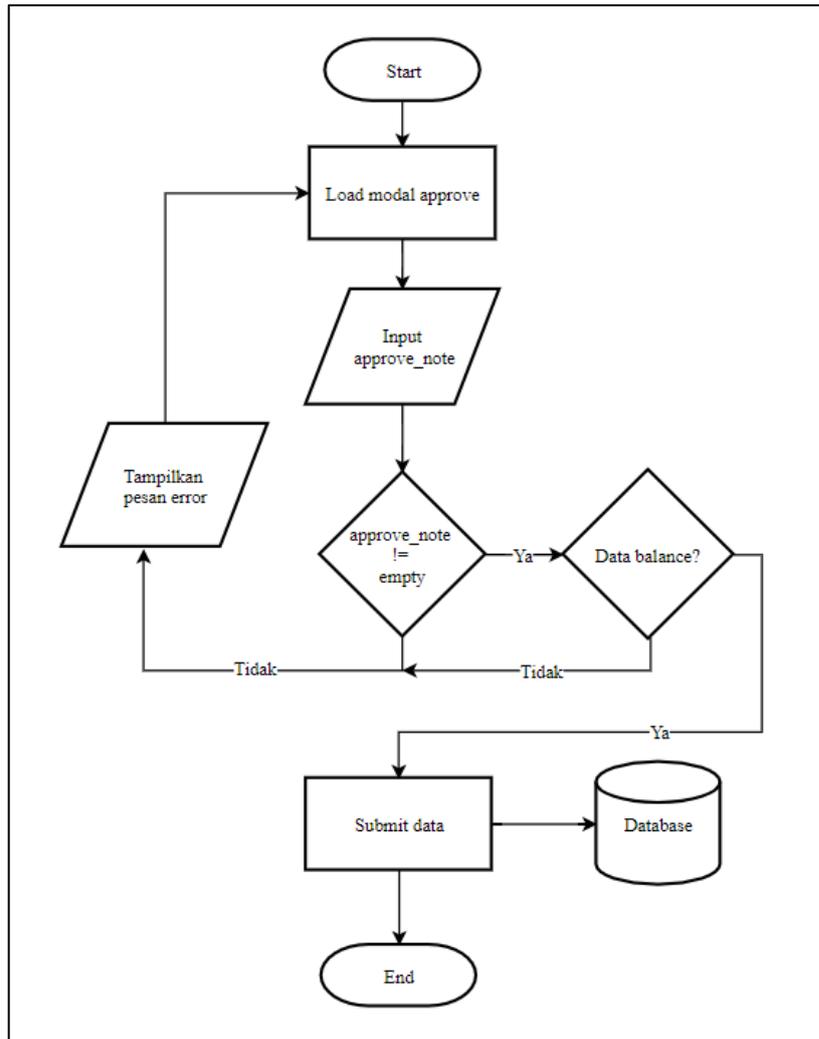
Gambar 3.11 *Flowchart* Proses *Login*

Gambar 3.11 menunjukkan proses *login* dari web *Reconcile Admin*. User akan memasukan *email* dan *password*, setelah data di-*submit* akan diperiksa apakah *email* dan *password* terdaftar di *database* atau tidak, jika memang terdaftar maka proses *login* selesai.



Gambar 3.12 *Flowchart* menampilkan data *settlement*

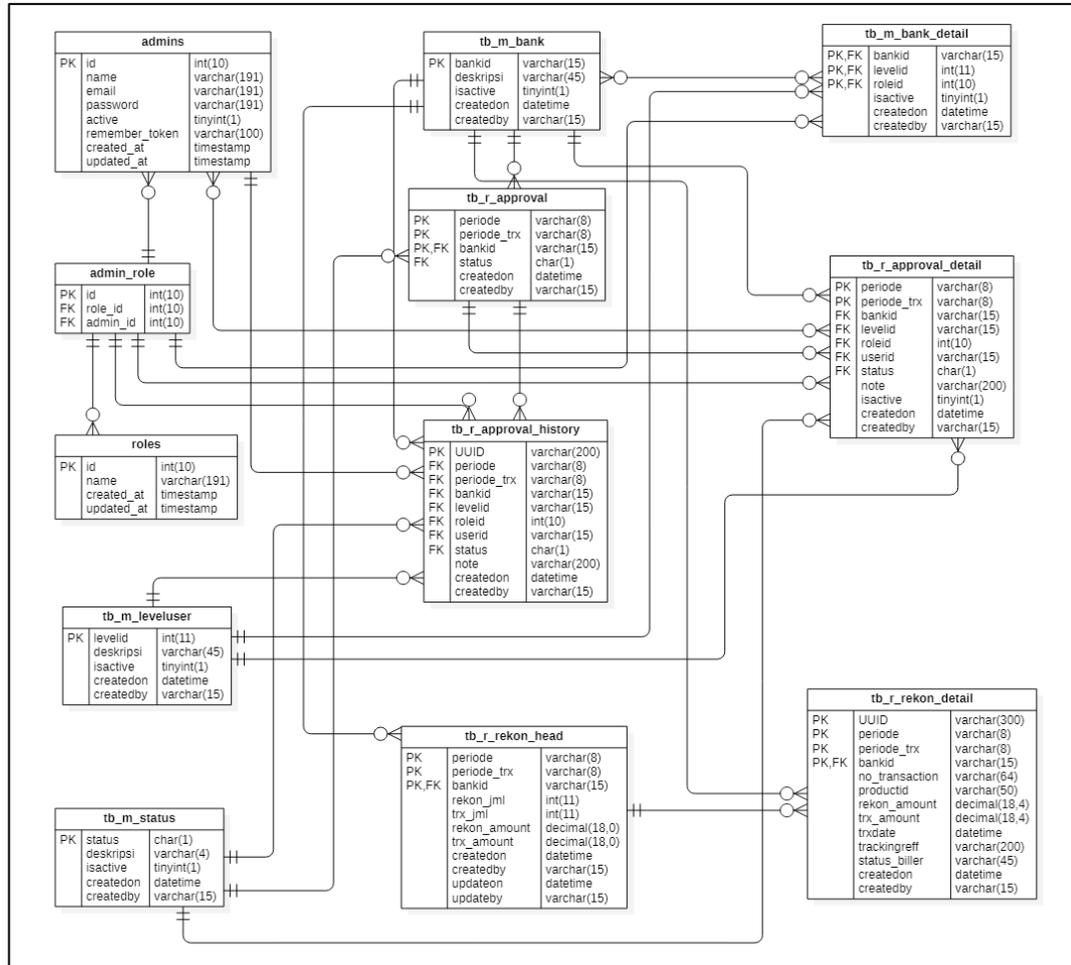
Pada Gambar 3.12 menunjukkan proses untuk menampilkan data dari database. Pada modul Report Rekonsiliasi Virtual Account data yang diambil dari database sesuai dengan *role* dan *level user*, sehingga hanya *user* yang memiliki *privilege* di data tersebut yang dapat melihat data dan dapat melakukan *approval*. Namun, pada modul Report History data yang diambil dari *database* adalah data yang sudah di-*approve*.



Gambar 3.13 *Flowchart data approval*

Gambar 3.13 menunjukkan proses *approval* sebuah data. Pertama, *user* akan memasukkan data *approve\_note*. Setelah data *approve\_note* di-submit aplikasi web akan memeriksa apakah data yang ingin di *approve* sudah *balance* atau belum. Jika data sudah *balance*, maka data di *database* akan di-*update* dan data *approve\_note* akan disimpan ke *database*.

## 5) Entity Relationship Diagram



Gambar 3.14 Entity Relationship Diagram

*Entity Relationship Diagram* pada web *Reconcile Admin* dapat dilihat pada Gambar 3.14. Terdapat 12 tabel yang digunakan dalam mengembangkan modul *Report Rekonsiliasi Virtual Account* dan *Report History*. Berikut adalah penjelasan singkat dari setiap tabel yang dipakai.

### a) *admins*

Tabel ini digunakan untuk menyimpan data dari *user* sehingga dapat melakukan verifikasi *login* dari data yang dimasukkan oleh *user*. Selain itu,

tabel ini juga digunakan untuk menentukan data *user* yang akan dimasukkan ke *database* setelah *approval*.

b) *admin\_role*

Tabel ini digunakan untuk menyimpan *role* dari *user* sehingga *privilege* dari *user* dapat ditentukan.

c) *roles*

Table ini digunakan untuk menyimpan deskripsi dari suatu *role*.

d) *tb\_m\_leveluser*

Tabel ini digunakan untuk menyimpan *level* dari *user* sehingga *privilege* dari *user* dapat ditentukan.

e) *tb\_m\_status*

Tabel ini digunakan untuk menyimpan deskripsi dari tindakan *approval* oleh *user*.

f) *tb\_m\_bank*

Tabel ini digunakan untuk menyimpan informasi bank.

g) *tb\_m\_bank\_detail*

Tabel ini digunakan untuk menyimpan informasi detail mengenai *privilege user* dengan data yang terkait bank tertentu.

h) *tb\_r\_approval*

Tabel ini digunakan untuk menyimpan informasi mengenai *approval* dari data transaksi.

i) *tb\_r\_approval\_detail*

Tabel ini digunakan untuk menyimpan informasi detail mengenai *approval* dari data transaksi.

j) *tb\_r\_approval\_history*

Tabel ini digunakan untuk menyimpan tindakan *approve* yang dilakukan oleh *user*.

k) *tb\_r\_rekon\_head*

Tabel ini digunakan untuk menyimpan data total transaksi yang terjadi pada satu periode tertentu. Data dari tabel ini akan ditampilkan di halaman Report Rekonsiliasi Virtual Account dan halaman Report History. Data dari tabel ini yang akan di *approve* oleh *user* yang memiliki *privilege* untuk melakukannya.

l) *tb\_r\_rekon\_detail*

Tabel ini digunakan untuk menyimpan informasi mengenai seluruh transaksi pada satu periode tertentu.

### **3.3.2 Kendala yang Ditemukan**

Kendala-kendala yang ditemukan dalam pelaksanaan kerja magang di PT Inovasi Daya Solusi adalah sebagai berikut.

- 1) Mempelajari bahasa pemrograman dan *framework* yang dipakai dalam pengembangan *project*.
- 2) Memahami kegunaan kolom yang ada pada table di *database* yang sudah dibuat.
- 3) Memahami hasil akhir tugas yang diinginkan oleh supervisi.

### 3.3.3 Solusi atas Kendala yang Ditemukan

Solusi atas kendala-kendala yang ditemukan dalam pelaksanaan kerja magang di PT Inovasi Daya Solusi adalah sebagai berikut.

- 1) Belajar dari *tutorial* mengenai bahasa pemrograman dan *framework* yang dipakai dalam pengembangan *project*.
- 2) Mencari solusi di internet saat menemukan *error* saat melakukan pengembangan *project*.
- 3) Bertanya kepada supervisi mengenai tujuan dari *project* yang dikembangkan dan detail-detail seputar *project*.