

LAPORAN TUGAS BESAR SISTEM BASIS DATA

Tema: Website Linkaja

Dosen Pengampu: Sarah Purnamawati ST., MSc.



DISUSUN OLEH:

KELOMPOK 4

Ivan Tandella (NIM 201402022)

Jefry (NIM 201402049)

Tsabitah Muflihza (NIM 201402070)

Bagas Ramadhan Siregar (NIM 201402100)

Vanissy Arbashika Putri (NIM 201402103)

UNIVERSITAS SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
TEKNOLOGI INFORMASI

2021

KATA PENGANTAR

Puji syukur kami panjatkan kehadiran Tuhan Yang Maha Esa Allah SWT karena dengan rahmat-Nya kami dapat menyelesaikan laporan tugas besar Sistem Basis Data yang bertema Website Linkaja ini dengan baik. Tidak lupa kami ucapkan terima kasih kepada yang terhormat Ibu Sarah Purnamawati ST., MSc. dan Abangda Arsyah Fikri selaku Asisten Lab kami dan juga seluruh pihak yang membantu kami dalam pengerjaan tugas besar ini.

Tujuan kami menulis laporan ini adalah untuk memenuhi tugas pada mata kuliah Sistem Basis Data, laporan ini kami susun dengan sepuh hati berdasarkan informasi yang kami peroleh dari beberapa sumber pada media elektronik, dan kami berharap laporan yang kami susun ini dapat menambah wawasan para pembaca mengenai tugas besar yang kami kerjakan.

Kami sebagai penulis menyadari bahwa terdapat banyak kekurangan pada laporan yang kami buat ini. Oleh karena itu, kami sangat mengharapkan kritik dan saran dari para pembaca demi kesempurnaan laporan ini, semoga laporan yang kami buat ini dapat berguna bagi para pembaca laporan kami.

Medan, November 2021

DAFTAR ISI

KATA PENGANTAR	1
DAFTAR ISI.....	2
BAB I.....	3
PENDAHULUAN	3
1.1 Latar Belakang	3
1.2 Definisi Linkaja.....	3
1.3 Tujuan.....	4
1.4 Manfaat.....	4
BAB II.....	5
PEMBAHASAN	5
2.1 Proses Bisnis	5
2.2 Analisis Entitas, Atribut, Dan Relationship (ERD).....	6
2.3 Relational Database Schema	6
2.4 Tabel Database	7
2.5 Tampilan Website	11
2.6 Query	24
BAB III	39
PENUTUP.....	39
3.1 Kesimpulan.....	39

BAB I

PENDAHULUAN

1.1 Latar Belakang

Basis Data/Database merupakan sebuah koleksi dari data yang saling berelasi, Data perlu disimpan dalam suatu database untuk keperluan penyediaan informasi lebih lanjut. Data dalam database perlu diorganisasikan sedemikian rupa sehingga informasi yang terkandung didalamnya mudah diakses. Database merupakan komponen utama sistem informasi karena semua informasi untuk pengambilan keputusan berasal dari data di database. Pengelolaan database yang buruk dapat mengakibatkan ketidaktersediaan data penting yang digunakan untuk menghasilkan informasi yang diperlukan.

Basis data sangat menentukan kelancaran jalannya sebuah aplikasi web serta perkembangannya, oleh karena itu basis data harus diatur sedemikian rupa agar sebuah aplikasi dapat berjalan dengan baik. Dengan memanfaatkan basis data sebagai kekuatan dan peluang untuk kelancaran aplikasi web serta perkembangannya, maka manajemen basis data diperlukan untuk menjalankan fungsi dan tugas dari unit-unit aplikasi, agar informasi yang dibutuhkan dalam pengambilan keputusan untuk menunjang keberhasilan tujuan dan sasaran aplikasi dapat ditemukan dengan cepat dan akurat.

Berdasarkan latar belakang tersebut kami membuat sistem aplikasi LinkAja. Aplikasi ini merupakan duplikasi dari aplikasi Linkaja. Aplikasi ini dibangun dengan menggunakan framework Laravel versi 8 dan database MySQL. Aplikasi ini diharapkan dapat mempermudah admin dalam mengelola data top up dan sejenisnya serta mempermudah pengguna dalam memakai aplikasi uang digital. Adapun latar belakang lain dari pembuatan aplikasi Linkaja ini yaitu untuk memenuhi tugas besar pada mata kuliah Sistem Basis Data.

1.2 Definisi Linkaja

LinkAja adalah sebuah layanan uang elektronik yang berbasis aplikasi untuk melakukan berbagai transaksi nontunai dengan mudah dan praktis. LinkAja diresmikan pada tanggal 30 Juni 2019 di Jakarta. LinkAja dulunya bernama T-Cash. Layanan yang satu ini dapat digunakan sebagaimana layaknya layanan keuangan berbasis digital lainnya, yang membuat berbagai transaksi keuangan bisa dilakukan dengan mudah dan cepat. Kamu bisa aktifkan dan

menikmati beragam kemudahan dalam melakukan transaksi nontunai melalui aplikasi ini. Nikmati kemudahan dan kenyamanan menggunakan LinkAja, transaksi semudah menggunakan smartphone di genggaman.

1.3 Tujuan

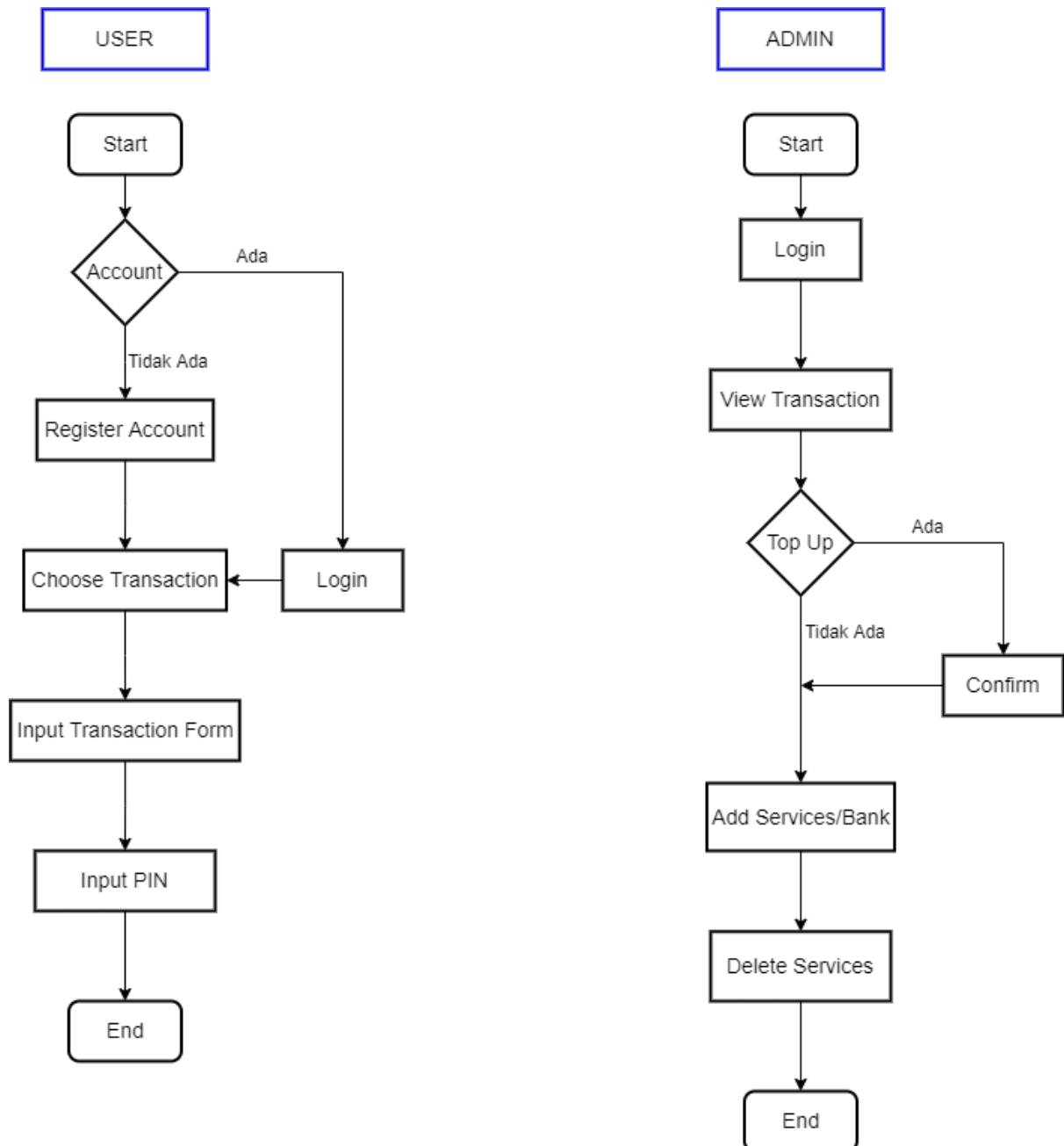
1. Untuk mempermudah layanan uang digital dan pengelolaan data pada aplikasi Linkaja.
2. Mendapatkan hasil data yang lebih efisien dan akurat dalam pengelolaan data.

1.4 Manfaat

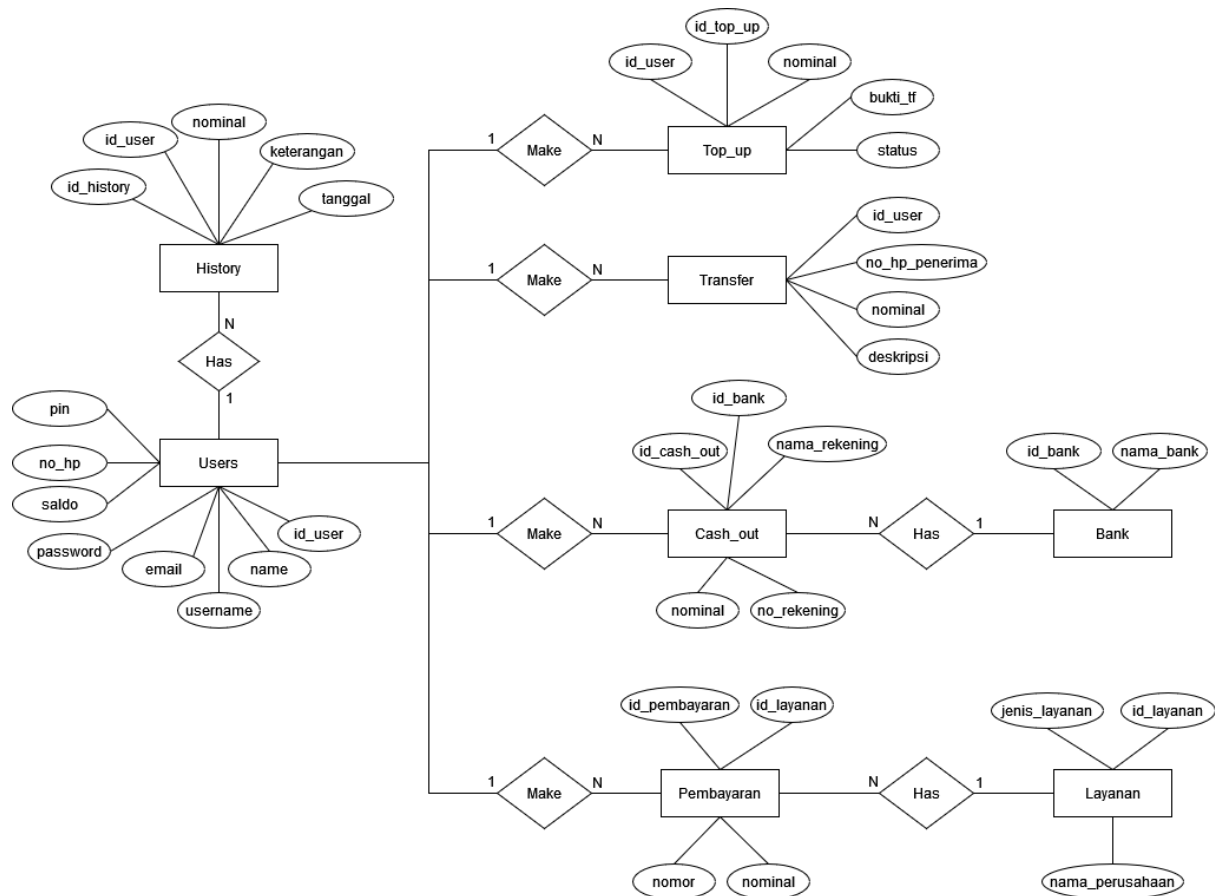
1. Dapat memahami seperti apa manajemen database pada aplikasi LinkAja.
2. Dapat memahami pengaplikasian pembuatan database dengan MySQL dalam pembuatan website aplikasi LinkAja.

BAB II PEMBAHASAN

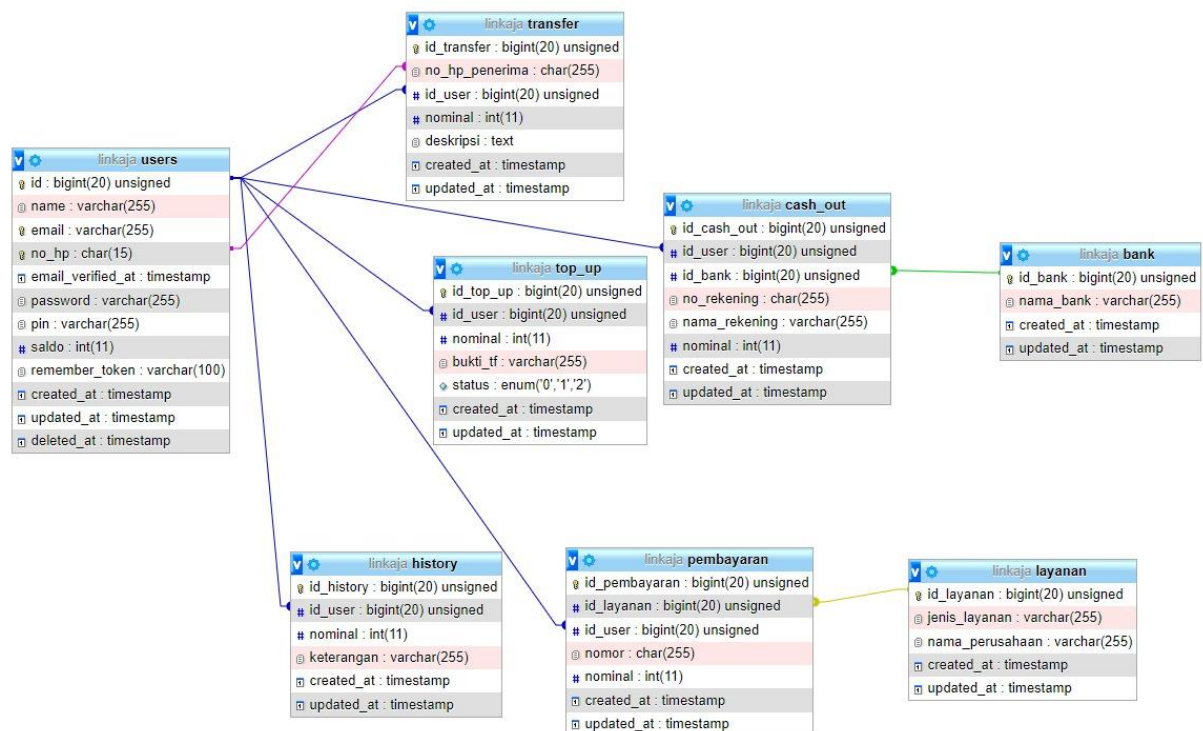
2.1 Proses Bisnis



2.2 Analisis Entitas, Atribut, Dan Relationship (ERD)



2.3 Relational Database Schema



2.4 Tabel Database

❖ Tabel Users

Server: 127.0.0.1 » Database: linkaja » Table: users

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 name	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	3 email	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	4 no_hp	char(15)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	5 email_verified_at	timestamp			Yes	NULL			Change Drop More
<input type="checkbox"/>	6 password	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	7 pin	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	8 saldo	int(11)			No	0			Change Drop More
<input type="checkbox"/>	9 remember_token	varchar(100)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	10 created_at	timestamp			Yes	NULL			Change Drop More
<input type="checkbox"/>	11 updated_at	timestamp			Yes	NULL			Change Drop More
<input type="checkbox"/>	12 deleted_at	timestamp			Yes	NULL			Change Drop More

☐ Check all With selected: Browse Change Drop Primary Unique Index Fulltext Add to central columns

❖ Tabel History

Server: 127.0.0.1 » Database: linkaja » Table: history

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id_history	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 id_user	bigint(20)		UNSIGNED	No	None			Change Drop More
<input type="checkbox"/>	3 nominal	int(11)			No	None			Change Drop More
<input type="checkbox"/>	4 keterangan	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	5 created_at	timestamp			Yes	NULL			Change Drop More
<input type="checkbox"/>	6 updated_at	timestamp			Yes	NULL			Change Drop More

☐ Check all With selected: Browse Change Drop Primary Unique Index Fulltext Add to central columns

Remove from central columns

Print Propose table structure Track table Move columns Normalize

Add column(s)

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id_history	10	A	No	
Edit Drop	history_id_user_foreign	BTREE	No	No	id_user	5	A	No	

Create an index on columns

❖ Tabel Top_Up

Server: 127.0.0.1 » Database: linkaja » Table: top_up

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_top_up	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	id_user	bigint(20)		UNSIGNED	No	None			Change Drop More
3	nominal	int(11)			No	None			Change Drop More
4	bukti_tf	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
5	status	enum('0', '1', '2')	utf8mb4_unicode_ci		No	0			Change Drop More
6	created_at	timestamp			Yes	NULL			Change Drop More
7	updated_at	timestamp			Yes	NULL			Change Drop More

☐ Check all
 With selected: Browse Change Drop Primary Unique Index Fulltext Add to central columns

Remove from central columns

Print Propose table structure Track table Move columns Normalize

Add 1 column(s) after updated_at Go

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id_top_up	3	A	No	
Edit Drop	top_up_id_user_foreign	BTREE	No	No	id_user	3	A	No	

Create an index on 1 columns Go

Console

❖ Tabel Transfer

Server: 127.0.0.1 » Database: linkaja » Table: transfer

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_transfer	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	no_hp_penerima	char(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	id_user	bigint(20)		UNSIGNED	No	None			Change Drop More
4	nominal	int(11)			No	None			Change Drop More
5	deskripsi	text	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
6	created_at	timestamp			Yes	NULL			Change Drop More
7	updated_at	timestamp			Yes	NULL			Change Drop More

☐ Check all
 With selected: Browse Change Drop Primary Unique Index Fulltext Add to central columns

Remove from central columns

Print Propose table structure Track table Move columns Normalize

Add 1 column(s) after updated_at Go

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id_transfer	2	A	No	
Edit Drop	transfer_id_user_foreign	BTREE	No	No	id_user	2	A	No	
Edit Drop	transfer_no_hp_penerima_foreign	BTREE	No	No	no_hp_penerima	2	A	No	

Create an index on 1 columns Go

Console

❖ Tabel Cash_Out

Server: 127.0.0.1 » Database: linkaja » Table: cash_out

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_cash_out	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	id_user	bigint(20)		UNSIGNED	No	None			Change Drop More
3	id_bank	bigint(20)		UNSIGNED	No	None			Change Drop More
4	no_rekening	char(255)	utf8mb4_unicode_ci		No	None			Change Drop More
5	nama_rekening	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
6	nominal	int(11)			No	None			Change Drop More
7	created_at	timestamp			Yes	NULL			Change Drop More
8	updated_at	timestamp			Yes	NULL			Change Drop More

☐ Check all With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#) [Fulltext](#) [Add to central columns](#)
[Remove from central columns](#)

[Print](#) [Propose table structure](#) [Track table](#) [Move columns](#) [Normalize](#)
[Add](#) 1 column(s) after updated_at [Go](#)

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id_cash_out	1	A	No	
Edit Drop	cash_out_id_user_foreign	BTREE	No	No	id_user	1	A	No	
Edit Drop	cash_out_id_bank_foreign	BTREE	No	No	id_bank	1	A	No	

❖ Tabel Bank

Server: 127.0.0.1 » Database: linkaja » Table: bank

[Table structure](#) [Relation view](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_bank	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	nama_bank	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	created_at	timestamp			Yes	NULL			Change Drop More
4	updated_at	timestamp			Yes	NULL			Change Drop More

☐ Check all With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#) [Fulltext](#) [Add to central columns](#)
[Remove from central columns](#)

[Print](#) [Propose table structure](#) [Track table](#) [Move columns](#) [Normalize](#)
[Add](#) 1 column(s) after updated_at [Go](#)

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id_bank	5	A	No	

Create an index on 1 columns [Go](#)

❖ Tabel Pembayaran

Server: 127.0.0.1 » Database: linkaja » Table: pembayaran

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id_pembayaran	bigint(20)	UNSIGNED	No	None		AUTO_INCREMENT	
<input type="checkbox"/>	2	id_layanan	bigint(20)	UNSIGNED	No	None			
<input type="checkbox"/>	3	id_user	bigint(20)	UNSIGNED	No	None			
<input type="checkbox"/>	4	nomor	char(255)	utf8mb4_unicode_ci	No	None			
<input type="checkbox"/>	5	nominal	int(11)		No	None			
<input type="checkbox"/>	6	created_at	timestamp		Yes	NULL			
<input type="checkbox"/>	7	updated_at	timestamp		Yes	NULL			

☐ Check all With selected: Browse Change Drop Primary Unique Index Fulltext Add to central columns
 Remove from central columns

Print Propose table structure Track table Move columns Normalize

Add 1 column(s) after updated_at Go

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
	PRIMARY	BTREE	Yes	No	id_pembayaran	4	A	No	
	pembayaran_id_user_foreign	BTREE	No	No	id_user	2	A	No	
	pembayaran_id_layanan_foreign	BTREE	No	No	id_layanan	4	A	No	

❖ Tabel Layanan

Server: 127.0.0.1 » Database: linkaja » Table: layanan

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id_layanan	bigint(20)	UNSIGNED	No	None		AUTO_INCREMENT	
<input type="checkbox"/>	2	jenis_layanan	varchar(255)	utf8mb4_unicode_ci	No	None			
<input type="checkbox"/>	3	nama_perusahaan	varchar(255)	utf8mb4_unicode_ci	No	None			
<input type="checkbox"/>	4	created_at	timestamp		Yes	NULL			
<input type="checkbox"/>	5	updated_at	timestamp		Yes	NULL			

☐ Check all With selected: Browse Change Drop Primary Unique Index Fulltext Add to central columns
 Remove from central columns

Print Propose table structure Track table Move columns Normalize

Add 1 column(s) after updated_at Go

Indexes

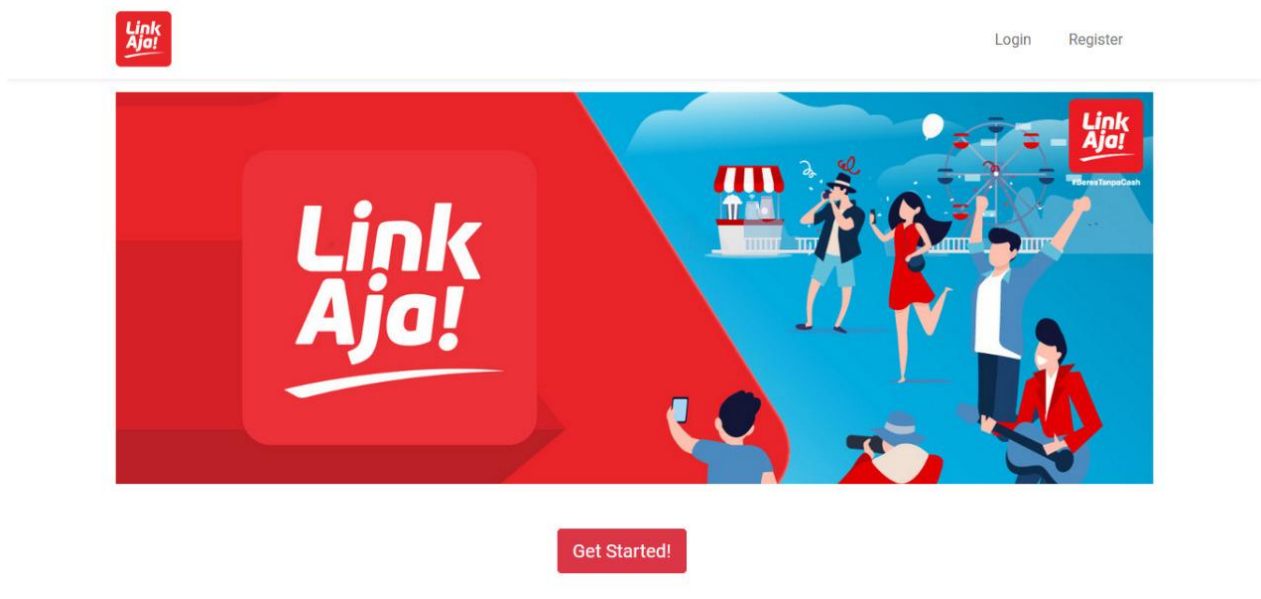
Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
	PRIMARY	BTREE	Yes	No	id_layanan	10	A	No	

Create an index on 1 columns Go

2.5 Tampilan Website

A. Tampilan User

Landing Page, merupakan halaman awal sebelum masuk ke menu Login atau Register.



Register, halaman untuk melakukan registrasi bagi pengguna yang belum mempunyai akun.

The screenshot shows the Link Aja! Register page. At the top left is the Link Aja! logo. At the top right are links for 'Login' and 'Register'. The main content is a registration form with a red header labeled 'Register'. The form contains the following fields: 'Name', 'E-Mail Address', 'Phone Number' (with a placeholder '08xxxxx'), 'Password', and 'Confirm Password'. Below the fields is a link that says 'Already have an account? [Login](#)'. At the bottom of the form is a red button labeled 'Register'.

Login, adalah halaman yang digunakan untuk masuk ke home website bagi pengguna yang sudah memiliki akun.

Link Aja!

Login Register

Login

Enter your phone number

Enter your password

Don't have an account? [Register here](#)

Login

Link Aja!

Follow us

f t i y in

LinkAja adalah layanan keuangan elektronik berbasis aplikasi yang akan

Home, merupakan halaman utama yang berisi menu-menu yang terdapat pada halaman user.

Link Aja!

Home History Profile Logout


Hi, Ivan Tandella Rp.485000

Top-up Send Cash Out More

Payment & Bills

PLN PDAM Pulsa Internet

History, merupakan halaman yang menampilkan riwayat transaksi apa saja yang pernah terjadi.



HomeHistoryProfileLogout

History


Top-Up Rejected
2021-12-08 02:43:14Rp.50000

Top-Up
2021-12-08 02:38:56Rp.100000

Pay Internet 123413412
2021-12-08 01:52:39Rp.200000

Buy Pulsa 08123456789
2021-12-08 01:52:06Rp.11000


Profile, halaman yang berisi detail profil user dan tombol untuk melakukan update pin.








HomeHistoryProfileLogout

Profile

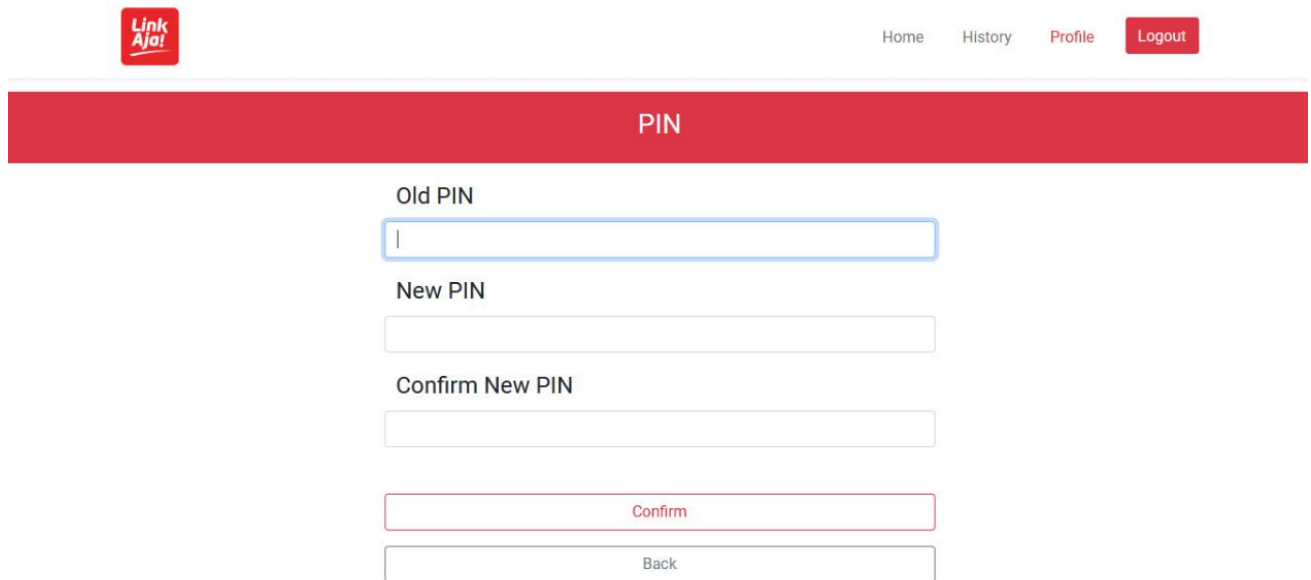
Nama : Ivan Tandella
Email : ivantandella@gmail.com
Phone Number : 081381229986
PIN : *****
BackUpdate PIN



Follow us


LinkAja adalah layanan keuangan elektronik berbasis aplikasi yang akan

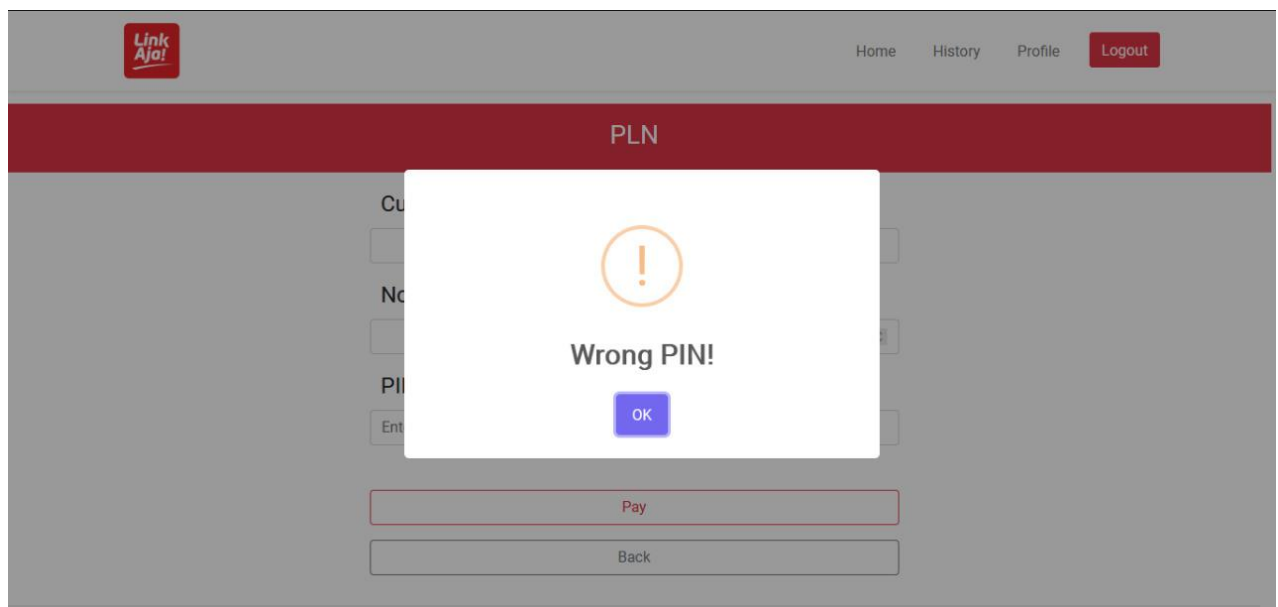
Update PIN, berguna bagi user yang ingin melakukan update atau memperbarui PIN pada akunnya



The screenshot shows the 'Link Aja!' logo in the top left corner. The top navigation bar includes links for 'Home', 'History', 'Profile', and a red 'Logout' button. Below this is a red header bar with the text 'PIN' in white. The main form area contains the following elements:

- 'Old PIN' label above a text input field.
- 'New PIN' label above a text input field.
- 'Confirm New PIN' label above a text input field.
- A red 'Confirm' button.
- A white 'Back' button.

PIN Validation, merupakan notifikasi validasi yang terjadi pada saat user salah memasukkan PIN ketika melakukan transaksi.

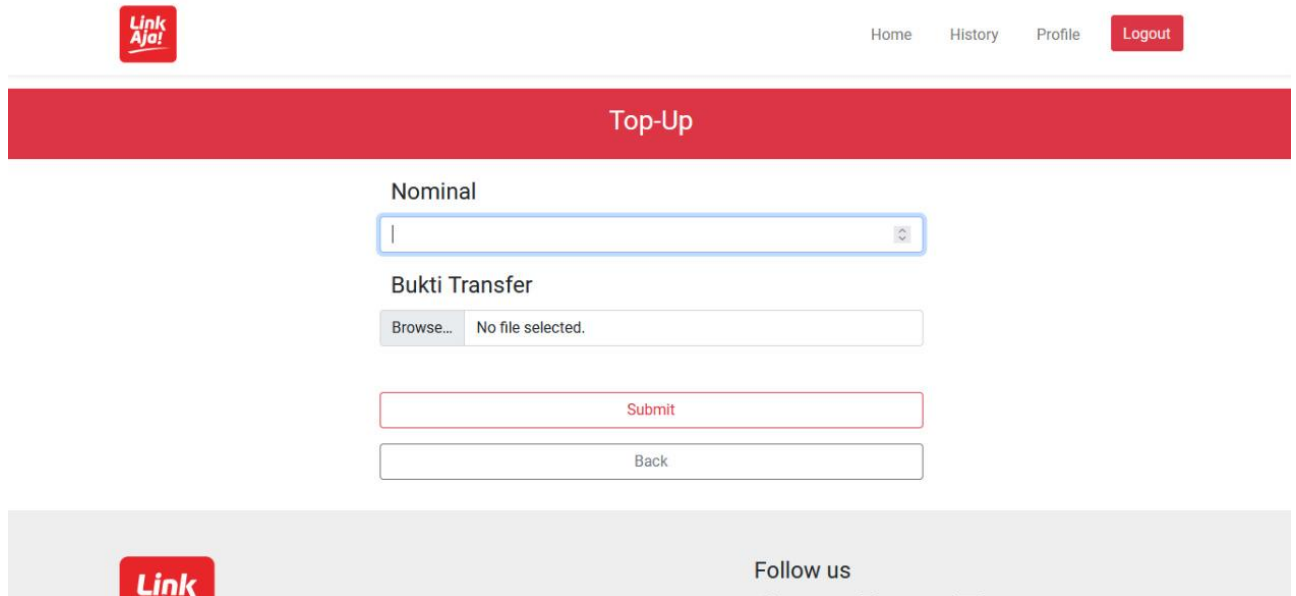


This screenshot shows the same 'Link Aja!' interface as the previous one, but with a modal dialog box displayed in the center. The dialog box has a white background and contains:

- An orange circular icon with a white exclamation mark.
- The text 'Wrong PIN!' in bold black font.
- A blue 'OK' button.

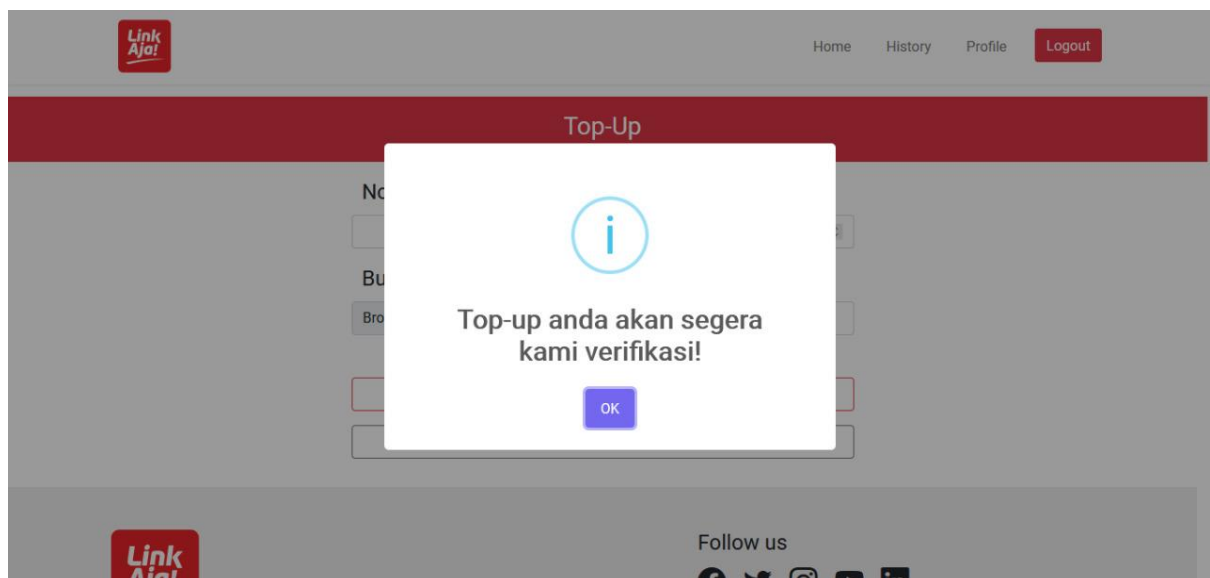
The background of the page is dimmed, showing the 'PIN' header and parts of the form fields.

Top Up, adalah halaman yang berfungsi bagi pengguna yang ingin melakukan pengisian saldo, halaman top up berisi sebuah form untuk memasukkan nominal top up dan bukti transfer.

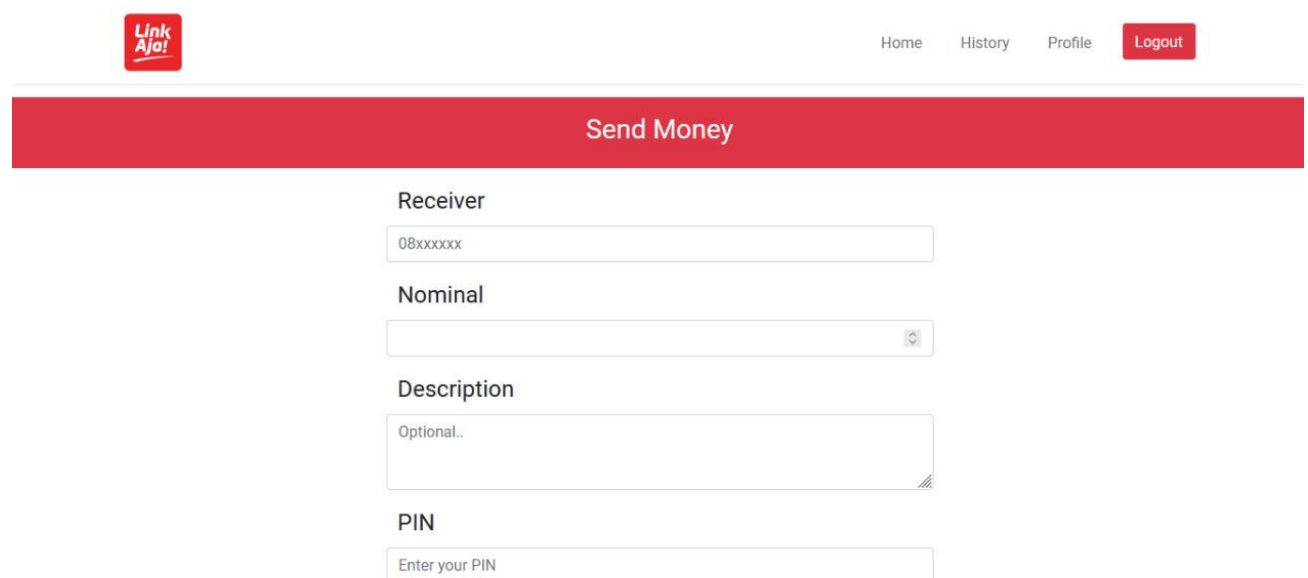


The screenshot shows the 'Top-Up' page of the Link Aja! application. At the top, there is a navigation bar with the Link Aja! logo on the left and links for 'Home', 'History', 'Profile', and a red 'Logout' button on the right. Below the navigation bar is a red header section with the text 'Top-Up' in white. The main content area contains a form with two sections: 'Nominal' and 'Bukti Transfer'. The 'Nominal' section has a text input field with a blue border and a small icon on the right. The 'Bukti Transfer' section has a 'Browse...' button and a text area that says 'No file selected.'. Below these sections are two buttons: a red 'Submit' button and a white 'Back' button. At the bottom of the page, there is a grey footer section with the Link Aja! logo on the left and the text 'Follow us' on the right, followed by social media icons.

Top Up Verification Request Notif, merupakan sebuah notifikasi yang menyampaikan pada user untuk menunggu admin menverifikasi top up yg dilakukan.

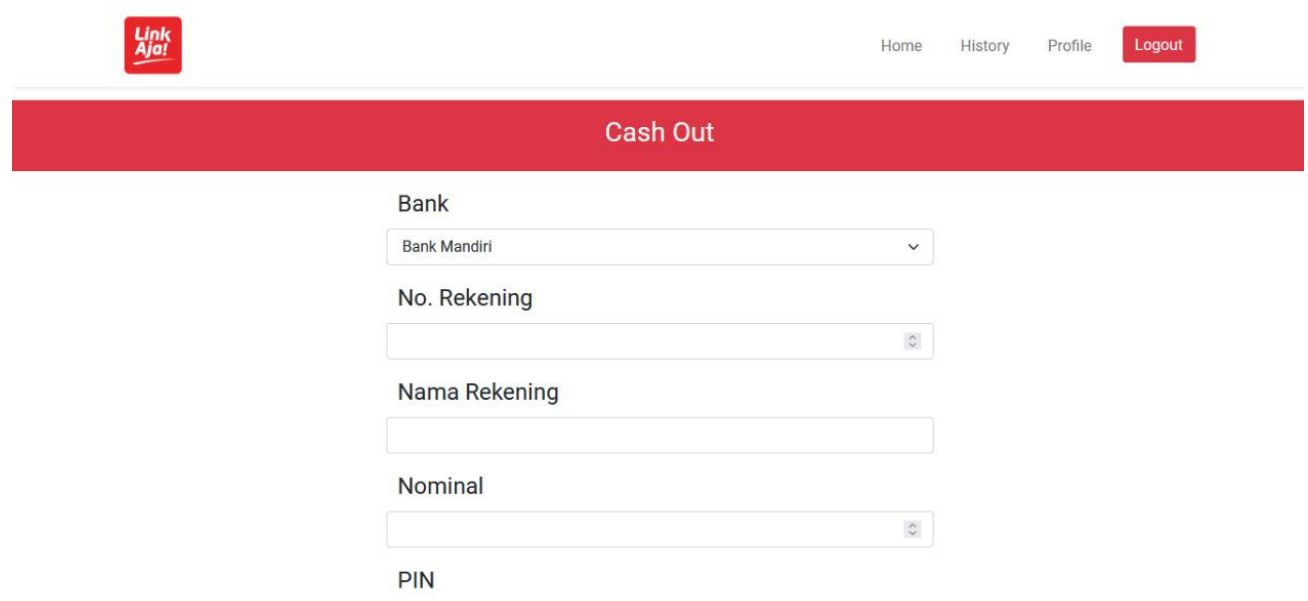


Send Money, merupakan sebuah fungsi yang memungkinkan pengguna untuk melakukan pengiriman uang dari satu akun ke akun yang lainnya, halaman send money berisi form untuk memasukkan nomor penerima, nominal, deskripsi, dan PIN.



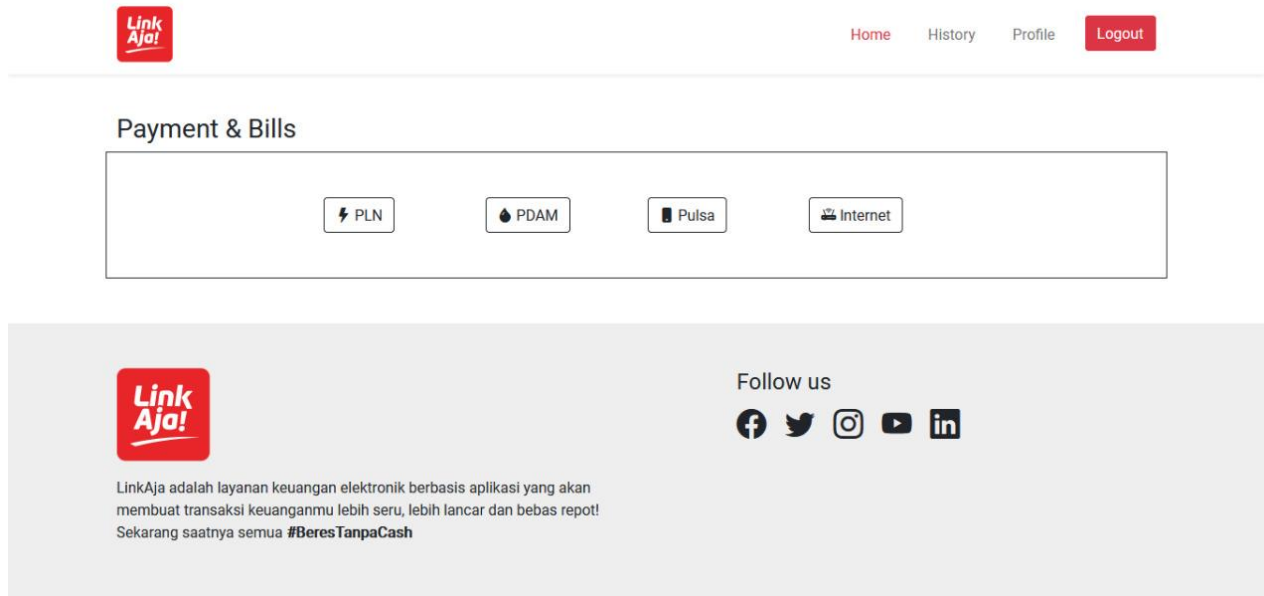
The screenshot shows the 'Send Money' page of the LinkAja application. At the top, there is a navigation bar with the LinkAja logo on the left and links for 'Home', 'History', 'Profile', and a red 'Logout' button on the right. Below the navigation bar is a red header section with the text 'Send Money' in white. The main content area contains a form with the following fields: 'Receiver' (a text input field with the placeholder '08xxxxxx'), 'Nominal' (a numeric input field with a decimal icon), 'Description' (a text input field with the placeholder 'Optional..'), and 'PIN' (a text input field with the placeholder 'Enter your PIN').

Cash Out, merupakan fitur untuk melakukan penarikan saldo LinkAja ke rekening bank yang dipilih, halaman cash out berisi form untuk memasukkan bank yang dipilih, nomor rekening, nama rekening, nominal, dan PIN.



The screenshot shows the 'Cash Out' page of the LinkAja application. At the top, there is a navigation bar with the LinkAja logo on the left and links for 'Home', 'History', 'Profile', and a red 'Logout' button on the right. Below the navigation bar is a red header section with the text 'Cash Out' in white. The main content area contains a form with the following fields: 'Bank' (a dropdown menu with 'Bank Mandiri' selected), 'No. Rekening' (a numeric input field with a decimal icon), 'Nama Rekening' (a text input field), 'Nominal' (a numeric input field with a decimal icon), and 'PIN' (a text input field).

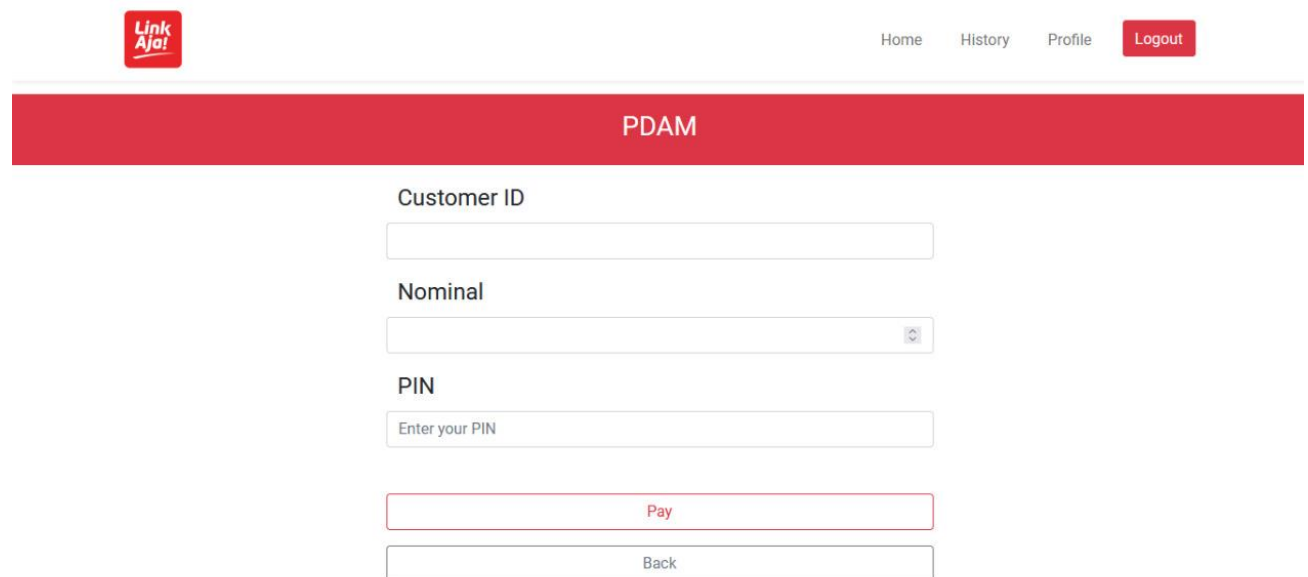
Payment and Bill, merupakan menu yang berisi layanan-layanan yang kami sediakan pada website LinkAja, layanan-layanan yang ada : PLN, PDAM, Pulsa, Internet.



PLN, berfungsi bagi pengguna yang akan melakukan pembayaran listrik dengan menggunakan saldo LinkAja, halaman PLN berisi form untuk memasukkan ID Pelanggan, nominal, dan PIN.

This screenshot shows the 'PLN' payment form on the LinkAja website. The top navigation bar is identical to the previous screenshot. Below the navigation bar, a red banner with the text 'PLN' is displayed. Underneath the banner, the form consists of the following elements: a label 'Customer ID' followed by a text input field; a label 'Nominal' followed by a text input field with a currency symbol icon on the right; a label 'PIN' followed by a text input field with the placeholder text 'Enter your PIN'; a red 'Pay' button; and a white 'Back' button.

PDAM, berguna untuk pengguna yang ingin melakukan pembayaran air dengan menggunakan saldo LinkAja, halaman PDAM berisi form untuk memasukkan ID Pelanggan, nominal, dan PIN.



LinkAja

Home History Profile Logout

PDAM

Customer ID

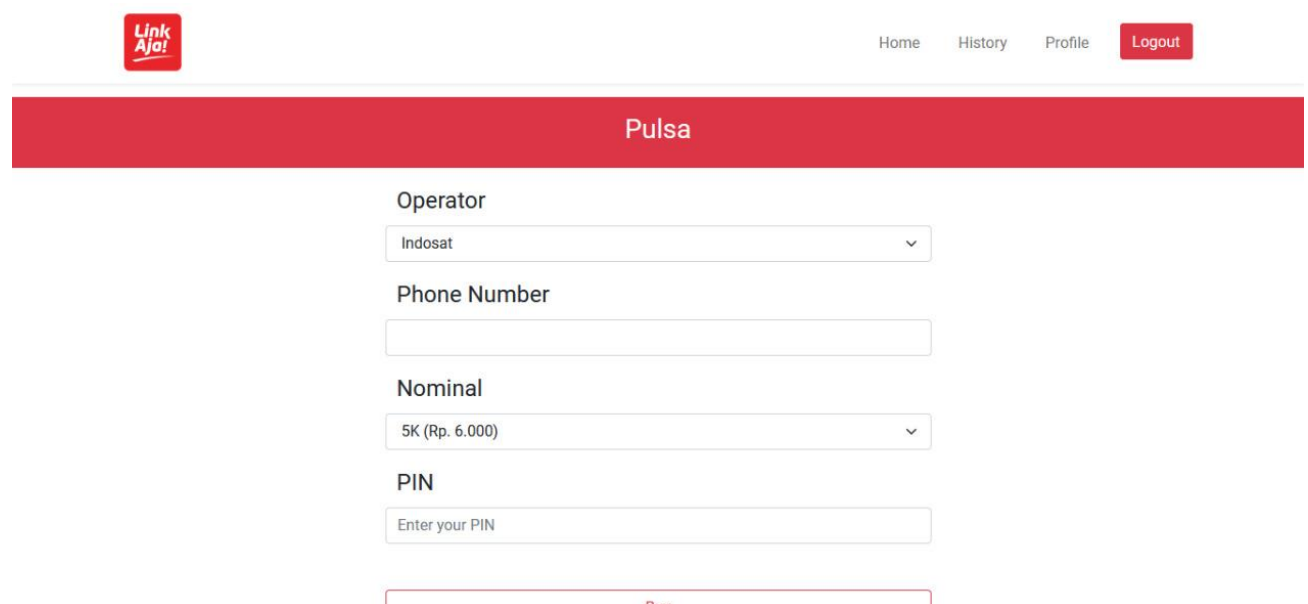
Nominal

PIN

Pay

Back

Pulsa, berfungsi bagi pengguna yang ingin mengisi pulsa dengan menggunakan saldo LinkAja, halaman pulsa berisi form untuk memilih operator pulsa yang diinginkan, nomor telepon, nominal, dan PIN anda.



LinkAja

Home History Profile Logout

Pulsa

Operator

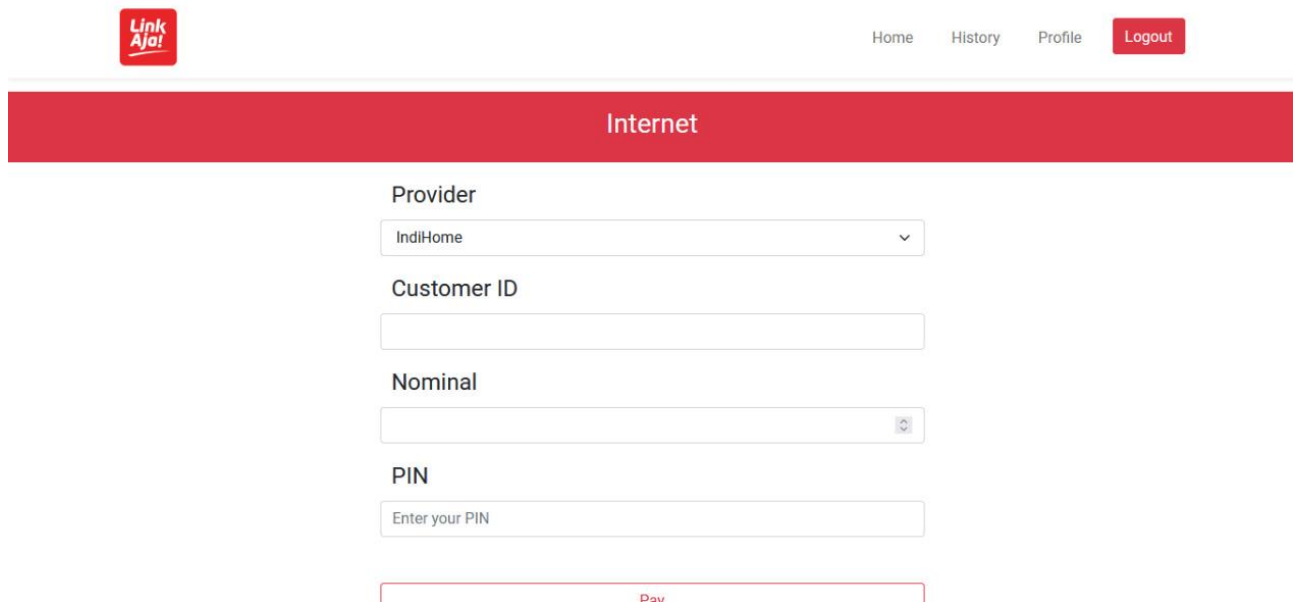
Phone Number

Nominal

PIN

Pay

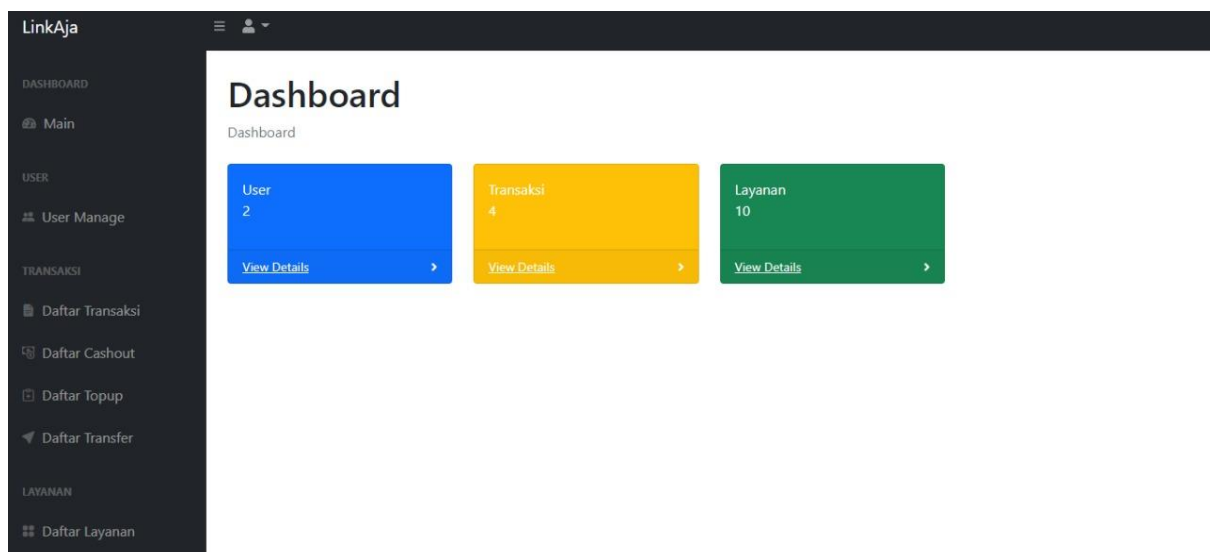
Internet, berguna bagi pengguna yang akan melakukan pembayaran internet menggunakan saldo LinkAja, halaman internet berisi form untuk memilih provider wifi yang akan dibayar, ID pelanggan, nominal, dan PIN yang dimiliki.



The screenshot shows the LinkAja 'Internet' payment interface. At the top, there is a navigation bar with the LinkAja logo and links for Home, History, Profile, and Logout. Below this is a red header with the word 'Internet'. The main form contains four input fields: 'Provider' (a dropdown menu currently showing 'IndiHome'), 'Customer ID' (a text input field), 'Nominal' (a text input field with a currency icon on the right), and 'PIN' (a text input field with the placeholder 'Enter your PIN'). At the bottom of the form, there is a red button labeled 'Pay'.

B. Tampilan Admin

Dashboard, agar admin dapat melihat jumlah data-data yang ada, pada halaman dashboard admin terdapat jumlah user, jumlah transaksi, dan jumlah layanan.



The screenshot displays the LinkAja Admin Dashboard. On the left is a dark sidebar with the LinkAja logo and a menu with categories: DASHBOARD (Main), USER (User Manage), TRANSAKSI (Daftar Transaksi, Daftar Cashout, Daftar Topup, Daftar Transfer), and LAYANAN (Daftar Layanan). The main content area is titled 'Dashboard' and shows three summary cards: 'User' with a count of 2, 'Transaksi' with a count of 4, and 'Layanan' with a count of 10. Each card has a 'View Details' link with a right-pointing arrow.

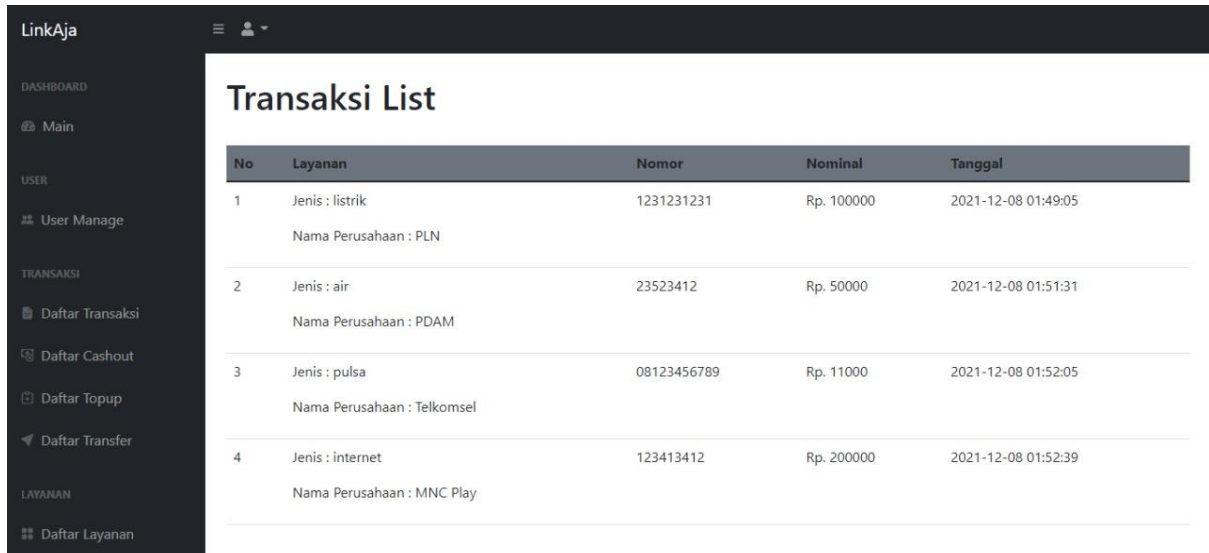
User Manage, berfungsi untuk mengatur akun-akun yang telah terdaftar, admin dapat menghapus akun juga dapat menekan tombol yang akan menuju ke riwayat transaksi.

No	Nama	Email	No Hp	Saldo	Aksi
1	Ivan Tandella	ivantandella@gmail.com	081381229986	624000 IDR	Hapus Riwayat Transaksi
2	Tsabitah Muflihza	tsabitah@gmail.com	08123456789	15000 IDR	Hapus Riwayat Transaksi

Riwayat Transaksi, admin dapat melihat riwayat transaksi yang ada pada setiap user/akun yang terdaftar, pada riwayat transaksi terdapat nomor, keterangan, nominal, dan tanggal dilakukannya transaksi.

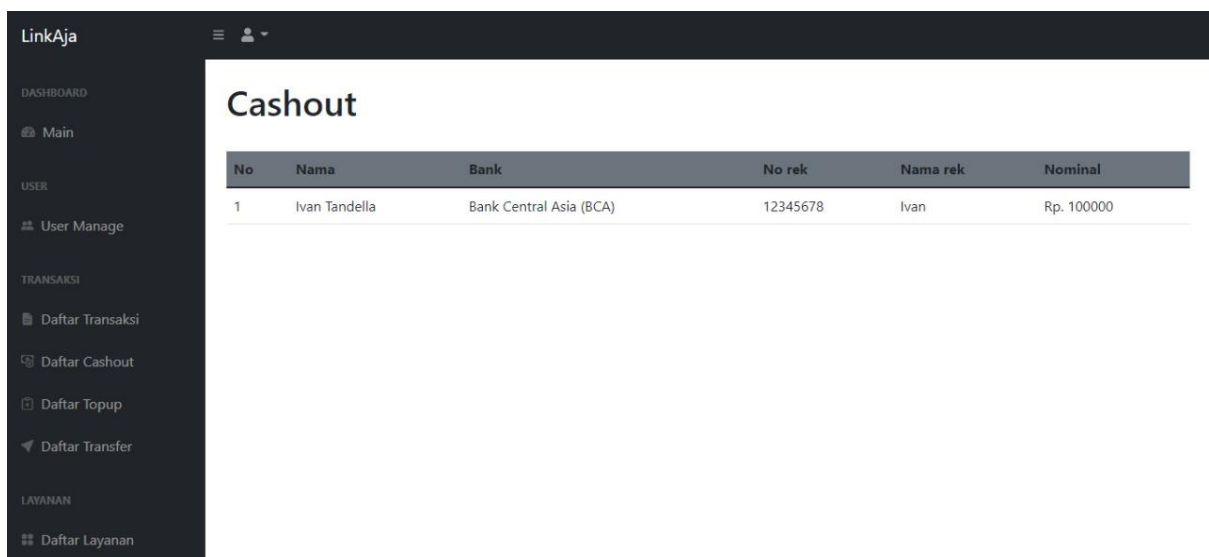
No	Keterangan	Nominal	Tanggal
1	Top-Up Rejected	Rp.50000	2021-12-08 02:43:14
2	Top-Up	Rp.100000	2021-12-08 02:38:56
3	Pay Internet 123413412	Rp.200000	2021-12-08 01:52:39
4	Buy Pulsa 08123456789	Rp.11000	2021-12-08 01:52:06
5	Pay PDAM 23523412	Rp.50000	2021-12-08 01:51:31
6	Pay PLN 1231231231	Rp.100000	2021-12-08 01:49:05
7	Send Money to 08123456789	Rp.5000	2021-12-08 01:45:08
8	Send Money to 08123456789	Rp.10000	2021-12-08 01:40:42
9	Cash Out to 12345678	Rp.100000	2021-12-08 01:35:13

Daftar Transaksi, berisi daftar transaksi yang terjadi, pada transaksi list terdapat nomor, nama layanan, nomor telepon, nominal, dan tanggal dilakukannya transaksi.



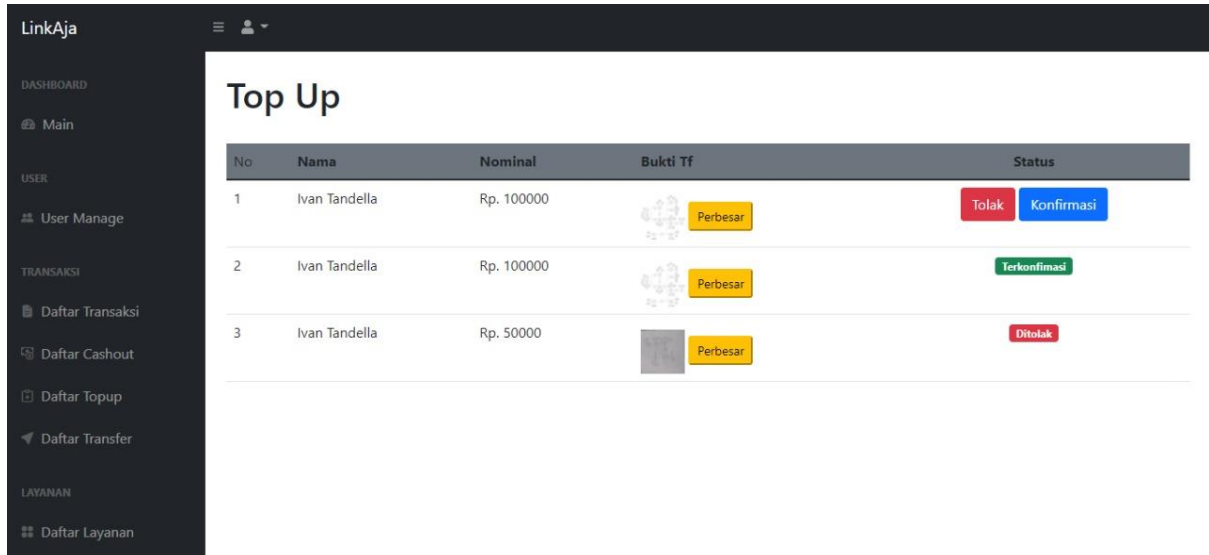
No	Layanan	Nomor	Nominal	Tanggal
1	Jenis : listrik Nama Perusahaan : PLN	1231231231	Rp. 100000	2021-12-08 01:49:05
2	Jenis : air Nama Perusahaan : PDAM	23523412	Rp. 50000	2021-12-08 01:51:31
3	Jenis : pulsa Nama Perusahaan : Telkomsel	08123456789	Rp. 11000	2021-12-08 01:52:05
4	Jenis : internet Nama Perusahaan : MNC Play	123413412	Rp. 200000	2021-12-08 01:52:39




Daftar Cashout, agar admin dapat melihat daftar cashout yang dilakukan oleh pengguna, pada halaman ini terdapat nomor, nama, nama bank, no rekening, nama rekening, dan nominal.



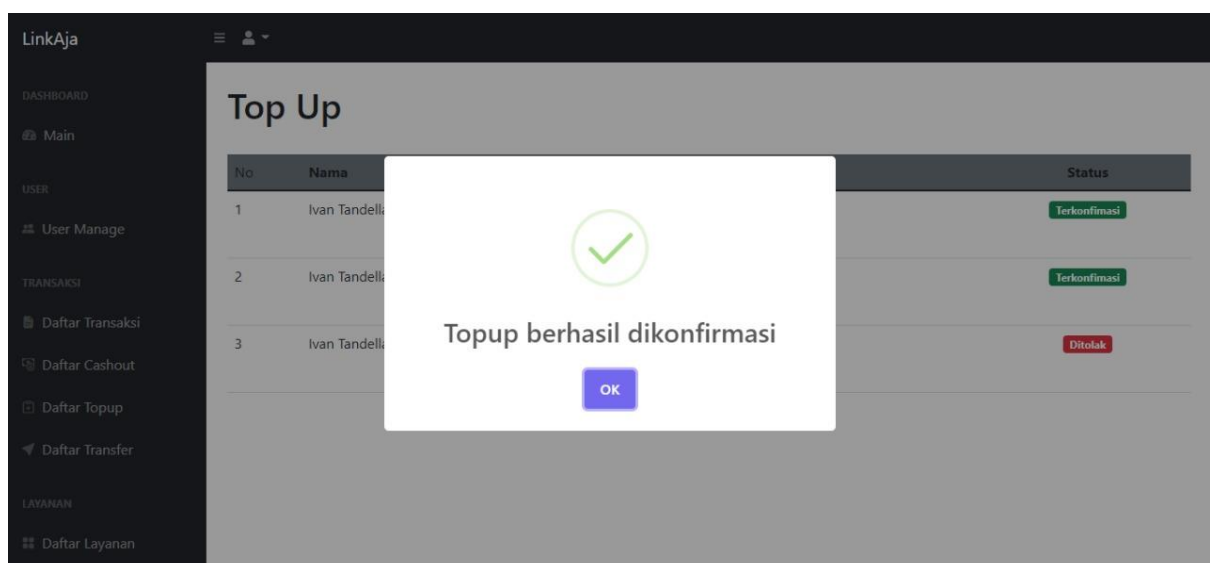
No	Nama	Bank	No rek	Nama rek	Nominal
1	Ivan Tandella	Bank Central Asia (BCA)	12345678	Ivan	Rp. 100000

Daftar Topup, untuk melihat daftar top up, juga untuk mengkonfirmasi dan melihat status top up yang dibuat oleh pengguna, pada halaman ini admin dapat melihat nomor, nama, nominal, dan bukti transfer, admin dapat menerima ataupun menolak request top up pengguna.

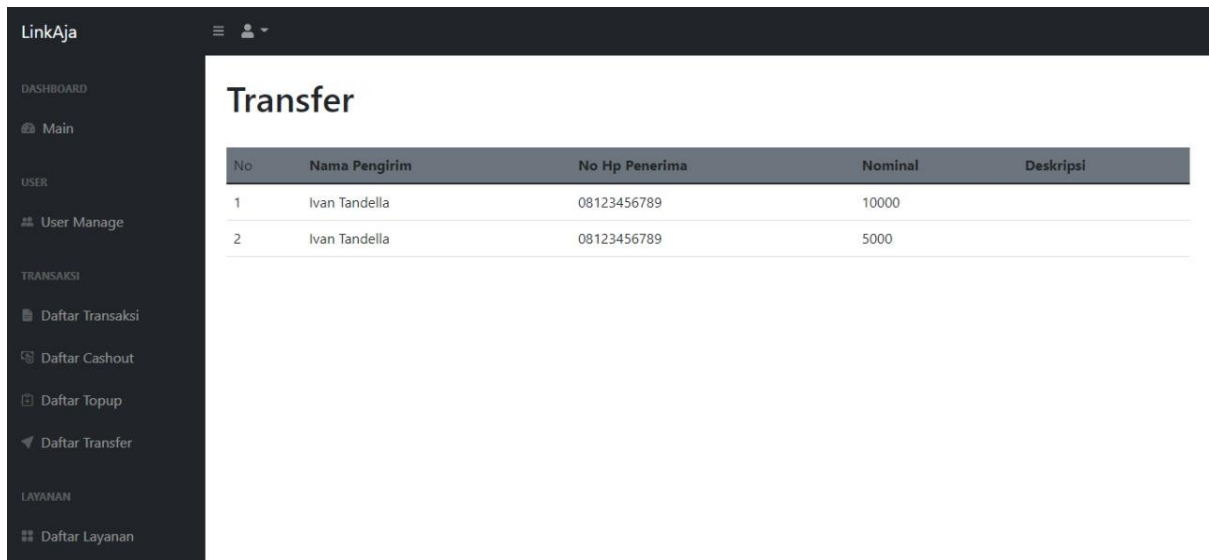


No	Nama	Nominal	Bukti Tf	Status
1	Ivan Tandella	Rp. 100000	 Perbesar	Tolak Konfirmasi
2	Ivan Tandella	Rp. 100000	 Perbesar	Terkonfirmasi
3	Ivan Tandella	Rp. 50000	 Perbesar	Ditolak

Top Up Confirmation Notif, notifikasi dibawah ini akan muncul ketika admin mengkonfirmasi request top up dari pengguna.

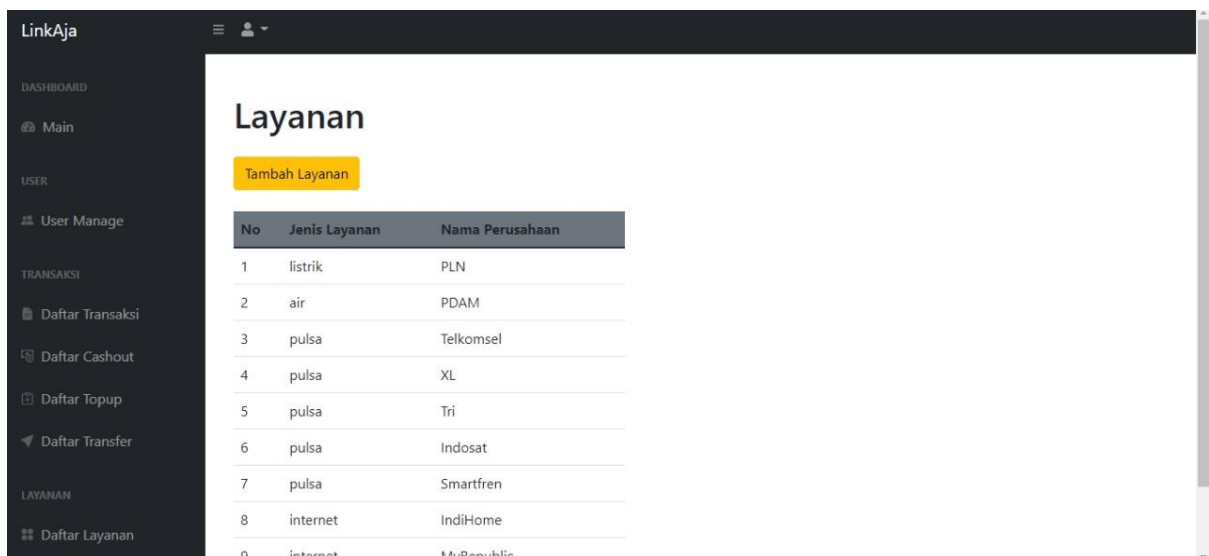


Daftar Transfer, untuk melihat list transfer atau send money yang dilakukan oleh pengguna, pada halaman transfer terdapat no, nama pengirim, nomor telepon penerima, nominal, dan deskripsi.



No	Nama Pengirim	No Hp Penerima	Nominal	Deskripsi
1	Ivan Tandella	08123456789	10000	
2	Ivan Tandella	08123456789	5000	

Daftar layanan, merupakan sebuah halaman yang berisi daftar layanan, pada halaman ini juga terdapat tombol tambah layanan untuk melakukan penambahan pada layanan yang tersedia.



No	Jenis Layanan	Nama Perusahaan
1	listrik	PLN
2	air	PDAM
3	pulsa	Telkomsel
4	pulsa	XL
5	pulsa	Tri
6	pulsa	Indosat
7	pulsa	Smartfren
8	internet	IndiHome
9	internet	MvReoublic

Tambah Layanan, merupakan sebuah form untuk menulis atau mengisi penambahan layanan yang akan ditambahkan, layanan yang dapat ditambahkan adalah pulsa dan internet, admin juga harus mengisi nama perusahaan.

The screenshot shows the LinkAja admin interface. On the left is a sidebar with navigation links: DASHBOARD, Main, USER, User Manage, TRANSAKSI, Daftar Transaksi, Daftar Cashout, Daftar Topup, Daftar Transfer, LAYANAN, and Daftar Layanan. The main content area is titled 'Layanan' and contains a table of services. A modal window titled 'Tambah Layanan' is open, allowing the addition of a new service. The modal includes a dropdown menu for 'Jenis layanan' (currently set to 'Pulsa'), a text input field for 'Nama Perusahaan', and a blue 'Tambah Layanan' button.

No	Jenis Layanan
1	listrik
2	air
3	pulsa
4	pulsa
5	pulsa
6	pulsa
7	pulsa
8	internet
9	internet

2.6 Query

Admin

Query dibawah ini merupakan query yang digunakan untuk mengambil seluruh data dari tabel user, tabel pembayaran, dan tabel layanan yang selanjutnya akan ditampilkan pada dashboard admin menggunakan method count() untuk mendapatkan jumlah record data pada masing-masing tabel.

```
class AdminController extends Controller
{
    public function index()
    {
        $title = "Dashboard";
        $users = user::all();
        // SELECT * FROM users;

        $transactions = pembayaran::all();
        // SELECT * FROM pembayaran;

        $layanan = layanan::all();
        // SELECT * FROM layanan;

        return view('admin.dashboard', ['title' => $title, 'users' => $users, 'transactions' => $transactions, 'layanan' => $layanan]);
    }
}
```

Query dibawah ini merupakan sebuah query yang digunakan untuk mengambil seluruh data dari tabel user untuk ditampilkan pada menu user setting pada bagian admin.

```

public function user()
{
    $title = "User Setting";
    $users = user::all();
    // SELECT * FROM users;

    $trash = user::onlyTrashed()->get();
    return view('admin.user', ['users' => $users, 'title' => $title, 'trash' => $trash]);
}

```

Query berikut ini merupakan query yang digunakan untuk menghapus user berdasarkan id yang ditentukan.

```

public function deleteUser($id)
{
    $user = user::find($id);
    $user->delete();
    // DELETE FROM users WHERE id = $id;

    return redirect()->back();
}

```

Query berikut ini merupakan sebuah query yang digunakan untuk mengambil seluruh data dari tabel pembayaran untuk ditampilkan pada menu Transaksi pada bagian admin.

```

public function transaksi()
{
    $title = "Transaksi";
    $transactions = pembayaran::all();
    // SELECT * FROM pembayaran;

    return view("admin.transaksi", ['transactions' => $transactions, 'title' => $title]);
}

```

Query dibawah ini adalah sebuah query yang digunakan untuk mengambil seluruh data dari tabel layanan untuk ditampilkan pada menu Daftar Layanan.

```

public function layanan()
{
    $title = "Daftar Layanan";
    $layanan = layanan::all();
    // SELECT * FROM layanan;

    return view("admin.layanan", ['title' => $title, 'layanan' => $layanan]);
}

```

Query berikut merupakan query yang digunakan pada fitur Tambah Layanan, query ini fungsinya untuk memasukkan data kedalam tabel layanan, tepatnya pada kolom jenis_layanan dan nama_perusahaan yang nilainya diambil dari sebuah isian form yang disediakan dan telah diisi oleh admin.

```
public function tambahLayanan(Request $keyword)
{
    layanan::create([
        'jenis_layanan' => $keyword->jenis_layanan,
        'nama_perusahaan' => $keyword->nama_perusahaan,
    ]);
    // INSERT INTO layanan (jenis_layanan, nama_perusahaan) VALUES ($keyword->jenis_layanan, $keyword->nama_perusahaan);

    return redirect()->back();
}
```

Query dibawah ini merupakan sebuah query yang digunakan untuk mengambil seluruh data dari tabel cash_out yang akan ditampilkan pada menu Daftar Cashout di halaman admin.

```
public function cashout()
{
    $title = "Daftar Cashout";
    $cashout = CashOut::all();
    // SELECT * FROM cash_out;

    return view("admin.cashout", ['title' => $title, 'cashouts' => $cashout]);
}
```

Query ini adalah sebuah query yang digunakan dalam mengambil seluruh data dari tabel top_up yang akan ditampilkan pada menu Daftar Top up pada halaman admin.

```
public function topup()
{
    $title = "Daftar Cashout";
    $topup = topup::all();
    // SELECT * FROM top_up;

    return view("admin.topup", ['title' => $title, 'topups' => $topup]);
}
```

Query ini merupakan sebuah query yang digunakan pada fitur konfirmasi top up, query pertama berfungsi untuk mengambil data dari tabel top_up berdasarkan id top_up yang ditentukan, kemudian query yang kedua digunakan untuk mengupdate statusnya menjadi 1

pada id yang telah ditentukan. Query yang ketiga berguna untuk mengambil data dari tabel user berdasarkan id yang dipilih, dan yang terakhir query keempat digunakan untuk meng-update saldo user.

```
public function konfirmasiTopup($id)
{
    $data = TopUp::where('id_top_up', $id)->first();
    // SELECT * FROM top_up WHERE id_top_up = $id;

    topup::where('id_top_up', $id)
        ->update([
            'status' => "1"
        ]);
    // UPDATE top_up SET status = '1' WHERE id_top_up = $id;

    $user = User::where('id', $data->id_user)->first();
    // SELECT * FROM users WHERE id = $data->id_user;

    User::where('id', $data->id_user)->update([
        'saldo' => $user->saldo + $data->nominal
    ]);
    // UPDATE users SET saldo = $user->saldo + $data->nominal WHERE id = $data->id_user;
```

Query dibawah ini merupakan query yang digunakan untuk memasukkan data kedalam tabel history, tepatnya pada kolom id_user, nominal, dan keterangan.

```
History::create([
    'id_user' => $user->id,
    'nominal' => $data->nominal,
    'keterangan' => 'Top-Up',
]);
// INSERT INTO history (id_user, nominal, keterangan) VALUES ($user->id, $data->nominal, 'Top-Up');

return back()->with('success', 'Topup berhasil dikonfirmasi');
```

Query dibawah ini merupakan sebuah query yang digunakan pada fitur tolak top up, query yang pertama berfungsi untuk mengambil data dari tabel top_up berdasarkan id top_up yang ditentukan, kemudian untuk query yang kedua digunakan untuk mengupdate statusnya menjadi 2 pada id yang telah ditentukan. Query yang ketiga berguna untuk mengambil data dari tabel user berdasarkan id yang dipilih, dan yang terakhir query keempat digunakan untuk memasukkan data kedalam tabel history, tepatnya pada kolom id_user, nominal, dan keterangan.

```

public function tolakTopup($id)
{
    $data = TopUp::where('id_top_up', $id)->first();
    // SELECT * FROM top_up WHERE id_top_up = $id;

    topup::where('id_top_up', $id)
        ->update([
            'status' => "2"
        ]);
    // UPDATE top_up SET status = '2' WHERE id_top_up = $id;

    $user = User::where('id', $data->id_user)->first();
    // SELECT * FROM users WHERE id = $data->id_user;

    History::create([
        'id_user' => $user->id,
        'nominal' => $data->nominal,
        'keterangan' => 'Top-Up Rejected',
    ]);
    // INSERT INTO history (id_user, nominal, keterangan) VALUES ($user->id, $data->nominal, 'Top-Up Rejected');

    return back()->with('success', 'Topup berhasil ditolak');
}

```

Query berikut ini merupakan sebuah query yang digunakan dalam mengambil seluruh data dari tabel transfer yang akan ditampilkan pada menu transfer yang terdapat pada halaman admin.

```

public function transfer()
{
    $title = 'Transfer';
    $transfer = transfer::all();
    // SELECT * FROM transfer;

    return view('admin.transfer', ['title' => $title, 'transfers' => $transfer]);
}

```

Query ini berfungsi untuk mengambil data dari tabel history berdasarkan id user kemudian diurutkan berdasarkan kolom created_at dari data yang paling baru hingga data yang paling lama.

```

public function history($id)
{
    $title = 'Riwayat';
    $histories = History::where('id_user', $id)->orderBy('created_at', 'desc')->get();
    // SELECT * FROM history WHERE id_user = $id ORDER BY created_at DESC;

    return view('admin.history', compact('histories', 'title'));
}

```


Cash-Out

Query ini berfungsi untuk mengambil data dari tabel bank berdasarkan kolom nama_bank yang terdapat pada tabel Bank dan kemudian diurutkan berdasarkan abjad.

```
public function index()
{
    $user = auth()->user();
    $banks = Bank::orderBy('nama_bank', 'asc')->get();
    // SELECT * FROM bank ORDER BY nama_bank ASC;

    if (!$user->pin) {
        return redirect(route('pin', $user->id))->with('info', 'Please create your PIN!');
    }

    return view('cashout', compact('user', 'banks'));
}
```

Query dibawah ini merupakan sebuah query yang digunakan untuk memasukkan data kedalam tabel cash_out, tepatnya pada kolom id_bank, no_rekening, nama_rekening, dan nominal. Kemudian query yang kedua juga kurang lebih sama yaitu memasukkan data kedalam tabel history, yaitu pada kolom id_user, nominal, dan keterangan. Yang terakhir query ketiga berfungsi untuk meg-update saldo user.

```
if (Hash::check($request->pin, $user->pin)) {
    CashOut::create([
        'id_user' => $user->id,
        'id_bank' => $request->id_bank,
        'no_rekening' => $request->no_rekening,
        'nama_rekening' => $request->nama_rekening,
        'nominal' => $request->nominal,
    ]);
    // INSERT INTO cash_out (id_bank, no_rekening, nama_rekening, nominal) VALUES ($request->id_bank, $request->no_rekening,
    $request->nama_rekening, $request->nominal);

    History::create([
        'id_user' => $user->id,
        'nominal' => $request->nominal,
        'keterangan' => 'Cash Out to ' . $request->no_rekening,
    ]);
    // INSERT INTO history (id_user, nominal, keterangan) VALUES ($user->id, $request->nominal, 'Cash Out to $request->no_rekening');

    User::where('id', $id)->update([
        'saldo' => $user->saldo - $request->nominal
    ]);
    // UPDATE users SET saldo = $user->saldo - $request->nominal WHERE id = $id;

    return back()->with('success', 'Berhasil!');
} else {
    return back()->with('warning', 'Wrong PIN!');
}
```

History user

Query ini berfungsi untuk mengambil data dari tabel history berdasarkan id user yang ditentukan, kemudian diurutkan berdasarkan kolom created_at dari data yang paling baru hingga data yang paling lama.

```
public function index()
{
    $histories = History::where('id_user', auth()->user()->id)->orderBy('created_at', 'desc')->get();
    // SELECT * FROM history WHERE id_user = auth()->user()->id ORDER BY created_at DESC;

    return view('history', compact('histories'));
}
```

Top up

Query dibawah ini merupakan query yang digunakan untuk memasukkan data kedalam tabel top_up, yaitu pada kolom id_user, nominal, dan bukti_tf.

```
TopUp::create([
    'id_user' => $request->id_user,
    'nominal' => $request->nominal,
    'bukti_tf' => $request->file('bukti_tf')->store('topup'),
]);
// INSERT INTO top_up (id_user, nominal, bukti_tf) VALUES ($request->id_user, $request->nominal,
$request->bukti_tf)
```

Pembayaran

Query berikut merupakan sebuah query yang digunakan untuk mengambil seluruh data dari tabel layanan yang nama_perusahaannya adalah PLN.

```
public function pln()
{
    $user = auth()->user();
    $layanan = Layanan::where('nama_perusahaan', 'PLN')->first();
    // SELECT * FROM layanan WHERE nama_perusahaan = 'PLN';

    if (!$user->pin) {
        return redirect(route('pin', $user->id))->with('info', 'Please create your PIN!');
    }

    return view('pln', compact('user', 'layanan'));
}
```

Query dibawah ini berguna untuk mengambil seluruh data dari tabel user berdasarkan id user yang ditentukan.

```
public function payPLN(Request $request, $id)
{
    $request->validate([
        'nomor' => 'required',
        'nominal' => 'required|numeric',
        'pin' => 'required|size:6',
    ]);

    $user = User::where('id', $id)->first();
    // SELECT * FROM users WHERE id = $id;

    if ($user->saldo < $request->nominal) {
        return back()->with('warning', 'Saldo anda tidak cukup!');
    }
}
```

Query dibawah ini merupakan query yang digunakan untuk memasukkan data kedalam tabel pembayaran, tepatnya pada kolom id_user, id_layanan, nomor, dan nominal. Dan query yang kedua digunakan untuk meng-update saldo user

```
if (Hash::check($request->pin, $user->pin)) {
    Pembayaran::create([
        'id_user' => $user->id,
        'id_layanan' => $request->id_layanan,
        'nomor' => $request->nomor,
        'nominal' => $request->nominal,
    ]);
    // INSERT INTO pembayaran (id_user, id_layanan, nomor, nominal) VALUES ($user->id, $request->id_layanan, $request->nomor, $request->nominal);

    User::where('id', $id)->update([
        'saldo' => $user->saldo - $request->nominal
    ]);
    // UPDATE users SET saldo = $user->saldo - $request->nominal WHERE id = $id;
}
```

Query berikut merupakan sebuah query yang digunakan untuk memasukkan data kedalam tabel history, tepatnya pada kolom id_user, nominal, dan keterangan.

```
History::create([
    'id_user' => $user->id,
    'nominal' => $request->nominal,
    'keterangan' => 'Pay PLN ' . $request->nomor,
]);
// INSERT INTO history (id_user, nominal, keterangan) VALUES ($user->id, $request->nominal, 'Pay PLN $request->nomor');

return back()->with('success', 'Berhasil melakukan pembayaran!');
} else {
    return back()->with('warning', 'Wrong PIN!');
}
```


Query dibawah ini digunakan untuk mengambil seluruh data dari tabel layanan yang nama_perusahaan adalah PDAM.

```
public function pdam()
{
    $user = auth()->user();
    $layanan = Layanan::where('nama_perusahaan', 'PDAM')->first();
    // SELECT * FROM layanan WHERE nama_perusahaan = 'PDAM';

    if (!$user->pin) {
        return redirect(route('pin', $user->id))->with('info', 'Please create your PIN!');
    }

    return view('pdam', compact('user', 'layanan'));
}
```

Query dibawah ini berguna untuk mengambil seluruh data dari tabel user berdasarkan id user yang telah dipilih.

```
public function payPDAM(Request $request, $id)
{
    $request->validate([
        'nomor' => 'required',
        'nominal' => 'required|numeric',
        'pin' => 'required|size:6',
    ]);

    $user = User::where('id', $id)->first();
    // SELECT * FROM users WHERE id = $id;

    if ($user->saldo < $request->nominal) {
        return back()->with('warning', 'Saldo anda tidak cukup!');
    }
}
```

Query dibawah ini merupakan sebuah query yang digunakan untuk memasukkan data kedalam tabel pembayaran, tepatnya pada kolom id_user, id_layanan, nomor, dan nominal. Kemudian query yang kedua berfungsi untuk meg-update saldo user. Query yang ketiga juga kurang lebih sama yaitu memasukkan data kedalam tabel history, yaitu pada kolom id_user, nominal, dan keterangan.

```

if (Hash::check($request->pin, $user->pin)) {
    Pembayaran::create([
        'id_user' => $user->id,
        'id_layanan' => $request->id_layanan,
        'nomor' => $request->nomor,
        'nominal' => $request->nominal,
    ]);
    // INSERT INTO pembayaran (id_user, id_layanan, nomor, nominal) VALUES ($user->id, $request->id_layanan, $request->nomor, $request->nominal);

    User::where('id', $id)->update([
        'saldo' => $user->saldo - $request->nominal
    ]);
    // UPDATE users SET saldo = $user->saldo - $request->nominal WHERE id = $id;

    History::create([
        'id_user' => $user->id,
        'nominal' => $request->nominal,
        'keterangan' => 'Pay PDAM ' . $request->nomor,
    ]);
    // INSERT INTO history (id_user, nominal, keterangan) VALUES ($user->id, $request->nominal, 'Pay PDAM $request->nomor');

    return back()->with('success', 'Berhasil melakukan pembayaran!');
} else {
    return back()->with('warning', 'Wrong PIN!');
}

```

Query dibawah ini berguna untuk mengambil seluruh data dari tabel layanan yang jenis_layanan adalah pulsa dan diurutkan berdasarkan nama_perusahaan.

```

public function pulsa()
{
    $user = auth()->user();
    $layanan = Layanan::where('jenis_layanan', 'pulsa')->orderBy('nama_perusahaan')->get();
    // SELECT * FROM layanan WHERE jenis_layanan = 'pulsa' ORDER BY nama_perusahaan;

    if (!$user->pin) {
        return redirect(route('pin', $user->id))->with('info', 'Please create your PIN!');
    }

    return view('pulsa', compact('user', 'layanan'));
}

```

Query berikut merupakan query yang berguna untuk mengambil seluruh data dari tabel user berdasarkan id yang telah dipilih.

```

public function payPulsa(Request $request, $id)
{
    $request->validate([
        'id_layanan' => 'required',
        'nomor' => 'required|numeric',
        'nominal' => 'required',
        'pin' => 'required|size:6',
    ]);

    $user = User::where('id', $id)->first();
    // SELECT * FROM users WHERE id = $id;

    if ($user->saldo < $request->nominal) {
        return back()->with('warning', 'Saldo anda tidak cukup!');
    }
}

```

Query dibawah ini merupakan sebuah query yang digunakan untuk memasukkan data kedalam tabel pembayaran, yaitu pada kolom id_user, id_layanan, nomor, dan nominal. Query yang kedua digunakan untuk meng-update saldo pengguna. Query ketiga berfungsi untuk memasukkan data kedalam tabel history, yaitu pada kolom id_user, nominal, dan keterangan.

```
if (Hash::check($request->pin, $user->pin)) {
    Pembayaran::create([
        'id_user' => $user->id,
        'id_layanan' => $request->id_layanan,
        'nomor' => $request->nomor,
        'nominal' => $request->nominal,
    ]);
    // INSERT INTO pembayaran (id_user, id_layanan, nomor, nominal) VALUES ($user->id, $request->id_layanan, $request->nomor, $request->nominal);

    User::where('id', $id)->update([
        'saldo' => $user->saldo - $request->nominal
    ]);
    // UPDATE users SET saldo = $user->saldo - $request->nominal WHERE id = $id;

    History::create([
        'id_user' => $user->id,
        'nominal' => $request->nominal,
        'keterangan' => 'Buy Pulsa ' . $request->nomor,
    ]);
    // INSERT INTO history (id_user, nominal, keterangan) VALUES ($user->id, $request->nominal, 'Buy Pulsa $request->nomor');

    return back()->with('success', 'Berhasil!');
} else {
    return back()->with('warning', 'Wrong PIN!');
}
```

Query dibawah ini digunakan untuk mengambil seluruh data dari tabel layanan yang jenis_layanan adalah internet dan kemudia diurutkan berdasarkan nama_perusahaan.

```
public function internet()
{
    $user = auth()->user();
    $layanan = Layanan::where('jenis_layanan', 'internet')->orderBy('nama_perusahaan')->get();
    // SELECT * FROM layanan WHERE jenis_layanan = 'internet' ORDER BY nama_perusahaan;

    if (!$user->pin) {
        return redirect(route('pin', $user->id))->with('info', 'Please create your PIN!');
    }

    return view('internet', compact('user', 'layanan'));
}
```

Query berikut merupakan query yang berguna untuk mengambil seluruh data dari tabel user berdasarkan id yang telah ditentukan.

```
public function payInternet(Request $request, $id)
{
    $request->validate([
        'id_layanan' => 'required',
        'nomor' => 'required|numeric',
        'nominal' => 'required',
        'pin' => 'required|size:6',
    ]);

    $user = User::where('id', $id)->first();
    // SELECT * FROM users WHERE id = $id;

    if ($user->saldo < $request->nominal) {
        return back()->with('warning', 'Saldo anda tidak cukup!');
    }
}
```

Query berikut ini merupakan sebuah query yang digunakan untuk memasukkan data kedalam tabel pembayaran, tepatnya pada kolom id_user, id_layanan, nomor, dan nominal. Kemudian query yang kedua digunakan untuk meng-update saldo pengguna berdasarkan id yang dipilih. Dan terakhir query ketiga berfungsi untuk memasukkan data kedalam tabel history, yaitu pada kolom id_user, nominal, dan keterangan.

```
if (Hash::check($request->pin, $user->pin)) {
    Pembayaran::create([
        'id_user' => $user->id,
        'id_layanan' => $request->id_layanan,
        'nomor' => $request->nomor,
        'nominal' => $request->nominal,
    ]);
    // INSERT INTO pembayaran (id_user, id_layanan, nomor, nominal) VALUES ($user->id, $request->id_layanan, $request->nomor, $request->nominal);

    User::where('id', $id)->update([
        'saldo' => $user->saldo - $request->nominal
    ]);
    // UPDATE users SET saldo = $user->saldo - $request->nominal WHERE id = $id;

    History::create([
        'id_user' => $user->id,
        'nominal' => $request->nominal,
        'keterangan' => 'Pay Internet ' . $request->nomor,
    ]);
    // INSERT INTO history (id_user, nominal, keterangan) VALUES ($user->id, $request->nominal, 'Pay Internet $request->nomor');

    return back()->with('success', 'Berhasil!');
} else {
    return back()->with('warning', 'Wrong PIN!');
}
```

User

Query berikut adalah sebuah query yang digunakan untuk mengambil seluruh data dari tabel user berdasarkan id yang telah ditentukan.

```
public function index()
{
    $user = User::where('id', auth()->user()->id)->first();
    // SELECT * FROM users WHERE id = auth()->user()->id;

    return view('profile', compact('user'));
}
```

Query dibawah ini merupakan sebuah query yang digunakan untuk mengambil seluruh data dari tabel user berdasarkan id yang dipilih.

```
public function pin($id)
{
    $user = User::where('id', $id)->first();
    // SELECT * FROM users WHERE id = $id;

    return view('pin', compact('user'));
}
```

Query berikut adalah sebuah query yang digunakan untuk mengambil seluruh data dari tabel user berdasarkan id yang dipilih.

```
public function updatePIN(Request $request, $id)
{
    $user = User::where('id', $id)->first();
    // SELECT * FROM users WHERE id = $id;

    if ($user->pin) {
        $request->validate([
            'oldPin' => 'required|size:6',
            'pin' => 'required|size:6|confirmed',
            'pin_confirmation' => 'required|size:6',
        ]);

        $currentPin = $user->pin;
        $oldPin = $request->oldPin;
    }
}
```


Query dibawah ini berguna untuk meng-update pin user pada menu update Pin sesuai dengan id yang telah dipilih jika pin-nya belum ada, kemudian untuk query yang kedua digunakan untuk membuat pin yang baru.

```
if (Hash::check($oldPin, $currentPin)) {
    $user->update([
        'pin' => Hash::make($request->pin)
    ]);
    // UPDATE users SET pin = $request->pin WHERE id = $id;

    return redirect(route('profile'))->with('success', 'PIN updated!');
} else {
    return back()
        ->withErrors(['oldPin' => 'Wrong pin'])
        ->with('warning', 'Wrong PIN!');
}
} else {
    $request->validate([
        'pin' => 'required|size:6|confirmed',
        'pin_confirmation' => 'required|size:6',
    ]);

    $user->update([
        'pin' => Hash::make($request->pin)
    ]);
    // UPDATE users SET pin = $request->pin WHERE id = $id;

    return redirect(route('profile'))->with('success', 'PIN updated!');
}
```

Transfer

Query dibawah ini merupakan sebuah query yang digunakan untuk mengambil seluruh data dari tabel user berdasarkan id yang ditentukan.

```
public function send(Request $request, $id)
{
    $request->validate([
        'no_hp_penerima' => 'required|numeric|exists:users,no_hp',
        'nominal' => 'required|numeric',
        'pin' => 'required|size:6',
    ]);

    $pengirim = User::where('id', $id)->first();
    // SELECT * FROM users WHERE id = $id;

    if ($pengirim->saldo < $request->nominal) {
        return back()->with('warning', 'Saldo anda tidak cukup!');
    }
}
```

Query berikut ini merupakan sebuah query yang digunakan untuk memasukkan data kedalam tabel transfer, tepatnya pada kolom no_hp_penerima, nominal, deskripsi, dan id_user. Kemudian query yang kedua digunakan untuk memasukkan data kedalam tabel history, yaitu pada kolom id_user, nominal, dan keterangan.

```

if (Hash::check($request->pin, $pengirim->pin)) {
    Transfer::create([
        'no_hp_penerima' => $request->no_hp_penerima,
        'nominal' => $request->nominal,
        'deskripsi' => $request->deskripsi,
        'id_user' => auth()->user()->id
    ]);
    // INSERT INTO transfer (no_hp_penerima, nominal, deskripsi, id_user) VALUES ($request->no_hp_penerima, $request->nominal,
    $request->deskripsi, auth()->user()->id);

    History::create([
        'id_user' => auth()->user()->id,
        'nominal' => $request->nominal,
        'keterangan' => 'Send Money to ' . $request->no_hp_penerima,
    ]);
    // INSERT INTO history (id_user, nominal, keterangan) VALUES (auth()->user()->id, $request->nominal, 'Send Money to
    $request->no_hp_penerima');

    User::where('id', $id)->update([
        'saldo' => $pengirim->saldo - $request->nominal
    ]);
    // UPDATE users SET saldo = $pengirim->saldo - $request->nominal WHERE id = $id;

    $user = User::where('no_hp', $request->no_hp_penerima)->first();
    // SELECT * FROM users WHERE no_hp = $request->no_hp_penerima;

    $penerima = User::where('id', $user->id)->first();
    // SELECT * FROM users WHERE id = $user->id;

```

Query dibawah ini merupakan query yang digunakan untuk meng-update saldo pengguna berdasarkan id yang telah ditentukan, kemudian query yang kedua merupakan query yang berfungsi untuk memasukkan data kedalam tabel history, tepatnya pada id_user, nominal, dan keterangan.

```

User::where('id', $user->id)->update([
    'saldo' => $penerima->saldo + $request->nominal
]);
// UPDATE users SET saldo = $penerima->saldo + $request->nominal WHERE id = $id;

History::create([
    'id_user' => $user->id,
    'nominal' => $request->nominal,
    'keterangan' => 'Received Money',
]);
// INSERT INTO history (id_user, nominal, keterangan) VALUES (auth()->user()->id, $request->nominal, 'Received Money');

return back()->with('success', 'Berhasil melakukan transfer!');
} else {
    return back()->with('warning', 'Wrong PIN!');
}

```

BAB III

PENUTUP

3.1 Kesimpulan

Kesimpulan dari laporan tugas besar yang kami kerjakan ini adalah dari semua deskripsi dan tampilan dari Website LinkAja yang telah kami selesaikan, maka dapat disimpulkan bahwa website ini dapat berjalan dan digunakan dengan baik dan mudah, pengaplikasian databasenya tidak terlalu sulit namun tetap memerlukan logika yang akurat agar dapat menghasilkan sebuah aplikasi yang baik dan mudah untuk digunakan.