

BAB I

Pendahuluan

A. Latar Belakang

Sistem keuangan merupakan struktur, proses, dan mekanisme yang terlibat dalam pengelolaan dan aliran dana dalam suatu lembaga, entitas, organisasi. Sistem keuangan yang dirancang untuk mengatur aspek keuangan internal dan eksternal, termasuk pengumpulan data, pengeluaran, pengelolaan anggaran, pelaporan keuangan, dan pengendalian keuangan. Sistem keuangan didirikan sebagai sarana pengelolaan data keuangan yang terkomputerisasi sehingga data keuangan dapat terarah dan dikelola secara efektif dan menjaga stabilitas keuangan. Selain itu, penyimpanan data dalam bentuk elektronik dapat memfasilitasi pengelolaan dan penyediaan data yang lebih cepat.

Lazismu merupakan lembaga zakat nasional yang bertujuan untuk memberdayakan masyarakat dengan memanfaatkan dana zakat, infak, wakaf, dan sumbangan lainnya baik dari individu, lembaga, perusahaan, maupun instansi yang berada di Yogyakarta. Namun, saat ini pengelolaan sistem keuangan dan pembukuan di Lazismu masih dilakukan secara manual menggunakan aplikasi Microsoft Excel. Hal ini menyebabkan proses tersebut menjadi tidak efektif bagi admin keuangan Lazismu, serta dapat mengalami kesalahan input data karena adanya *human error* dari admin.

Melihat permasalahan tersebut, maka dibuatlah Sistem Keuangan Lazismu yang memiliki fungsi untuk mengelola data keuangan terkait zakat, infak, dan wakaf yang didapatkan dari pemberi (muzakki) dan yang nantinya akan dibagikan pada penerima (mustahik). Sehingga sistem keuangan bagi admin ini dapat mengelola pemasukan dan pengeluaran dengan lebih ter-data dan terekap secara digital dalam pengelolaan data keuangannya.

B. Personel yang terlibat

Personel Tim Manajemen Metodologi Penelitian Teknologi Informasi dengan judul proyek “Sistem Keuangan Lazismu Yogyakarta” terdiri dari tujuh orang dengan pembagian *job description* yang sudah ditentukan.

Anggota Manajemen Tugas Proyek Lovlace.

Tabel 1. 1 Data Anggota Lovlace

Nama	NIM	Job Desc
Mohammad Amirul Kurniawan Putranto	2000018425	<i>Project manager & Software Tester</i>
Dinda Putri Dwi Yanti	2000018040	<i>Sekretaris & QAEng</i>
Faza Husain Muhammad Zaen	2000018060	<i>Back-End Developer</i>
Sindi Rella	2000018152	<i>Data Analyst</i>
Zakiya Nurul Baity	2000018414	<i>Front-End Developer</i>
Faqih 'Ainul Yaqin Nursaid	2000018415	<i>Database Engineer</i>
Abdilah Muttaqin	2000018418	<i>UI/UX Designer</i>

Pada tabel 1.1 menjelaskan manajemen tugas dari tim Lovlace mulai dari *Project manager* dan *UI/UX Designer*

C. Proses Mendapatkan Proyek

Proses mendapatkan proyek untuk Manajemen Proyek Teknologi Informasi (MPTI) ini adalah dengan mengikuti program kampus merdeka yaitu Studi Proyek Independen (SPI). ada program ini mitra atau klien dipilih oleh universitas yang telah menjalin kerja sama. pada saat mendaftar program Studi Proyek Independen setiap tim diminta untuk mengisi rencana proyek yang akan dibuat untuk menjadi pertimbangan pemilihan klien oleh universitas. rencana proyek yang akan dibuat tim kami yaitu “Sistem Keuangan”. Setelah mempertimbangkan rencana proyek akhirnya universitas memilih tim kami untuk menjalankan

program Studi Proyek Independen di Lazismu Yogyakarta.

D. Deskripsi Perancangan Proyek

Proyek kami menjelaskan tentang sistem keuangan Lazismu. Proyek ini berbasis *website* supaya dapat memudahkan *user* mengakses sistem yang bersifat *multi device*. Sistem ini diharapkan dapat memberikan kemudahan bagi *user* dalam mengelola keuangan Lazismu dengan lebih terdata dan terekap secara digital dalam pengelolaan keuangan. Untuk *sketch* tampilan awal dapat dilihat pada

Gambar 1.1 *Sketch Awal* dan Gambar 1.2 *Sketch Awal Website Bagian Dashboard*

The sketch shows a login page for 'Lazismu'. At the top center is the title 'Lazismu'. Below it is the text 'Sign In'. There are two input fields: one for 'Email' and one for 'Password'. Below the password field is a 'Sign In' button.

Gambar 1.1 *Sketch Awal*

The sketch shows a dashboard layout. On the left is a sidebar menu with the following items: Lazismu, Menu Utama, Beranda, Akun Debit Kredit, Front Office, Kas Besar, Kas Kecil, Kas Bank, Jurnal Umum, Others, Cabang, User, and Logout. The main content area is titled 'Beranda' and includes a navigation bar with buttons for 'Kas Besar', 'Front Off', 'Kas Kecil', 'Kas Bank', and 'Jurnal'. There is also an 'Administrator' link. Below the navigation bar is a 'Statistik' section with a line graph. The graph has a y-axis ranging from 0 to 1.8 and is labeled 'Data Masuk' and 'Data Keluar'.

Gambar 1. 2 *Sketch Awal Website Bagian Dashboard*

BAB II

Gambaran Instansi

A. Gambaran Umum

1. Sejarah

Lazismu atau Lembaga Amil Zakat, Infak, dan Shadaqah Muhammadiyah merupakan lembaga zakat tingkat nasional yang berkhidmat dalam pemberdayaan masyarakat melalui pendayagunaan dana zakat, infak, wakaf, dan dana kedermawanan lainnya. Lazismu merupakan bagian dari organisasi Islam terbesar di Indonesia, yaitu Muhammadiyah.

Sejarah Lazismu bermula seiring dengan berdirinya organisasi Muhammadiyah pada tahun 1912 oleh K.H. Ahmad Dahlan di Yogyakarta. Sejak awal berdirinya, Muhammadiyah memiliki fokus yang kuat pada kesejahteraan sosial dan ekonomi masyarakat. Salah satu upaya untuk mencapai tujuan tersebut adalah melalui pengelolaan zakat dan sedekah yang dilakukan secara terorganisir.

Pada awalnya, pengelolaan zakat di Muhammadiyah dilakukan secara sederhana, di mana masyarakat yang memiliki kelebihan harta memberikan zakat secara langsung kepada mustahik (penerima zakat) yang mereka ketahui. Namun, dengan berkembangnya Muhammadiyah sebagai organisasi yang lebih besar dan kompleks, diperlukan lembaga khusus untuk mengelola zakat dan sedekah dengan lebih terstruktur.

Pada tahun 1986, Muhammadiyah secara resmi mendirikan Lembaga Amil Zakat Muhammadiyah (Lazismu) yang bertujuan untuk menghimpun, mengelola, dan mendistribusikan dana zakat, infak, dan sedekah dengan lebih baik. Lazismu didirikan sebagai lembaga zakat tingkat nasional yang beroperasi di bawah naungan Muhammadiyah.

Seiring berjalannya waktu, Lazismu terus mengembangkan diri dan mengadopsi teknologi dalam pengelolaan sistem keuangan dan operasionalnya. Lazismu memiliki tugas mengumpulkan dana zakat dan sedekah dari masyarakat yang ingin berkontribusi dan melakukan pendayagunaan dana tersebut untuk program-program pemberdayaan masyarakat, kemanusiaan, pendidikan, kesehatan, dan bidang-bidang lain yang menjadi fokus Muhammadiyah.

Lazismu juga menjalankan kegiatan sosial, seperti program bantuan bagi mustahik, pengelolaan wakaf, pengembangan koperasi, pendidikan keuangan, dan pemberdayaan ekonomi. Lazismu bekerja sama dengan berbagai pihak, termasuk individu, lembaga, perusahaan, dan instansi lainnya, untuk meningkatkan kesejahteraan masyarakat dan melaksanakan prinsip-prinsip Islam dalam pengelolaan keuangan.

Dengan peran dan kontribusinya dalam pengelolaan dana zakat dan sedekah, Lazismu berperan penting dalam menjalankan misi Muhammadiyah untuk meningkatkan kesejahteraan sosial dan ekonomi masyarakat, serta mendorong pemberdayaan umat.

2. Visi dan Misi

Visi

Menjadi Lembaga Amil Zakat Terpercaya

Misi

- 1) Optimalisasi kualitas pengolahan ZIS yang Amanah, profesional dan transparan.
- 2) Optimalisasi pendayagunaan ZIS yang kreatif, inovatif dan produktif.
- 3) Optimalisasi pelayanan donator.

3. Alamat dan Kontak Klien

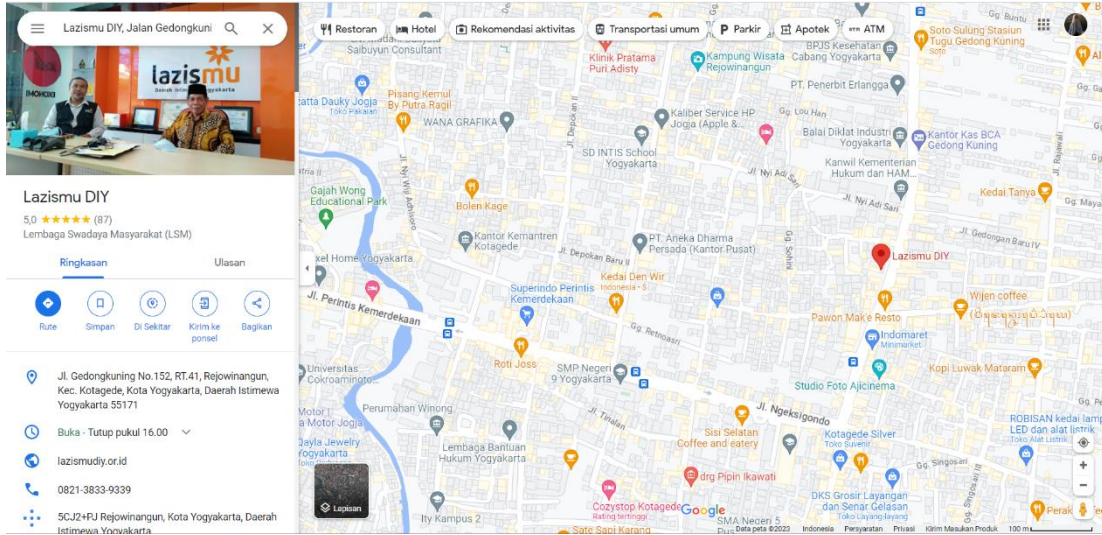
Nama : Lazismu Daerah Yogyakarta

Alamat : Jl. Gedongkuning No.152, RT.41, Rejowinangun, Kec.

Kotagede, Kota Yogyakarta, Daerah Istimewa Yogyakarta 55171, Indonesia.

Kontak : 0821-3833-9339

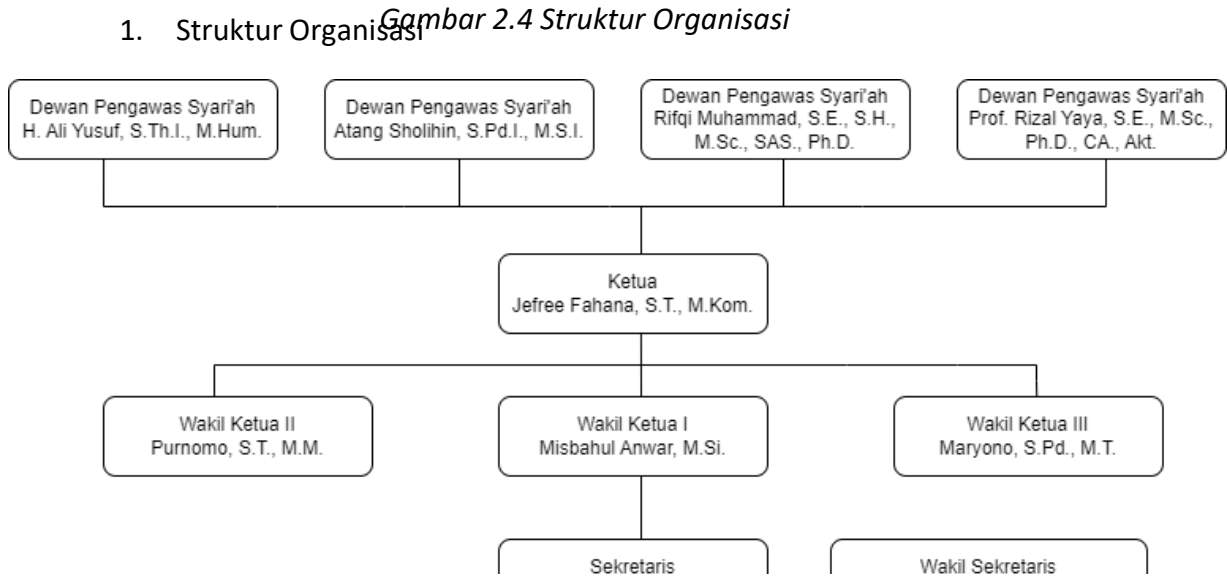
Peta Lokasi : <https://goo.gl/maps/MDi87Ecr7Ycc5ojm8>



Gambar 2.1 Lokasi Lazismu DIY

B. Struktur Organisasi Sumber Daya Manusia dan Sumber Daya Fisik

1. Struktur Organisasi



Gambar 2.10 Struktur Organisasi

Gambar 2.11 Proses Bisnis Lazismu

Gambar 2.12 Struktur Organisasi

Gambar 2.13 Proses Bisnis Lazismu

Gambar 2.14 Struktur Organisasi

Gambar 2.15 Proses Bisnis Lazismu

Gambar 2.16 Struktur Organisasi

2. Sumber Daya Manusia

Sumber daya manusia Lazismu terdiri dari jumlah pimpinan Lembaga dan jumlah karyawan Lazismu:

Tabel 2.1 Sumber Daya Manusia Lazismu

Profesi	Jumlah
Dewan Pengawas Syari'ah	4
Ketua	1
Wakil Ketua	3
Sekretaris	2
Anggota	15

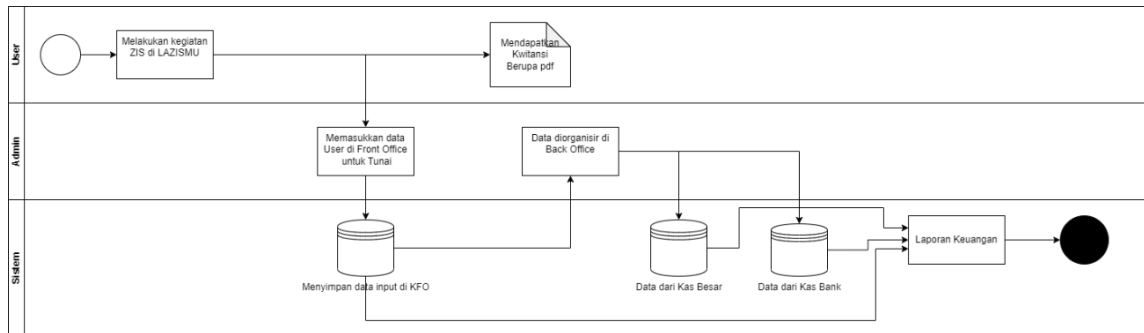
3. Sumber Daya Fisik

Sumber daya fisik adalah sumber daya yang memiliki wujud fisik yang digunakan oleh lembaga untuk mengelola sistem untuk menginputkan, menyimpan dan memproses data. Sumber daya fisik yang dimiliki Lazismu dapat dilihat pada tabel berikut:

Tabel 2.2 Sumber Daya Fisik Lazismu

No	Nama
1	Laptop
2	Mouse
3	Meja
4	Kursi

C. Gambaran Sistem atau Proses Bisnis pada Mitra/Klien saat ini



Gambar 2.17 Proses Bisnis Lazismu

BAB III

Rencana MPTI

A. Rencana Pembagian Tugas

Rencana pembagian tugas Lovlace terkait pembuatan Sistem Keuangan di Lazismu adalah sebagai berikut :

1. *Project Manager* dan *Software Tester* (Mohammad Amirul Kurniawan Putranto)
 - a. Membuat tim proyek
 - b. Memimpin penentuan pembagian *job desc* anggota tim
 - c. Mengkoordinasi dan mengatur waktu bimbingan
 - d. Merencanakan, melaksanakan, mengawasi, dan mengevaluasi pelaksanaan proyek
 - e. Mengkoordinasi dan memimpin setiap pertemuan atau rapat
 - f. Melakukan pengecekan atau testing terhadap bug di dalam aplikasi
2. Sekretaris / *QAEng* (Dinda Putri Dwi Yanti)
 - a. Membuat notulensi rapat tim Manajemen Proyek Teknologi Informasi
 - b. Mendokumentasikan rapat serta dokumen rapat tim
 - c. Membuat *logbook* tim
 - d. Mengkomunikasikan jadwal rapat dengan mitra/klien
 - e. Membuat laporan pengerjaan proyek tim
 - f. Merencanakan dan merancang strategi pengujian untuk memastikan bahwa seluruh fitur dan fungsi sistem diuji secara menyeluruh.
 - g. Melakukan pengujian fungsional dan non-fungsional untuk menguji sistem.
 - h. Melakukan pemantauan secara aktif terhadap sistem yang diuji untuk mengidentifikasi *bug*, kesalahan, atau masalah kualitas lainnya
 - i. Berperan dalam meningkatkan proses pengembangan dan pengujian. Dan menganalisis kelemahan dalam proses saat ini, memberikan saran

perbaikan, dan mengimplementasikan praktik terbaik untuk meningkatkan efisiensi dan kualitas pengujian.

3. *Back-End Developer* (Faza Husain Muhammad Zaen)
 - a. Mengimplementasikan program dan *database* hasil dari analisis sistem yang telah dibuat
 - b. Membuat sistem *Login* dan *dashboard* pada sistem
 - c. Membuat fitur *Login* dan CRUD
 - d. Memperbaiki program yang telah dievaluasi oleh klien
4. *Data Analyst* (Sindi Rella)
 - a. Melakukan analisis terhadap sistem yang akan dibuat
 - b. Membuat proses bisnis dan alur sistem
 - c. Membuat *Use case diagram*
 - d. Membuat *Activity Diagram*
 - e. Membuat *Entity Relationship Diagram*
 - f. Menganalisis kebutuhan bisnis, kebutuhan *user*, kebutuhan fungsional, non-fungsional, dan kebutuhan sistem
5. *Front-End Developer* (Zakiya Nurul Baity)
 - a. Mengimplementasikan rancangan tampilan yang telah dibuat
 - b. Menentukan *layout* atau tata letak dari komponen sistem yang akan dibuat
 - c. Memaksimalkan *User Experience (UX)*
 - d. Memastikan keselarasan antara fungsionalitas dengan estetika
 - e. Memastikan kompatibilitas *website*
 - f. Ikut serta membantu *Back-End Developer* dalam pembuatan sistem
6. *Database Engineer* (Faqih 'Ainul Yaqin Nursaid)
 - a. Bertanggung jawab untuk merancang, mengembangkan, mengelola, dan memelihara sistem *database*
 - b. Mengembangkan dan membangun *database* berdasarkan desain yang telah dibuat
 - c. Menganalisis *query* yang lambat, mengoptimalkan indeks, dan melakukan *tuning* parameter untuk memastikan *database* berjalan

dengan efisien dan responsif.

- d. Merancang skema *database* yang mencakup struktur tabel, relasi antara tabel, dan definisi atribut.

7. *UI/UX Designer* (Abdilah Muttaqin)

- a. Menyusun tata letak halaman yang ada di *website*.
- b. Mendesain interaksi setiap elemen yang ada di halaman sehingga memudahkan admin.
- c. Membuat *prototype user interface* sekaligus mengimplementasikan secara langsung.
- d. Menampilkan *interface* dengan keseragaman yang baik dari segi warna, jenis, huruf, gambar, *icon*, dan lainnya.
- e. Merencanakan anggaran proyek.
- f. Bertanggung jawab untuk mengelola keuangan tim.
- g. Menyusun laporan keuangan berkala yang mencerminkan status keuangan.
- h. Memantau pengeluaran proyek secara rutin untuk memastikan bahwa pengeluaran tetap berada dalam batas anggaran yang telah ditetapkan.

B. Perancangan Sistem

1. Analisis Kebutuhan Sistem

Ada dua jenis kebutuhan sistem, yaitu kebutuhan fungsional dan non-fungsional. Analisis kebutuhan fungsional adalah pernyataan layanan yang harus disediakan untuk sistem untuk melakukan input tertentu dalam keadaan tertentu. Analisis kebutuhan non-fungsional mencakup batasan pada layanan atau fungsi yang disediakan oleh sistem (Kosasi & Kuway, 2012). Adapun analisis kebutuhan fungsional dan non-fungsional dijabarkan sebagai berikut:

- a. Kebutuhan Fungsional
 - 1) Sistem menampilkan halaman dalam bentuk *website*
 - 2) Sistem memiliki fitur *Login* untuk admin
 - 3) Sistem memiliki fitur beranda

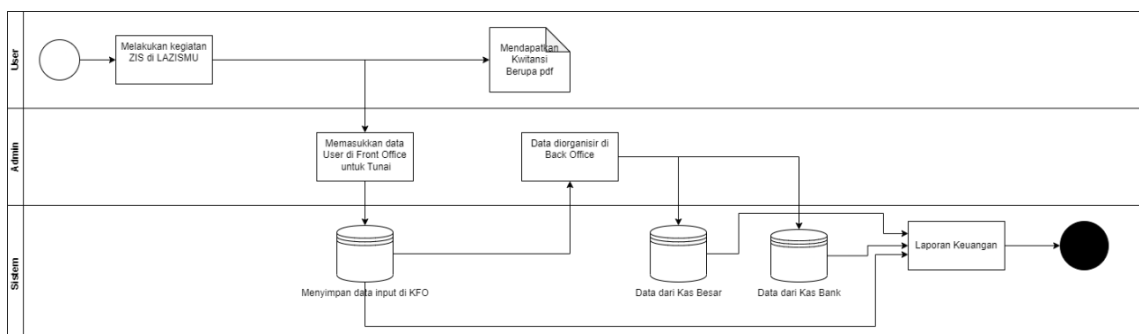
- 4) Sistem dapat melihat data dan mengeluarkan output berupa *file* dengan format *.xlsx*
- 5) Sistem dapat mengelola data muzakki
- 6) Sistem memiliki fitur pencarian data muzakki
- 7) Sistem memiliki fitur manajemen *user*
- 8) Sistem memiliki fitur unduh data keuangan
- 9) Sistem memiliki fitur CRUD

b. Kebutuhan Non-Fungsional

- 1) Sistem dapat digunakan sepanjang waktu
- 2) Sistem menggunakan Bahasa yang umum dan mudah dimengerti
- 3) Tampilan desain *interface* variatif dan mudah digunakan sesuai dengan *experience user*
- 4) Sistem harus dapat memastikan bahwa data yang digunakan dalam sistem harus terlindungi dari akses yang tidak berwenang atau data tidak dapat diakses oleh pengguna yang tidak berhak
- 5) Sistem kompatibel terhadap komputer dan *smartphone*

2. Proses Bisnis Diagram

Didalam proses bisnis diagram Sistem Keuangan Lazizmu terdapat tiga yaitu admin, *user* dan sistem yang nanti digunakan pada pegawai Lazizmu.



Gambar 3.1 Proses Bisnis Bagian Admin

Pada bagian admin memiliki fitur yaitu antara lain :

➤ Beranda

statistik perkembangan pemasukan dan pengeluaran dalam bentuk linechart. Pada sidebar terdapat menu bernada, COA, Frontoffice, Kas Besar, Kas Kecil, Kas bank, Muzakki, dan Other yang berisi menu cabang, *user*, kemudian tombol *Logout*. Setiap menu yang diklik akan berubah warna menjadi oranye seperti pada menu beranda di atas.

➤ Manajemen *user*

Pada fitur ini admin dapat melakukan pencarian dari data *user* serta melakukan CRUD selanjutnya admin dapat melakukan *import* dan ekspor berbentuk *xlsx*. Selain itu pada COA terdapat *summary* mengenai jumlah akun debit dan akun debit.

3. Use Case Diagram

Use Case Diagram di atas menggambarkan fungsionalitas yang diharapkan dari sebuah sistem *website* keuangan Lazismu. *Use Case Diagram* ini digunakan untuk merepresentasikan interaksi antara aktor admin dengan sistem yang ada.

Aktor yang terlibat pada sistem ini yaitu admin. Admin adalah orang yang dapat mengakses sistem *website* keuangan Lazismu, mulai dari *Sign In* lalu mengelola semua data yang diperlukan pada menu-menu yang tersedia, mengecek data, melihat *list* cabang kantor dan *user* yang ada, serta admin dapat melakukan *Logout*.

Penjelasan terkait Use Case yang dibuat:

- Input Data Keuangan

Menginput data keuangan seperti data uang masuk dari seorang muzakki yang ingin berzakat. Dengan aksi dari admin menambahkan tanggal

muzakki mendaftar, no. bukti admin yang menerima pendaftaran tersebut, uraian transaksi, nama muzakki, nama admin penerima pendaftaran, memilih akun antara debit atau kredit, jumlah zakat, tempat/cabang melakukan pembayaran, dan no. ref.

- Edit Data Keuangan

Admin dapat mengedit data keuangan yang sudah dibuat, seperti melakukan *update* data dan lain-lainnya.

- Lihat Data Keuangan

Admin dapat melihat *list* data-data keuangan yang sudah ada pada sistem atau melihat hasil edit data keuangan yang dilakukan.

- Hapus Data Keuangan

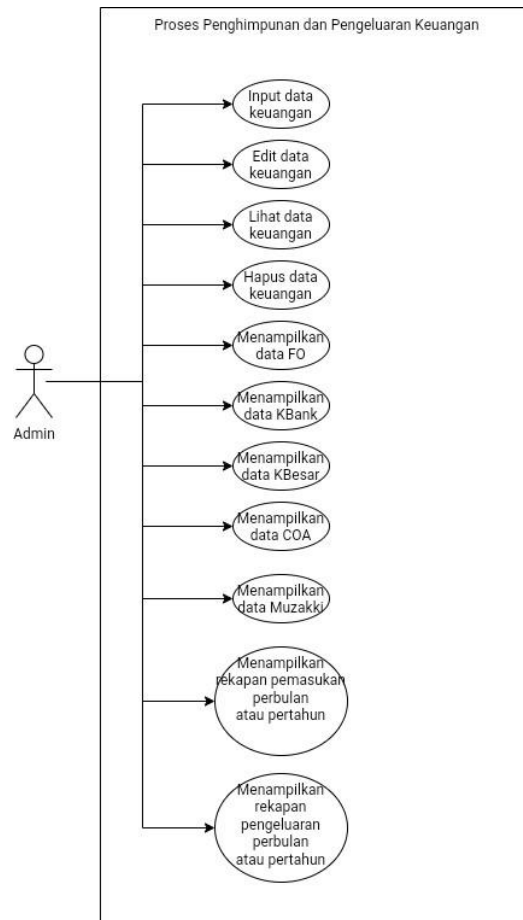
Admin dapat menghapus data keuangan yang sudah dibuat, seperti melakukan *delete* data yang sudah tidak diperlukan dan lain-lainnya.

- Menampilkan data *Front Office* (FO), Kas Bank, Kas Besar, CoA, dan Muzakki

Admin dapat menampilkan data-data yang tersedia pada menu *Front Office*, Kas Bank, dan Kas Besar.

- Menampilkan rekapan pemasukan

Admin dapat menampilkan rekapan pemasukan uang dari muzakki untuk zakat dengan rentang waktu yang dipilih yaitu secara per bulan.



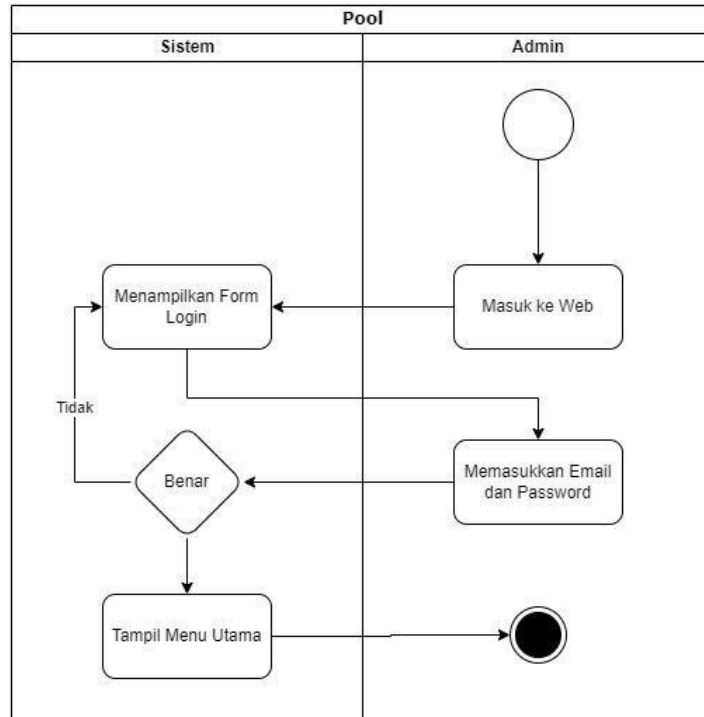
Gambar 3.2 Use Case Diagram Sistem Keuangan Lazimu

4. Activity Diagram

Activity diagram merupakan pemodelan untuk melihat bagaimana cara suatu calon aplikasi bekerja menjalankan fungsi-fungsinya. Dalam *Activity* diagram sistem informasi kepegawaian ini terdapat empat *Activity* yaitu admin, kas besar, kas kecil, kas bank, dan CoA.

a. Admin

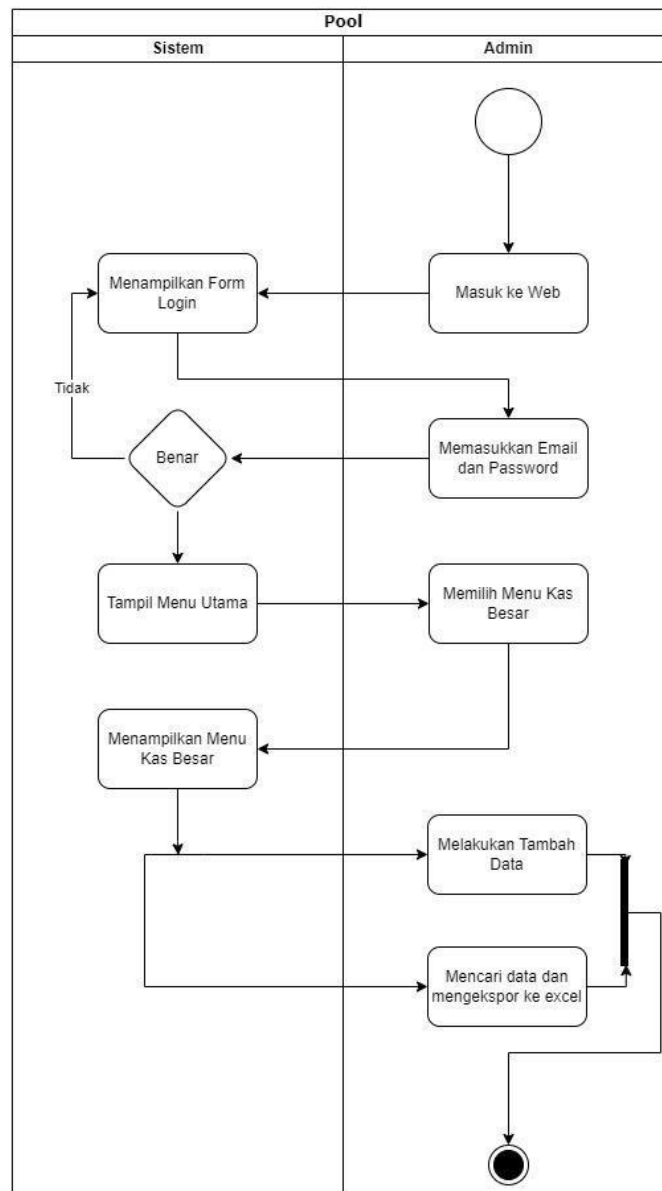
Gambar 3.3 merupakan *Activity* diagram pada saat admin masuk ke *website* kemudian akan tampil menu *Login* di mana admin menginputkan *email* dan *password* sehingga *website* akan menampilkan halaman menu utama.



Gambar 3.3 Activity Diagram Admin

b. Kas Besar

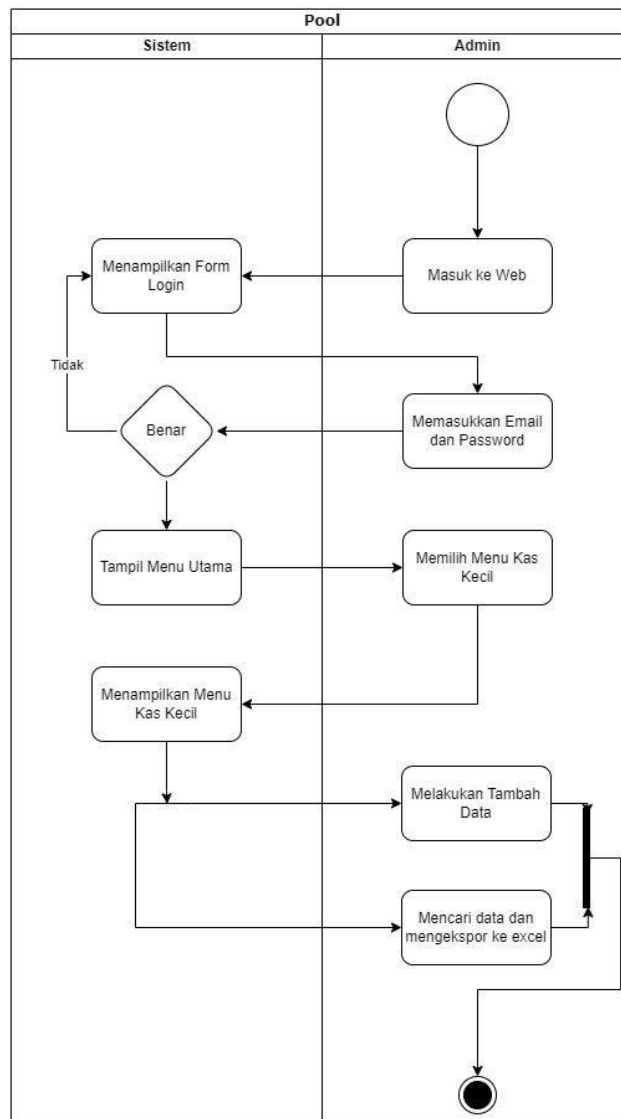
Gambar 3.4 merupakan *Activity* diagram pada saat admin akan menuju menu kas besar sehingga sistem akan menampilkan menu tampilan kas besar yang berisikan data kasir dari kas besar.



Gambar 3.4 Activity Diagram Kas Besar

c. Kas Kecil

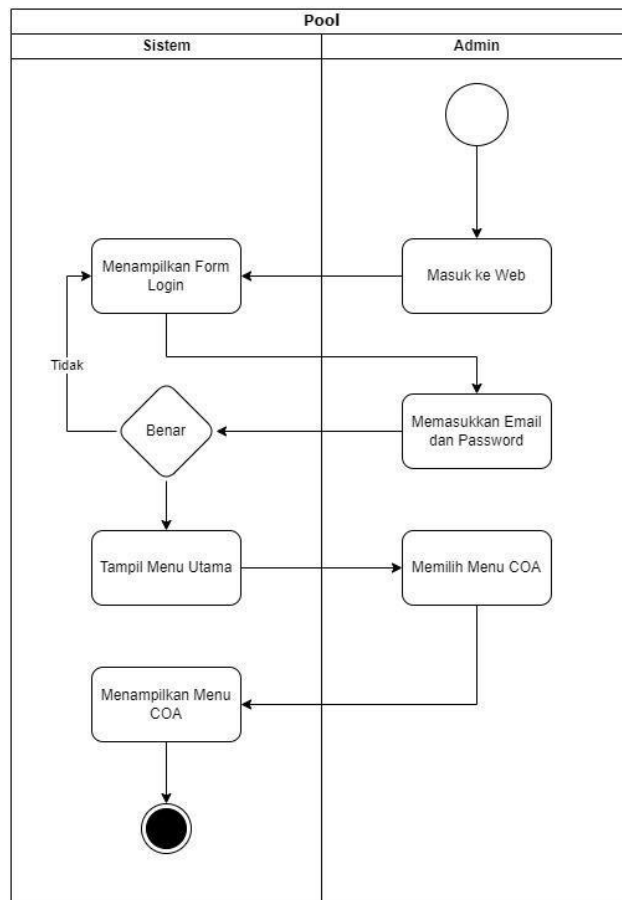
Gambar 3.5 merupakan *Activity* diagram pada saat admin akan menuju menu kas kecil sehingga sistem akan menampilkan menu tampilan kas kecil yang berisikan data kasir dari kas kecil.



Gambar 3.5 Activity Diagram Kas Kecil

d. COA

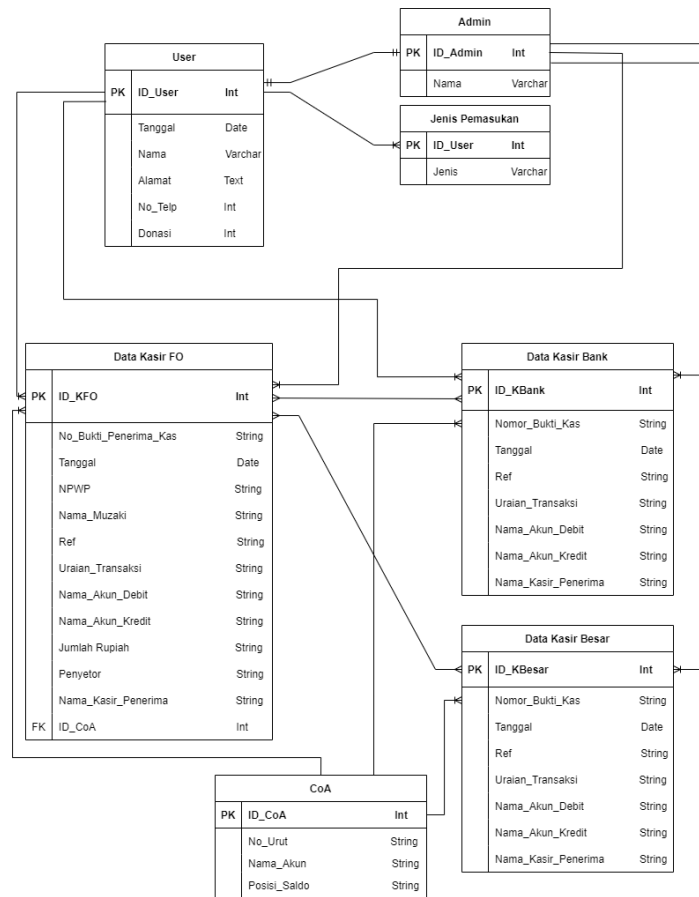
Pada *Activity* diagram tersebut, admin akan menuju menu kas besar sehingga sistem akan menampilkan menu tampilan CoA yang berisikan data Kode Akun (*Chart of Account*) dalam bentuk tabel.



Gambar 3.6 Activity Diagram COA

5. Rancangan Struktur *Database*

Basis data untuk sistem keuangan Lazismu adalah komponen krusial dalam mengelola semua transaksi keuangan yang terkait dengan penerimaan, pengeluaran, dan pelaporan dana zakat, infak, dan shadaqah. Rancangan struktur *database* untuk sistem ini harus dirancang dengan cermat agar mendukung efisiensi, akurasi, dan keamanan dalam mengelola data keuangan yang sensitif. Keamanan data harus dijaga melalui enkripsi kata sandi pengguna dan penerapan aturan akses yang ketat. Selain itu, penggunaan indeks pada kolom-kolom yang sering digunakan untuk pencarian dapat meningkatkan kinerja sistem.



Gambar 3.7 Struktur Database Sistem Keuangan Lazismu

6. Rencana Sistem

❖ Tampilan *Login/Sign In*

Menu *Login* atau *Sign In* dibuat statis di mana hanya admin atau pegawai yang bertugas yang dapat mengakses, *password* bisa di sembunyikan dan ditampilkan dengan mengklik *icon* mata, terdapat logo Lazismu DIY.



Sign In

Kelola keuangan dengan sistem yang baru, semoga lebih membantu

Email Address

admin@lazismu.com

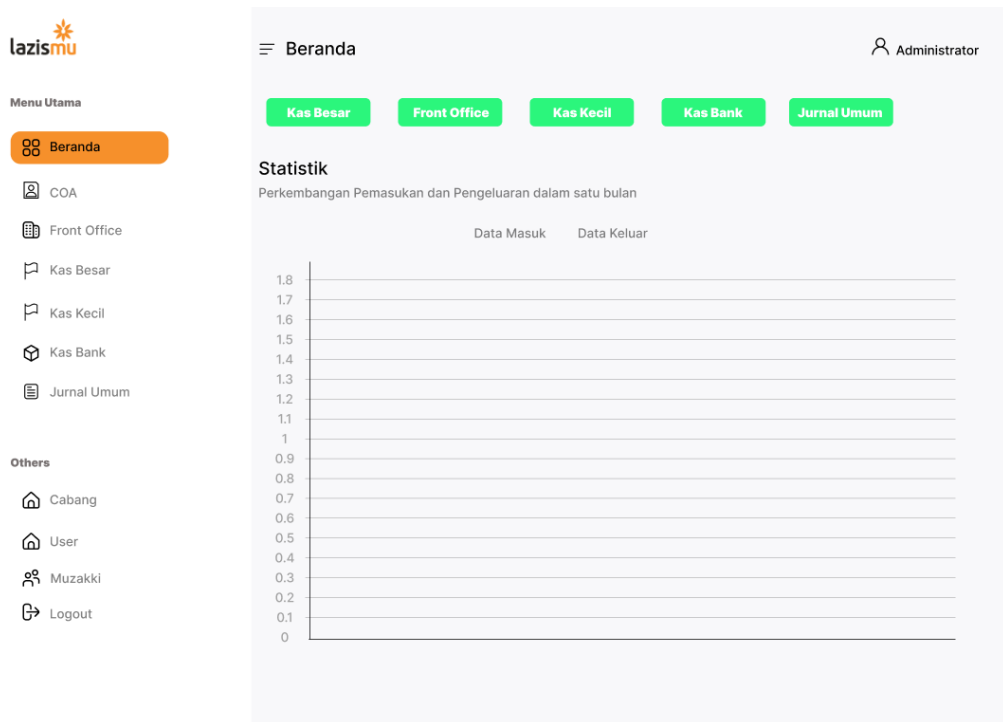
Password

Sign In

Gambar 3.8 Tampilan Login/Sign In

❖ Beranda

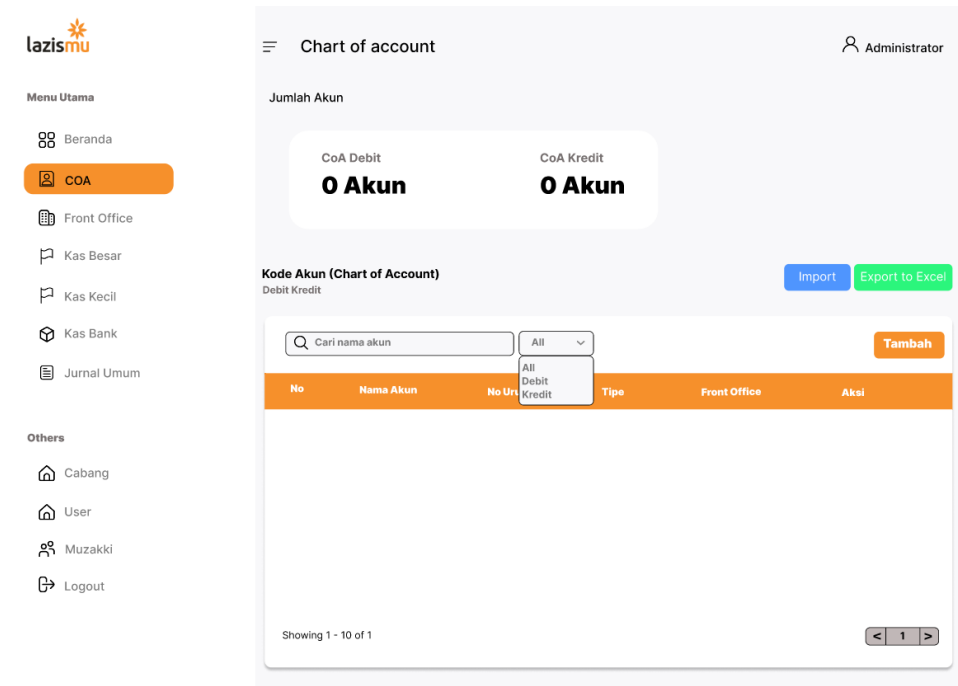
Terdapat statistik perkembangan pemasukan dan pengeluaran dalam bentuk *line chart*. Pada *side bar* terdapat menu Beranda, CoA, *Front Office*, Kas Besar, Kas Kecil, Kas bank, Muzakki, dan *Other* yang berisi menu Cabang, *User*, kemudian tombol *Logout*. Setiap menu yang diklik akan berubah warna menjadi oranye seperti pada menu beranda di atas.



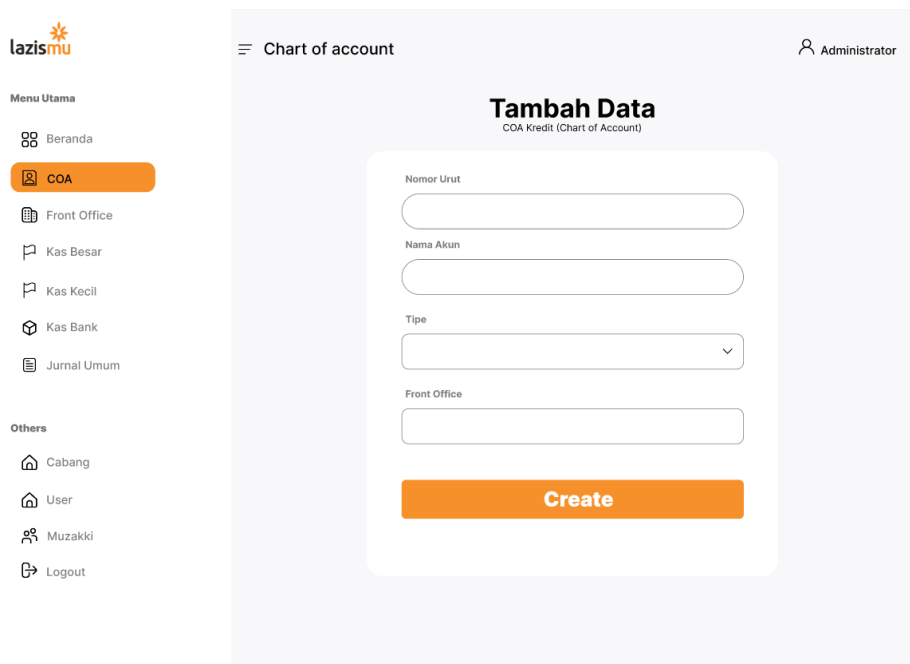
Gambar 3.9 Tampilan Beranda

❖ CoA

Pada menu CoA atau *Chart of Account* terdapat jumlah akun CoA debit dan kredit, kemudian terdapat juga menu *import* dan *export* dalam bentuk *file excel*, serta tabel berisikan nama, kode, dan tipe yang didapat dari pengisian *form* CoA yang dapat di akses dengan mengklik menu tambah seperti pada gambar di bawah ini.



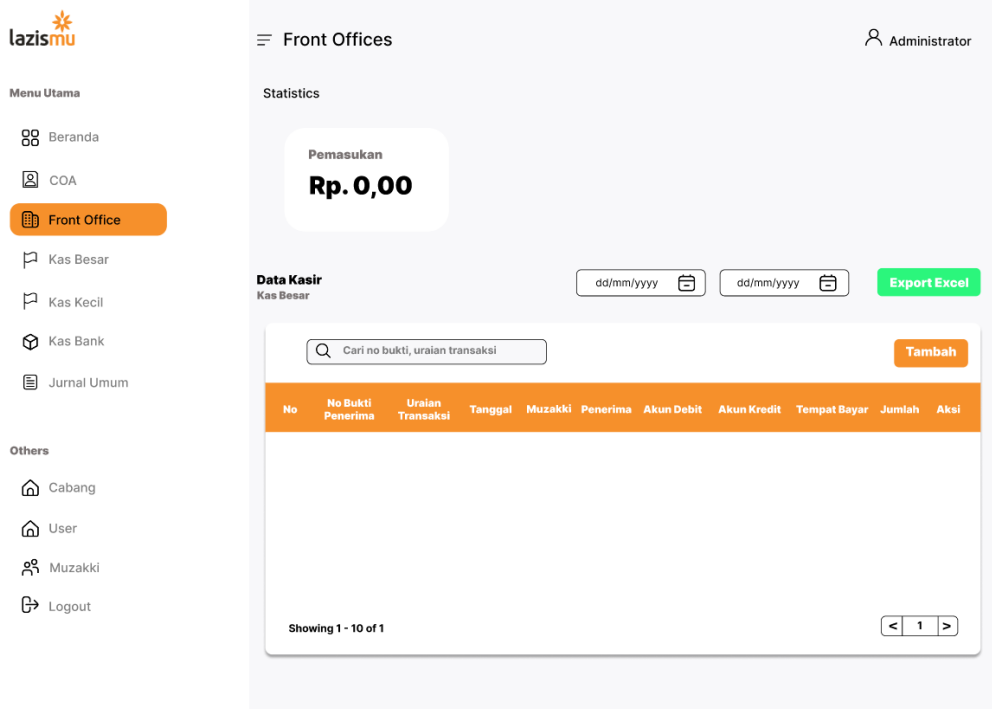
Gambar 3.10 Tampilan CoA



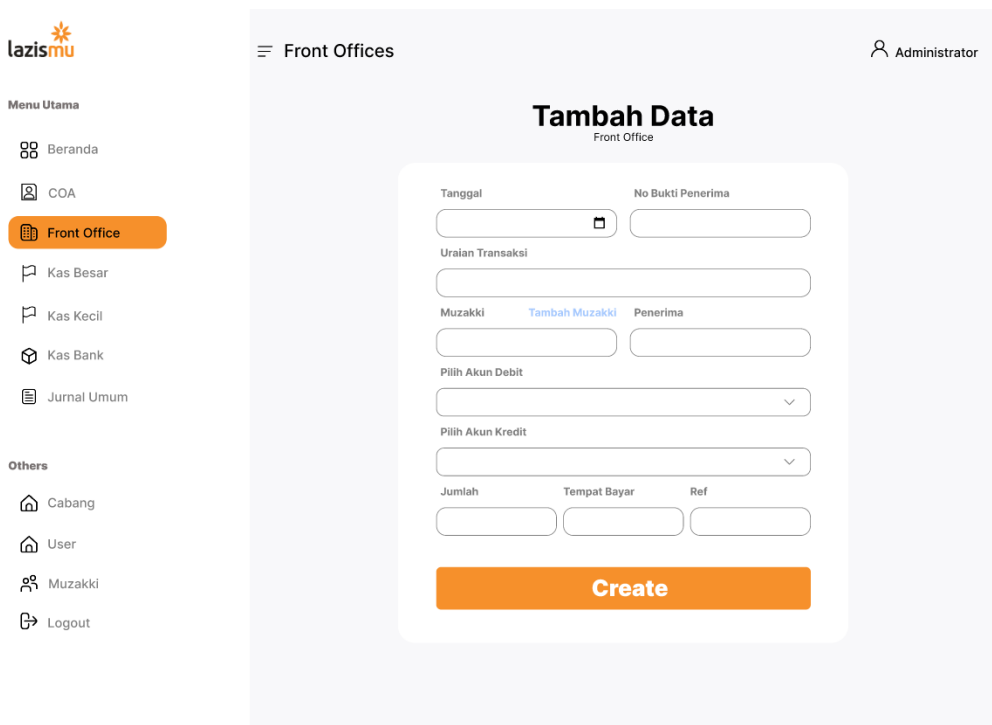
Gambar 3.11 Tampilan Tambah Data CoA

❖ *Front Office*

Pada menu *Front Office* terdapat data jumlah pemasukan total, menu *export*, menu tanggal kemudian menu cari no. bukti transaksi atau uraian transaksi, dan terdapat tabel yang datanya didapat dari menginputkan data pada menu tambah seperti gambar di bawah ini.



Gambar 3.12 Tampilan Front Office



Gambar 3.13 Tampilan Tambah Data Pada Front Office

❖ Kas Besar

Tidak beda jauh dari sebelumnya pada Kas Besar terdapat total pemasukan dan pengeluaran kemudian yang membedakan dari menu sebelumnya hanya pada tabel dan input datanya yang terdapat pada menu tambah di mana terdapat dua jenis input data yaitu data pemasukan dan pengeluaran yang di mana untuk mengubah *form*-nya dengan mengklik tombol pengeluaran atau pemasukan pada menu tersebut seperti pada gambar di bawah ini.

The screenshot displays the 'Kas Besar' (General Ledger) interface. On the left is a sidebar menu with 'Kas Besar' selected. The main area shows a header 'Kas Besar' and a user profile 'Administrator'. Below is a 'Statistics' section with two cards: 'Pemasukan Rp. 0,00' and 'Pengeluaran Rp. 943.750,00'. Underneath is the 'Data Kasir' section, which includes search filters for 'dd/mm/yyyy' and an 'Export Excel' button. A search bar contains the text 'Cari uraian transaksi, no bukti'. Below the search bar is a table with the following columns: No, No bukti kas, Tanggal, Ref, Uraian transaksi, Nama akun debit, Nama akun kredit, Nama kasir, Jumlah, and Edit. The table currently shows one row. At the bottom, it indicates 'Showing 1 - 10 of 1' and has pagination controls.

Gambar 3.14 Tampilan Kas Besar

❖ Pemasukan

The screenshot shows the 'Tambah Data' (Add Data) form for 'Pemasukan' (Income) in the 'Kas Besar' (General Ledger) section. The form is titled 'Tambah Data' with a subtitle 'Kas Besar'. It features a navigation menu on the left with 'Kas Besar' highlighted. The form fields include: 'Tanggal' (Date) with a calendar icon, 'No Bukti Penerima' (Receipt Number) with a text input, 'Uraian Transaksi' (Transaction Description) with a text input, 'Penerima' (Receiver) with a text input, 'Pilih Akun Debit' (Select Debit Account) with a dropdown menu, 'Pilih Akun Kredit' (Select Credit Account) with a dropdown menu, 'Jumlah' (Amount) with a text input, and 'Ref' (Reference) with a dropdown menu showing 'JKBB'. A 'Create' button is at the bottom.

Gambar 3.15 Tampilan Pemasukan Pada Kas Besar

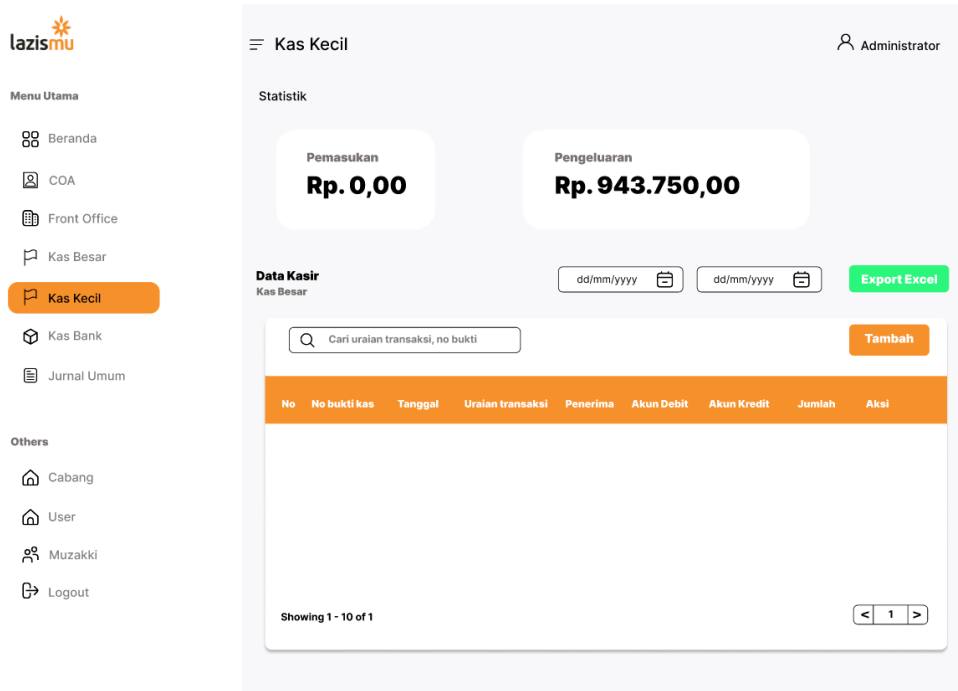
❖ Pengeluaran

The screenshot shows the 'Tambah Data' (Add Data) form for 'Pengeluaran' (Expense) in the 'Kas Besar' (General Ledger) section. The form is titled 'Tambah Data' with a subtitle 'Kas Besar'. It features a navigation menu on the left with 'Kas Besar' highlighted. The form fields include: 'Tanggal' (Date) with a calendar icon, 'No Bukti Penerima' (Receipt Number) with a text input, 'Uraian Transaksi' (Transaction Description) with a text input, 'Penerima' (Receiver) with a text input, 'Pilih Akun Debit' (Select Debit Account) with a dropdown menu, 'Pilih Akun Kredit' (Select Credit Account) with a dropdown menu, 'Jumlah' (Amount) with a text input, and 'Ref' (Reference) with a dropdown menu showing 'JKBB'. A 'Create' button is at the bottom.

Gambar 3.16 Tampilan Pengeluaran Pada Kas Besar

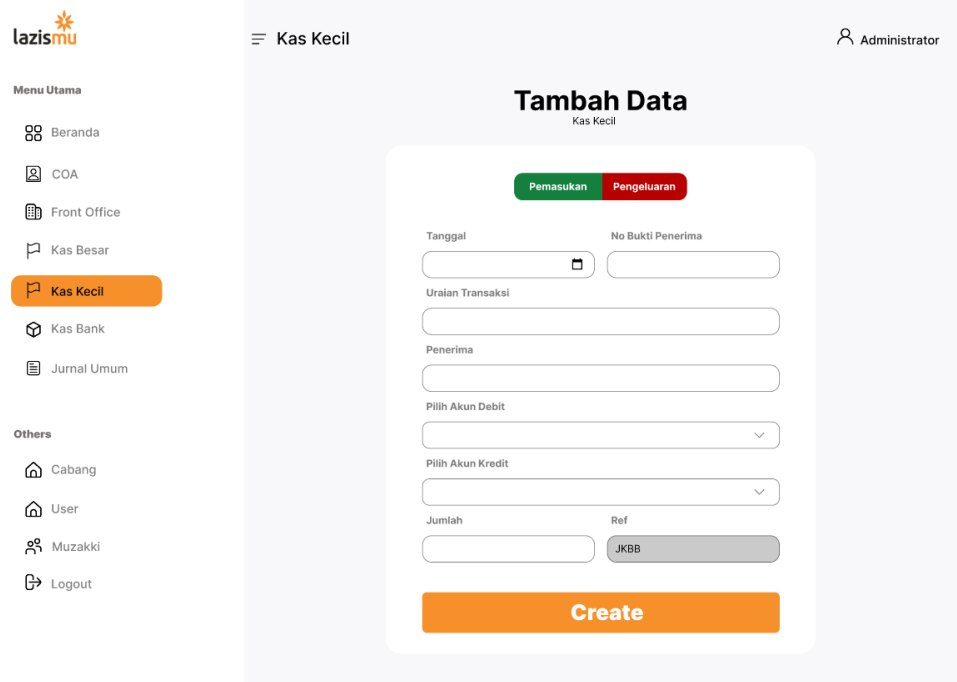
❖ Kas Kecil

Pada menu Kas Kecil sama persis dengan menu Kas Besar yang membedakan hanya pada tabelnya, untuk *form* pada menu tambah juga sama terdapat pemasukan dan pengeluaran.



Gambar 3.17 Tampilan Kas Kecil

❖ Pemasukkan



Gambar 3.18 Tampilan Pemasukan Pada Kas Kecil

❖ Pengeluaran

The screenshot shows the 'Tambah Data' form for 'Kas Kecil'. The form is titled 'Tambah Data' with a subtitle 'Kas Kecil'. It features a navigation menu on the left with 'Kas Kecil' highlighted. The form itself has a header with 'Pemasukan' and 'Pengeluaran' tabs. The main form area contains the following fields: 'Tanggal' (date), 'No Bukti Penerima' (receipt number), 'Uraian Transaksi' (transaction description), 'Penerima' (recipient), 'Pilih Akun Debit' (debit account dropdown), 'Pilih Akun Kredit' (credit account dropdown), 'Jumlah' (amount), and 'Ref' (reference, currently 'JKBB'). A large orange 'Create' button is at the bottom.

Gambar 3.19 Tampilan Pengeluaran Pada Kas Kecil

❖ Kas Bank

Untuk Kas bank tampilannya masih sama seperti menu sebelumnya yang membedakan hanya isi dari tabelnya. Untuk *formnya* juga sama

The screenshot shows the 'Kas Bank' dashboard. It features a navigation menu on the left with 'Kas Bank' highlighted. The main content area is titled 'Kas Bank' and includes a 'Statistics' section with three cards: 'Pemasukan Rp. 0,00', 'Pengeluaran Rp. 0,00', and 'Jumlah Saldo Rp. 0,00'. Below the statistics is a 'Data Kasir' section with a search bar, date filters, and an 'Export to Excel' button. A table with columns for 'No', 'No bukti kas', 'Tanggal', 'Uraian transaksi', 'Penerima', 'Akun Debit', 'Akun Kredit', 'Jumlah', and 'Aksi' is shown. The table currently displays one row. At the bottom, it says 'Showing 1 - 10 of 1'.

Gambar 3.20 Tampilan Kas Bank

❖ Pemasukan

The screenshot shows the 'Tambah Data' (Add Data) form for 'Kas Bank' in the 'lazismu' system. The form is titled 'Tambah Data' with a subtitle 'Kas Bank'. It features a navigation menu on the left with 'Kas Bank' highlighted. The form itself has two tabs: 'Pemasukan' (Income) and 'Pengeluaran' (Expense), with 'Pemasukan' selected. The form fields include: 'Tanggal' (Date) with a calendar icon, 'No Bukti Penerima' (Receipt Number), 'Uraian Transaksi' (Transaction Description), 'Penerima' (Receiver), 'Pilih Akun Debit' (Select Debit Account), 'Pilih Akun Kredit' (Select Credit Account), 'Jumlah' (Amount), and 'Ref' (Reference) with a dropdown menu showing 'JKBB'. A large orange 'Create' button is at the bottom.

Gambar 3.21 Tampilan Pemasukan Pada Kas Bank

❖ Pengeluaran

The screenshot shows the 'Tambah Data' (Add Data) form for 'Kas Bank' in the 'lazismu' system. The form is titled 'Tambah Data' with a subtitle 'Kas Bank'. It features a navigation menu on the left with 'Kas Bank' highlighted. The form itself has two tabs: 'Pemasukan' (Income) and 'Pengeluaran' (Expense), with 'Pengeluaran' selected. The form fields include: 'Tanggal' (Date) with a calendar icon, 'No Bukti Penerima' (Receipt Number), 'Uraian Transaksi' (Transaction Description), 'Penerima' (Receiver), 'Pilih Akun Debit' (Select Debit Account), 'Pilih Akun Kredit' (Select Credit Account), 'Jumlah' (Amount), and 'Ref' (Reference) with a dropdown menu showing 'JKBB'. A large orange 'Create' button is at the bottom.

Gambar 3.22 Tampilan Pengeluaran Pada Kas Bank

❖ Jurnal Umum

lazismu

Menu Utama

- Beranda
- COA
- Front Office
- Kas Besar
- Kas Kecil
- Kas Bank
- Jurnal Umum**

Others

- Cabang
- User
- Muzakki
- Logout

Jurnal umum Administrator

Statistics

Saldo
Rp. 0,00

Data Kasir
Jurnal umum

dd/mm/yyyy dd/mm/yyyy [Export to Excel](#)

Cari uraian transaksi, no bukti [Tambah](#)

No	No bukti penerima	Uraian transaksi	Ref	Tanggal	Akun Debit	Akun Kredit	Jumlah	Aksi
Showing 1 - 10 of 1								

Gambar 3.23 Tampilan Jurnal Umum

❖ Tambah Data

lazismu

Menu Utama

- Beranda
- COA
- Front Office
- Kas Besar
- Kas Kecil
- Kas Bank
- Jurnal Umum**

Others

- Cabang
- User
- Muzakki
- Logout

Jurnal umum Administrator

Tambah Data
Jurnal Umum

Tanggal No Bukti

Uraian Transaksi

Pilih Akun Debit

Pilih Akun Kredit

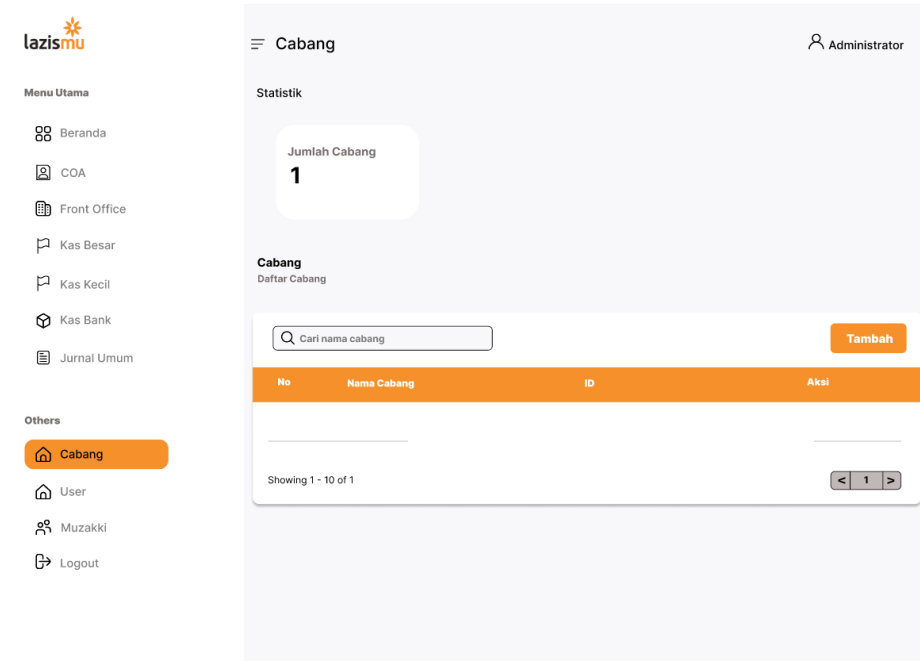
Jumlah Ref

[Create](#)

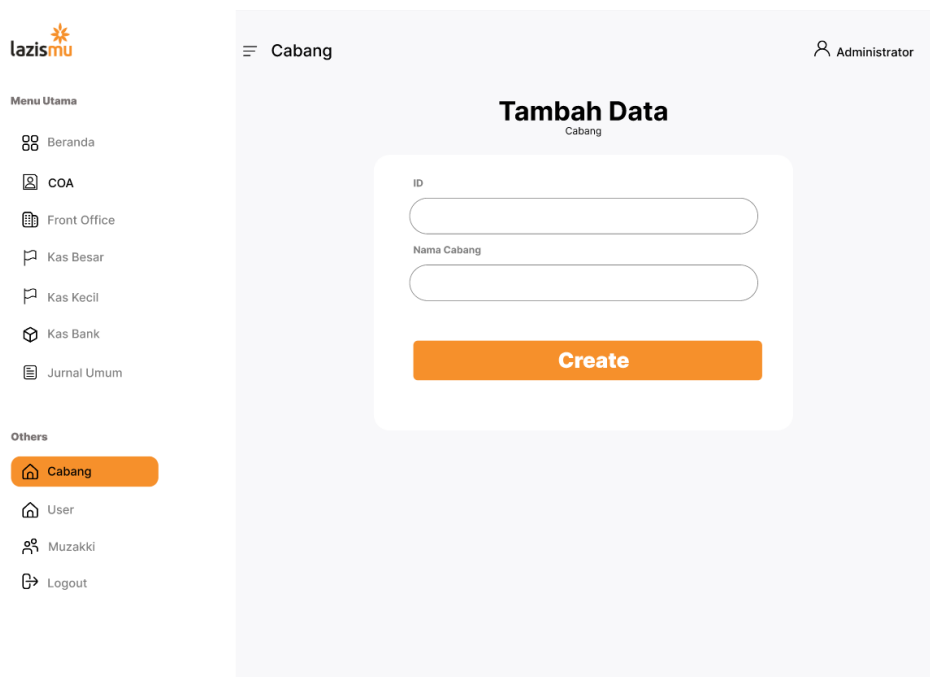
Gambar 3.24 Tampilan Tambah Data Jurnal Umum

❖ Cabang

Yang membedakan pada menu cabang yaitu informasi jumlah cabang dan informasi yang ditampilkan pada tabel, serta *form* yang diisi pada menu tambah.



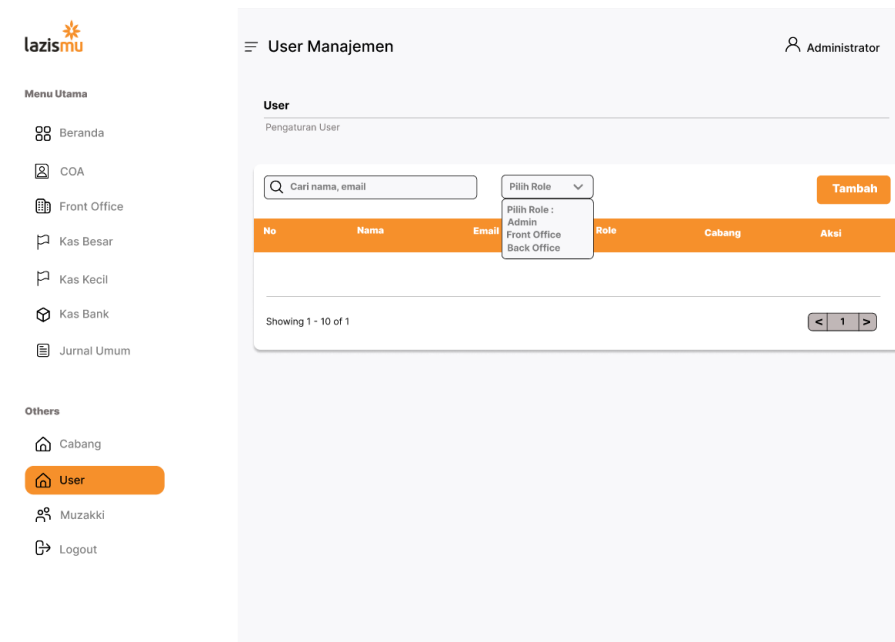
Gambar 3.25 Tampilan Cabang



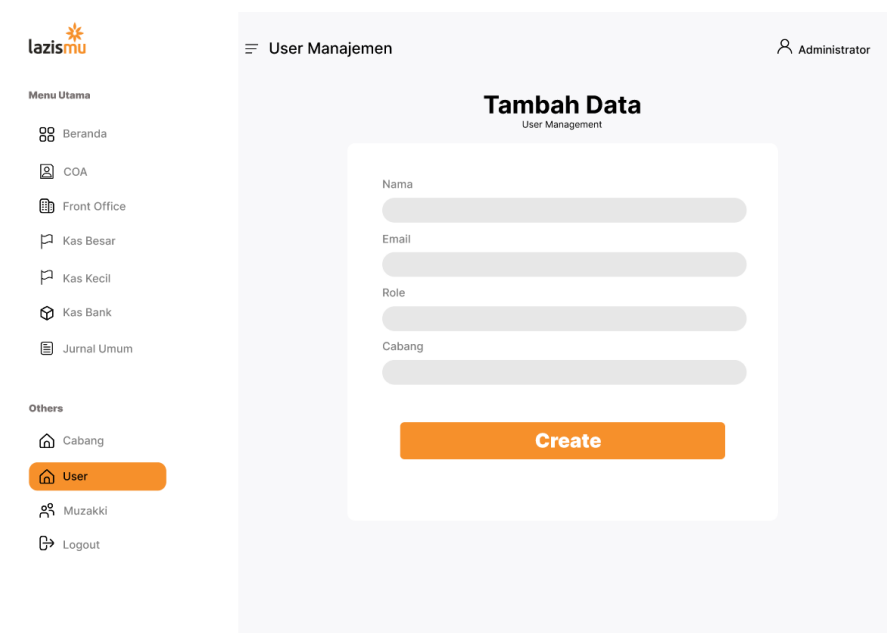
Gambar 3.26 Tampilan Tambah Data Pada Cabang

❖ User

Pada menu *User* terdapat tabel yang berisi nama, *email*, *role*, dan cabang kemudian pada menu tambah seperti di atas mengisikan menu yang diperlukan pada tabel sebelumnya. Terdapat *dropdown* di atas tabel untuk memilih *role* yang digunakan untuk mencari *user* sesuai *role*-nya.



Gambar 3.27 Tampilan User



Gambar 3.28 Tampilan Tambah Data Pada User

❖ Muzakki

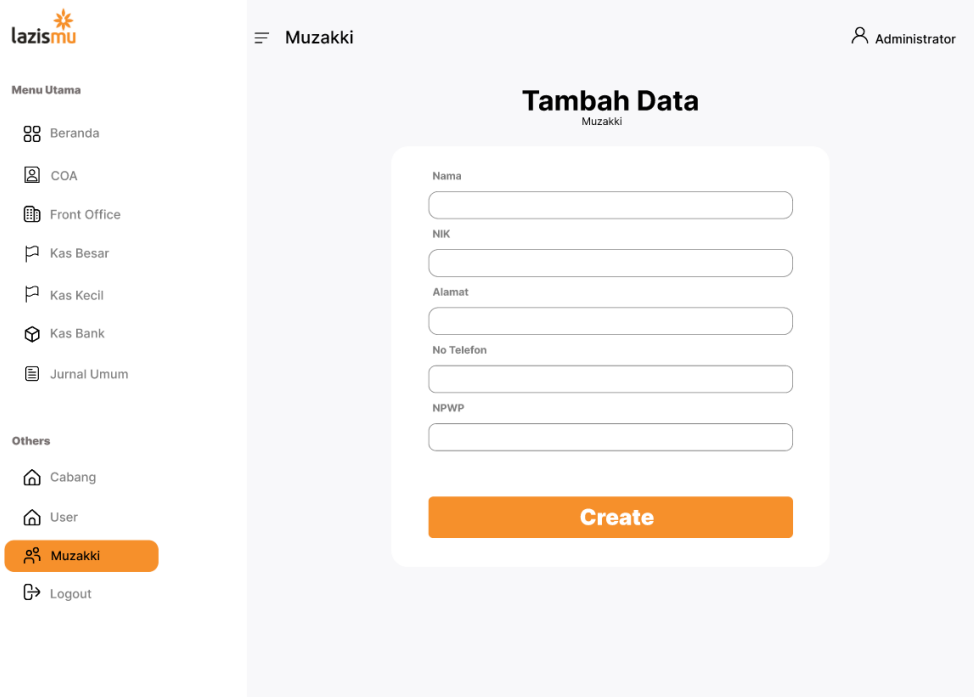
Pada menu Muzakki masih sama dari menu sebelumnya yang membedakan hanya pada jumlah data muzakki dan pada menu tambah yang hanya ada 1 form.

Kemudian *Logout* ketika diklik akan kembali ke menu *Sign in* di awal.

Pada aksi setiap tabel digunakan untuk *delete* dan edit data.

The screenshot shows the 'Muzakki' management page. On the left is a sidebar with the 'lazismu' logo and a menu. The main area has a header 'Muzakki' and 'Administrator' user info. Below is a 'Statistics' section showing 'Jumlah Muzakki' as 1. A 'Data Kasir' section includes filters for 'Kas Besar' and 'Kas Kecil', date pickers, and an 'Export Excel' button. A search bar for 'NIK' and a 'Tambah' button are also present. A table with columns: No, Muzakki, Lembaga, No telfon, NIK, NPWP, Aksi is shown, currently displaying 1 record. A pagination bar at the bottom indicates 'Showing 1 - 10 of 1'.

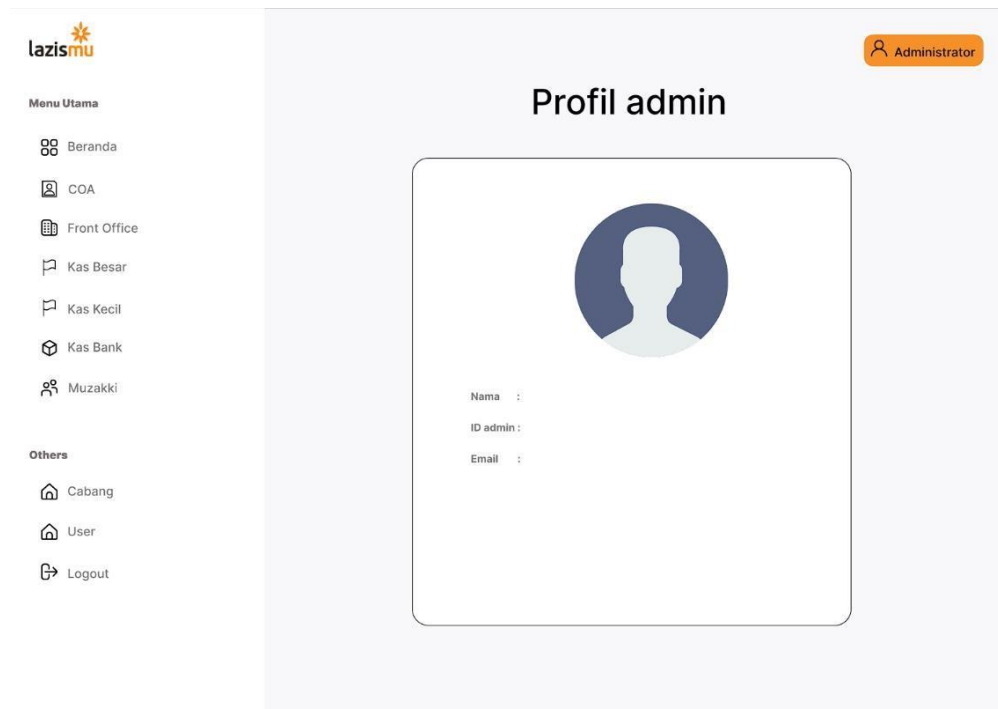
Gambar 3.29 Tampilan Muzakki



Gambar 3.30 Tampilan Tambah Data Pada Muzakki

❖ Menu Administrator

Pada menu Administrator berisi profil admin yang sedang mengakses, yang berisikan foto admin, nama, id admin, dan *email*.



Gambar 3. 31 Tampilan Menu administrator

❖ Pengujian Sistem

Berdasarkan *website* yang telah dibuat maka dilakukan pengujian fungsional sistem dengan metode *Black Box* dan UAT (*User Acceptance Test*). Berikut adalah tabel pengujian Sistem Keuangan Lazismu dengan pengujian *Black Box* sebagai berikut.

Tabel 3.1 Pengujian Black Box

Pengujian Black Box				
Admin				
NO.	FITUR	Input	Expected Result (Output)	Actual Result
1.	Halaman Login	Admin menginputkan <i>Username</i> dan <i>Password</i> lalu mengisi dan mengklik <i>button Sign In</i>	Masuk Ke dalam halaman <i>dashboard</i> sistem	Berhasil
		Admin menginputkan <i>Username</i> benar dan <i>Password</i> salah lalu mengklik <i>button Sign In</i>	Tidak bisa masuk ke dalam sistem atau masih berada di menu <i>Sign In</i>	Berhasil
		Admin menginputkan <i>Username</i> salah dan <i>password</i> benar lalu mengklik <i>button Sign In</i>	Tidak bisa masuk ke dalam sistem atau masih berada di menu <i>Sign In</i>	Berhasil
		Admin menginputkan <i>Username</i> dan <i>password</i> salah lalu mengklik <i>button Sign In</i>	Tidak bisa masuk ke dalam sistem atau masih berada di menu <i>Sign In</i>	Berhasil
2.	Halaman Menu CoA	Admin menginputkan nama akun yang akan dicari	Data nama akun yang dicari muncul	Berhasil
		Admin mengklik <i>button</i> Tambah	Masuk ke dalam halaman tambah data CoA	Berhasil
		Admin menginputkan nomor urut, nama akun, Tipe (Debit/Kredit), <i>Front Office</i> (Iya/Tidak), lalu klik Tambah	Data akun baru ditambahkan	Berhasil
		Admin mengklik Edit pada kolom Aksi	Masuk ke dalam halaman edit data	Berhasil
		Admin mengubah data CoA, lalu mengklik <i>button Update</i>	Data akun CoA berubah	Berhasil
		Admin mengklik <i>Delete</i> pada kolom Aksi, lalu mengklik Konfirmasi	Data CoA terhapus	Berhasil
	Admin menginputkan tanggal awal dan tanggal akhir	Tanggal sesuai inputan kalender	Berhasil	

		Admin klik <i>button Export to excel</i>	Data di- <i>export</i> dalam format excel	Tidak berhasil Catatan: nomor masih teracak, muncul kolom cabang id, <i>Front Office</i> opsi Ya/Tidak nya tidak muncul.
		Admin mengklik <i>button Import</i> dan memilih <i>file</i> dari eksternal kemudian klik Konfirmasi	Data berhasil tampil pada tabel akun	Berhasil
		<i>Summary</i> untuk informasi jumlah akun CoA Debit dan CoA Kredit	Informasi Jumlah Akun CoA Debit dan COA Kredit muncul	Berhasil
		Menambahkan akun CoA Debit atau Kredit maka jumlah di <i>Summary</i> berubah sesuai dengan jumlah yang ada	Informasi jumlah Akun CoA debit atau kredit berubah	Berhasil
		Menghapus akun CoA Debit atau Kredit maka jumlah di <i>Summary</i> berubah	Informasi jumlah Akun CoA debit atau kredit berkurang	Berhasil
3.	Menu <i>Front Office</i>	Admin mengklik <i>button Tambah</i>	Masuk ke dalam halaman tambah data	Berhasil
		Admin menginputkan no. bukti yang akan dicari	Data yang dicari berdasarkan No. bukti muncul	Berhasil
		Admin menginputkan uraian transaksi yang akan dicari	Data yang dicari berdasarkan uraian transaksi muncul	Berhasil
		Admin menginputkan no. bukti penerima, uraian transaksi, tanggal, muzakki, penerima, akun debit, akun kredit, tempat bayar, jumlah klik Tambah	Data yang melakukan ZIS ditambahkan	Berhasil
		Admin mengklik aksi Edit pada kolom Aksi	Masuk ke dalam halaman edit data	Berhasil
		Admin mengubah data muzakki lalu mengklik <i>button Update</i>	Data muzakki berubah	Berhasil
		Admin mengklik aksi <i>Delete</i> pada kolom Aksi, lalu klik konfirmasi	Data muzakki terhapus	Berhasil
		Admin menginputkan tanggal awal dan tanggal akhir	Tanggal sesuai inputan kalender	Berhasil
		Admin klik <i>button Export to excel</i>	Data di- <i>export</i> dalam format excel	Berhasil

				Catatan: <i>miss</i> di bagian uraian transaksi.
		Admin mengklik <i>button</i> Tambah, jika belum ada nama muzakki bisa mengklik <i>link</i> Tambah Muzakki	Masuk ke dalam halaman Tambah Akun Muzakki di menu Muzakki	Berhasil
		<i>Summary</i> untuk informasi saldo pemasukan dari <i>Front Office</i>	Muncul pemasukan setelah data diinputkan	Berhasil
		Menambahkan data <i>Front Office</i> maka jumlah saldo pemasukan di <i>Summary</i> berubah sesuai dengan jumlah yang diinputkan	<i>Summary</i> saldo pemasukan bertambah	Berhasil
		Menghapus data <i>Front Office</i> maka jumlah di <i>Summary</i> berubah	<i>Summary</i> Saldo pemasukan berkurang	Berhasil
4.	Menu Kas Besar	Admin mengklik <i>button</i> Tambah	Masuk ke dalam halaman tambah data	Berhasil
		Admin menginputkan no. bukti yang akan dicari	Data yang dicari berdasarkan No. bukti muncul	Berhasil
		Admin menginputkan uraian transaksi yang akan dicari	Data yang dicari berdasarkan uraian transaksi muncul	Berhasil
		Admin memilih input pengeluaran atau pemasukan, lalu menginputkan no. bukti kas, tanggal, ref, uraian transaksi, nama akun debit, nama akun kredit, jumlah, lalu klik <i>button</i> Tambah	Data muzakki yang melakukan transaksi <i>cash</i> ditambahkan	Berhasil
		Admin mengklik edit pada kolom Aksi	Masuk ke dalam halaman edit data kas besar	Berhasil
		Admin mengubah data transaksi kas besar, lalu mengklik <i>button Update</i>	Data kas besar berubah	Berhasil
		Admin mengklik <i>Delete</i> pada kolom Aksi, lalu mengklik konfirmasi	Data kas besar terhapus	Berhasil
		Admin menginputkan tanggal awal dan tanggal akhir	Tanggal sesuai inputan kalender	Berhasil
		Admin klik <i>button Export to excel</i>	Data diexport dalam format excel	Tidak berhasil Catatan: data tidak masuk di excel
		<i>Summary</i> untuk informasi saldo pemasukan, pengeluaran dan jumlah saldo dari kas besar	Muncul saldo pemasukan, saldo pengeluaran dan jumlah saldo setelah data diinputkan	Berhasil

		Menambahkan data kas besar maka jumlah saldo pemasukan, pengeluaran dan jumlah saldo di <i>Summary</i> berubah sesuai dengan jumlah yang diinputkan	<i>Summary</i> saldo pemasukan atau pengeluaran dan jumlah saldo bertambah	Berhasil
		Mengedit data kas besar yang termasuk pemasukan atau pengeluaran maka jumlah di <i>Summary</i> berubah	<i>Summary</i> saldo pemasukan atau pengeluaran dan jumlah saldo berubah	Berhasil
		Menghapus data kas besar yang termasuk pemasukan atau pengeluaran maka jumlah di <i>Summary</i> berubah	<i>Summary</i> saldo pemasukan atau pengeluaran dan jumlah saldo berkurang	Berhasil
5.	Menu Kas Kecil	Admin mengklik <i>button</i> Tambah	Masuk ke dalam halaman tambah data kas kecil	Berhasil
		Admin menginputkan no. bukti yang akan dicari	Data yang dicari berdasarkan No. bukti muncul	Berhasil
		Admin menginputkan uraian transaksi yang akan dicari	Data yang dicari berdasarkan uraian transaksi muncul	Berhasil
		Admin menginputkan no. bukti kas, uraian transaksi, tanggal, penerima, akun debit, akun kredit, jumlah klik Tambah	Data transaksi kas kecil ditambahkan	Berhasil
		Admin mengklik Edit pada kolom Aksi	Masuk ke dalam halaman edit data	Berhasil
		Admin mengubah data transaksi kas kecil, lalu mengklik <i>button Update</i>	Data kas kecil berubah	Berhasil
		Admin mengklik <i>Delete</i> pada kolom Aksi, lalu mengklik konfirmasi	Data kas kecil terhapus	Berhasil
		Admin menginputkan tanggal awal dan tanggal akhir	Tanggal sesuai inputan kalender	Berhasil
		Admin klik <i>button Export to excel</i>	Data di- <i>export</i> dalam format excel	Tidak berhasil Catatan: data tidak masuk di excel
		<i>Summary</i> untuk informasi saldo pemasukan, pengeluaran dan jumlah saldo dari kas kecil	Muncul saldo pemasukan, saldo pengeluaran dan jumlah saldo setelah data diinputkan	Berhasil
		Menambahkan data kas kecil maka jumlah saldo pemasukan, pengeluaran dan jumlah saldo di	<i>Summary</i> saldo pemasukan atau pengeluaran dan	Berhasil

		<i>Summary</i> berubah sesuai dengan jumlah yang diinputkan	jumlah saldo bertambah	
		Mengedit data kas kecil yang termasuk pemasukan atau pengeluaran maka jumlah di <i>Summary</i> berubah	<i>Summary</i> saldo pemasukan atau pengeluaran dan jumlah saldo berubah	Berhasil
		Menghapus data kas kecil yang termasuk pemasukan atau pengeluaran maka jumlah di <i>Summary</i> berubah	<i>Summary</i> saldo pemasukan atau pengeluaran dan jumlah saldo berkurang	Berhasil
6.	Menu Kas Bank	Admin menambahkan data transaksi kas bank dengan mengklik <i>button</i> Tambah	Masuk ke dalam halaman tambah data	Berhasil
		Admin menginputkan no. bukti yang akan dicari	Data yang dicari berdasarkan No. bukti muncul	Berhasil
		Admin menginputkan uraian transaksi yang akan dicari	Data yang dicari berdasarkan uraian transaksi muncul	Berhasil
		Admin menginputkan no. bukti kas, uraian transaksi, tanggal, ref, nama akun debit, nama akun kredit, jumlah, nama kasir klik <i>button</i> Tambah	Data kas bank ditambahkan	Berhasil
		Admin mengklik Edit pada kolom Aksi	Masuk ke dalam halaman edit data	Berhasil
		Admin mengubah data kas bank lalu mengklik <i>button Update</i>	Data kas bank berubah	Berhasil
		Admin mengklik <i>Delete</i> pada kolom Aksi, lalu mengklik konfirmasi	Data kas bank terhapus	Berhasil
		Admin menginputkan tanggal awal dan tanggal akhir	Tanggal sesuai inputan kalender	Berhasil
		Admin klik <i>button Export to excel</i>	Data diexport dalam format excel	Tidak berhasil Catatan: data tidak masuk di excel
		<i>Summary</i> untuk informasi saldo pemasukan, pengeluaran dan jumlah saldo dari kas bank	Muncul saldo pemasukan, saldo pengeluaran dan jumlah saldo setelah data diinputkan	Berhasil
		Menambahkan data kas bank maka jumlah saldo pemasukan, pengeluaran dan jumlah saldo di <i>Summary</i> berubah sesuai dengan jumlah yang diinputkan	<i>Summary</i> saldo pemasukan atau pengeluaran dan jumlah saldo bertambah	Berhasil
		Mengedit data kas bank yang termasuk pemasukan atau	<i>Summary</i> saldo pemasukan atau	Berhasil

		pengeluaran maka jumlah di <i>Summary</i> berubah	pengeluaran dan jumlah saldo berubah	
		Menghapus data kas bank yang termasuk pemasukan atau pengeluaran maka jumlah di <i>Summary</i> berubah	<i>Summary</i> saldo pemasukan atau pengeluaran dan jumlah saldo berkurang	Berhasil
7.	Menu Jurnal Umum	Admin mengklik <i>button</i> Tambah	Masuk ke dalam halaman tambah data	Berhasil
		Admin menginputkan no. bukti yang akan dicari	Data yang dicari berdasarkan No. bukti muncul	Berhasil
		Admin menginputkan uraian transaksi yang akan dicari	Data yang dicari berdasarkan uraian transaksi muncul	Berhasil
		Admin menginputkan no. bukti penerima, uraian transaksi, tanggal, ref, akun debit, akun kredit, jumlah klik Tambah	Data transaksi baru ditambahkan	Berhasil
		Admin mengklik Edit pada kolom Aksi	Masuk ke dalam halaman edit data	Berhasil
		Admin mengubah data transaksi lalu mengklik <i>button Update</i>	Data transaksi berubah	Berhasil
		Admin mengklik <i>Delete</i> pada kolom Aksi, lalu mengklik konfirmasi	Data transaksi terhapus	Berhasil
		Admin menginputkan tanggal awal dan tanggal akhir	Tanggal sesuai inputan kalender	Berhasil
		Admin klik <i>button Export to excel</i>	Data di- <i>export</i> dalam format excel	Tidak berhasil Catatan: data tidak masuk di excel
		<i>Summary</i> untuk informasi jumlah saldo pada jurnal umum	Informasi Jumlah saldo muncul	Berhasil
		Menambahkan saldo maka jumlah di <i>Summary</i> berubah sesuai dengan jumlah yang ada	Informasi jumlah saldo berubah	Berhasil
		Menghapus saldo maka jumlah di <i>Summary</i> berubah	Informasi jumlah saldo berkurang	Berhasil
8.	Menu Cabang	Admin menginputkan nama cabang yang akan dicari	Data yang dicari berdasarkan nama cabang muncul	Berhasil
		Admin mengklik <i>button</i> Tambah	Masuk ke dalam halaman tambah data	Berhasil
		Admin menginputkan nama cabang klik Tambah	Data cabang baru ditambahkan	Berhasil
		Admin mengklik Edit pada kolom Aksi	Masuk ke dalam halaman edit data	Berhasil
		Admin mengubah data cabang lalu mengklik <i>button Update</i>	Data cabang berubah	Berhasil
		Admin mengklik <i>Delete</i> pada kolom Aksi, lalu mengklik konfirmasi	Data cabang terhapus	Berhasil

		Summary untuk informasi jumlah cabang	Informasi Jumlah cabang muncul	Berhasil
		Menambahkan cabang maka jumlah di Summary berubah sesuai dengan jumlah yang ada	Informasi jumlah cabang berubah	Berhasil
		Menghapus cabang maka jumlah di Summary berubah	Informasi jumlah cabang berkurang	Berhasil
9.	Menu User	Admin mengklik <i>button</i> Tambah	Masuk ke dalam halaman tambah data <i>user</i>	Berhasil
		Admin menginputkan nama dan <i>email user</i> yang akan dicari	Data yang dicari berdasarkan nama <i>user</i> muncul	Berhasil
		Admin memilih <i>role</i> yaitu: Admin, <i>Front Office</i> , <i>Back Office</i>	Muncul data sesuai <i>role</i> yang di pilih	Berhasil
		Admin menginputkan nama, <i>email</i> , <i>password</i> , <i>role</i> , cabang klik Tambah	Data <i>user</i> baru ditambahkan	Berhasil
		Admin mengklik Edit pada kolom Aksi	Masuk ke dalam halaman edit data	Berhasil
		Admin mengubah data <i>user</i> lalu mengklik <i>button Update</i>	Data <i>user</i> berubah	Berhasil
		Admin mengklik <i>Delete</i> pada kolom Aksi, lalu mengklik konfirmasi	Data <i>user</i> terhapus	Berhasil
10.	Menu Muzakki	Admin menginputkan nama, NIK, Alamat, nomor telepon, NPWP kemudian mengklik <i>button</i> tambah	Masuk ke dalam halaman tambah akun muzakki	Berhasil
		Admin tidak menginputkan nama, NIK, Alamat, nomor telepon, NPWP kemudian mengklik <i>button</i> tambah	Tidak masuk ke dalam halaman muzakki	Berhasil
		Admin mengedit data muzakki dengan klik aksi edit	Masuk ke dalam halaman edit data muzakki	Berhasil
		Admin menghapus data muzakki	Data muzakki terhapus	Berhasil

Berikut adalah tabel pengujian UAT (*User Acceptance Test*) :

Tabel 3.2 Pengujian Sistem User Acceptance Test

No	Kebutuhan Fungsional yang akan diuji	Acceptance Criteria	Hasil (✓/X)	Keterangan Revisi
1	Sistem dapat menampilkan sistem keuangan Lazismu dalam bentuk <i>website</i>	Sistem akan menampilkan halaman <i>website</i> jika <i>user</i> mengakses domain	✓	
2	Sistem memiliki fitur <i>Login</i>	<i>User</i> akan diminta untuk <i>Login</i> sebelum masuk ke halaman utama <i>website</i>	✓	
3	Sistem dapat menampilkan menu COA, FO, Kas Besar, Kas Kecil, Kas Bank, <i>User</i> dan Cabang	Menu sistem keuangan meliputi menu COA, FO, Kas Besar, Kas Kecil, Kas Bank, <i>User</i> dan Cabang dengan masing-masing menampilkan tabel dan dapat melakukan input data	✓	Menu yang ditampilkan disesuaikan dengan data excel yang diberikan oleh Lazismu
4	Sistem dapat menampilkan pemasukan dan pengeluaran	Menampilkan data pemasukan dan pengeluaran yang dilakukan oleh <i>user</i>	✓	
5	Sistem memiliki fitur pencarian, manajemen <i>user</i> , unduh data keuangan dan CRUD	Terdapat Searching bar untuk mencari data dan terdapat fitur CRUD	✓	

C. Kesepakatan Proyek

Dalam pelaksanaan proyek ini tim Lovlace telah sepakat dengan pihak Lazismu untuk mengambil dan membuat proyek sesuai dengan apa yang telah diminta oleh pihak klien. Agar lebih jelasnya lagi, dapat dilihat dalam detail penjelasan di bawah ini:

1. Waktu Pengerjaan

Waktu yang digunakan untuk mengerjakan “Sistem Keuangan Lazismu” yaitu kurang lebih 3 bulan.

2. Fitur Proyek

Klien meminta untuk membuat *website* Sistem Keuangan dengan beberapa fitur diantaranya yaitu :

- *Login*

User akan diminta untuk *Login* saat memasuki *website*

- *Dashboard*

Dalam fitur ini dapat menampilkan halaman awal dari *Web Profile* setelah *Login*, terdapat menu-menu yang bisa diakses tergantung pada *role* dari *user*, terdapat juga statistik yang menampilkan jumlah pengeluaran dan pemasukan secara keseluruhan

- COA

Fitur ini bertujuan untuk menginputkan akun yang melakukan ZIS mulai dari kode, laporan, tipe (debit atau kredit)

- *Front Office*

Fitur ini bertujuan untuk menerima data ZIS dari kantor depan

- Kas besar

Fitur ini bertujuan untuk menampung data yang pembayarannya baik pengeluaran atau pemasukan dalam bentuk *cash*

- Kas kecil

Fitur ini bertujuan untuk menampung data yang pembayarannya baik pengeluaran atau pemasukan dengan transfer

- Kas bank

Fitur ini bertujuan untuk transaksi dari bank ke bank Lazismu

3. Biaya Proyek

Anggaran biaya sistem yang dijelaskan pada tabel 3.1 di bawah :

Tabel 3.3 Biaya Proyek

Pemasukan				
Jenis	Jumlah	Satuan	Harga	Jumlah
Iuran Anggota	7	Orang	50.000	Rp. 350.000
Total				Rp. 350.000
Pengeluaran				
Pengeluaran	Jumlah	Satuan	Harga	Jumlah
Hosting	2	Bulan	150.000	Rp. 350.000
Operasional	7	Orang	150.000	Rp. 1.050.000
Internet	7	Orang	50.000	Rp. 350.000
Transportasi	7	Orang	25.000	Rp. 175.000
Total				Rp. 1.925.000

D. Rencana Jadwal Pelaksanaan Proyek

Jadwal Perencanaan merupakan jadwal pelaksanaan dari kelompok Lovlace dalam proses “Sistem Keuangan Lazismu”. Waktu pelaksanaan untuk menyelesaikan tugas Manajemen Proyek Teknologi Informasi membutuhkan waktu yang kurang lebih 3 bulan. Waktu pelaksanaan dapat dilihat pada tabel berikut.

Tabel 3.4 Rencana Jadwal Proyek

No	Nama Kegiatan	Minggu Pelaksanaan															
		Mei				Juni				Juli				Agustus			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Penerjunan																
2	Analisis kebutuhan user, analisa																

	kebutuhan data, wawancara klien															
3	Perancangan kebutuhan sistem															
4	Perancangan desain <i>UI/UX</i>															
5	Proses slicing desain <i>UI/UX</i> ke dalam kodingan (<i>Front End</i>)															
6	Proses integrasi hasil <i>Front End</i> dengan <i>database</i> dan <i>coding (Back End)</i>															
7	Pengujian sistem															
8	Pemasangan <i>hosting</i> dan <i>domain</i>															
9	<i>Training</i> mitra															
10	Penyusunan laporan															

BAB IV Pelaksanaan MPTI

A. Realisasi Pembagian Tugas

Realisasi Pembagian Tugas proyek tim Lovlace telah dilaksanakan oleh anggota tim sesuai dengan tugasnya masing-masing.

- 1) *Project Manager & Tester* (Mohammad Amirul Kurniawan Putranto)

Tabel 4.1 Realisasi Tugas Project manager & Tester

No	Agenda Kerja	Hari/ Tanggal	Waktu	Hasil
1	Bersama-sama menyusun ERD	04 Juni 2023	5 Jam	ERD berhasil dibuat
2	Bertemu dengan Lazismu untuk mendiskusikan apa saja kebutuhannya dan bagaimana mengimplementasikannya	10 Juni 2023	2 Jam	Mendapatkan <i>file</i> excel untuk mempermudah dalam pembuatan sistem
3	Bertemu dengan Lazismu dan kelompok Stardust	04 Juli 2023	2 Jam	Pembagian tiap kelompok dan fitur atau menu apa yang dikerjakan tiap kelompok
4	Mengagendakan rapat internal untuk memperbaiki menu CoA dan FO	09 Juli 2023	3 Jam	Menyesuaikan menu dengan excel yang ada
5	Bertemu dengan Lazismu dan Pak Herman untuk mendiskusikan apa saja kebutuhan lain dari Lazismu dan bagaimana mengimplementasikannya	11 Juli 2023	2 Jam	Mengevaluasi kinerja tiap kelompok dan memberikan <i>job desc</i> yang sesuai
6	Bertemu dengan kelompok Stardust	17 Agustus 2023	1,5 Jam	Pembagian tiap kelompok untuk menyelesaikan sistem
7	Notulensi rapat bersama kelompok stardust ke-1	17 Mei 2023	2 Jam	Pembagian fitur yang akan dikerjakan masing-masing kelompok
8	Bertemu dengan Lazismu	23 Agustus 2023	2 Jam	Mempresentasikan sistem yang telah dibuat Mendapatkan beberapa revisi

2) Sekretaris & QAEng (Dinda Putri Dwi Yanti)

Tabel 4.2 Realisasi Tugas Sekretaris & QAEng

No	Agenda Kerja	Hari/ Tanggal	Waktu	Hasil
1	Notulensi Rapat Internal ke-1 : Pembagian <i>job desc</i> dan penentuan dosen pembimbing	16 Maret 2023	2 Jam	<ul style="list-style-type: none"> - Mohammad Amirul Kurniawan Putranto (PM & <i>Software Tester</i>) - Dinda Putri Dwi Yanti (Sekretaris & QAEng) - Faza Husain Muhammad Zaen (<i>Back-End Developer</i>) - Sindi Rella (<i>Data Analyst</i>) - Zakiya Nurul Baity (<i>Front-End Developer</i>) - Faqih 'Ainul Yaqin Nursaid (<i>Database Engineer</i>) - Abdilah Muttaqi (<i>UI/UX Designer</i>) <p>Dosen pembimbing : Murein Miksa Mardhia S.T., M.T.</p>
2	Notulensi Bimbingan bersama dosen ke-1	05 Mei 2023	1,5 Jam	Perkenalan anggota Memberitahukan <i>job desc</i> masing-masing anggota
3	Notulensi rapat bersama dosen pembimbing ke-2	09 April 2023	1 Jam	Diskusi serta dosen pembimbing memberikan arahan terkait program SPI dan pembahasan terkait sistem apa yang akan dipilih di antara dua sistem yang diajukan oleh klien
4	Notulensi rapat bersama klien ke-1	16 Mei 2023	2 Jam	Pendalaman materi proyek, pengenalan fitur dan pembagian <i>job desc</i> tim
5	Membuat Proposal proyek dan MOU	16 Mei – 17 Mei 2023	1 Jam	Proposal dan MOU telah disusun
6	Notulensi bimbingan bersama dosen pembimbing ke-3	17 Mei 2023	30 Menit	- Bimbingan mengenai logbook lebih lanjut, MOU dan

				laporan <i>progress</i> Mendapatkan revisi pada proposal dan MOU proyek
7	Notulensi rapat bersama kelompok Stardust ke-1	17 Mei 2023	2 Jam	Pembagian fitur yang akan dikerjakan masing-masing kelompok
8	Bertemu dengan Lazismu	23 Agustus 2023	2 Jam	- Mempresentasikan sistem yang telah dibuat - Mendapatkan beberapa revisi
9	Notulensi bimbingan bersama dosen pembimbing ke-4	26 Mei 2023	1 Jam	Mendapatkan tanda tangan <i>log book</i> kelompok
10	Notulensi rapat internal ke-2 : Pembahasan mengenai <i>job desc</i> kepada <i>system analyst, database engineer, dan front end</i>	4 Juni 2023	1 Jam	Mulai membuat <i>interface</i> yang telah dibuat, membuat ERD <i>database</i> , dan membuat implementasi kodingan tampilan awal
11	Notulensi rapat bersama klien ke-2	10 Juni 2023	2,5 Jam	Mendapatkan alur data keuangan yang akan digunakan untuk dijadikan <i>website</i>
12	Notulensi rapat bersama klien ke-3	04 Juli 2023	1,5 Jam	Membahas lanjutan program yang akan dibuat
13	Notulensi rapat internal ke-3 : <i>system analyst, database engineer, dan back end</i>	09 Juli 2023	3 Jam	<i>Database</i> sudah dibuat dan siap digunakan sesuai keperluan serta membuat <i>use case</i> dan membuat UML
14	Pembuatan laporan BAB 2	10 Juli 2023	1,5 Jam	Pembuatan laporan BAB 2 telah dikerjakan
15	Notulensi rapat bersama klien dan Pak Herman serta tim kelompok Stardust	11 Juli 2023	4 Jam	Revisi pada <i>back end</i> dibagikan kebutuhan Lazismu ke dalam daftar kebutuhan
16	Notulensi bimbingan bersama dosen pembimbing ke-5	17 Juli 2023	3 Jam	Bimbingan terkait hasil perancangan awal UML dan <i>UI Design</i> yang sudah dibuat dilanjutkan dengan <i>Front end</i> dan <i>Back end</i> . Mendapatkan revisi pada bagian <i>Data Analyst, front end, back end, dan laporan</i> . pada

				bagian <i>Data Analyst</i> , frontend, back end, dan laporan.
17	Membuat laporan BAB 3	20 Juli 2023	2 Jam	Pembuatan laporan BAB 3 telah dikerjakan
18	Notulensi rapat internal ke-4	11 Agustus 2023	2 Jam	<i>System analyst</i> membuat <i>Activity</i> diagram dan masing masing anggota melanjutkan <i>job desc</i> -nya
19	Membuat laporan BAB 4	12 Agustus 2023	1,5 Jam	Pembuatan laporan BAB 4 telah dikerjakan
20	Notulensi rapat bersama kelompok Stardust	17 Agustus 2023	2 Jam	Mendapatkan Analisis kebutuhan bisnis, kebutuhan <i>user</i> , kebutuhan fungsional, non-fungsional, dan kebutuhan sistem berhasil dibuat
21	Membuat logbook kelompok dari keseluruhan kegiatan MPTI	20 Agustus 2023	1 Jam	Pembuatan <i>log book</i> kelompok dari keseluruhan kegiatan telah dikerjakan
22	Notulensi rapat bersama klien ke-4	23 Agustus 2023	3 Jam	Melakukan evaluasi kekurangan dari sistem
23	Melakukan Pengujian sistem (<i>Black Box Testing</i>)	24 Agustus 2023	4 Jam	Melakukan pengujian <i>Black Box</i> sistem <i>user</i> dan admin, di mana hasil dari pengujian tersebut valid semua atau berhasil sesuai yang diharapkan.
24	Melakukan Pengujian <i>User Acceptance Testing</i>	25 Agustus 2023	4 Jam	Melakukan pengujian UAT pada sistem berbentuk <i>website</i> , fitur <i>Login</i> , menu CoA, FO, Kas Besar, Kas Kecil, Kas Bank, <i>User</i> , Cabang, sistem dapat menampilkan pemasukkan dan pengeluaran, fitur pencarian, manajemen <i>user</i> , unduh data keuangan, dan CRUD , di mana hasilnya akhir dari keseluruhan pengujian tersebut "ACCEPTABLE".

3) *Data Analyst* (Sindi Rella)

Tabel 4.3 Realisasi Tugas *Data Analyst*

No	Agenda Kerja	Hari/ Tanggal	Waktu	Hasil
1	Analisis sistem yang akan dibuat	25 Mei 2023	5 Jam	Melakukan analisis terhadap sistem yang akan dibuat seperti: Menentukan atribut yang digunakan oleh sistem
2	Merancang diagram sistem	04 Juni 2023	30 Menit	Berhasil membuat ERD
3	Implementasi rancangan <i>database</i> yang telah dibuat	09 Juli 2023	5,5 Jam	<i>Database</i> sudah dibuat dan siap digunakan sesuai keperluan serta membuat <i>Class Diagram</i>
4	Membuat <i>Activity Diagram</i> .	11 Agustus 2023	4 Jam	Menyusun kerangka <i>Activity diagram</i>
5	Menganalisis kebutuhan bisnis, kebutuhan <i>user</i> , kebutuhan fungsional, non-fungsional, dan kebutuhan sistem	17 Agustus 2023	2 Jam	Analisis kebutuhan bisnis, kebutuhan <i>user</i> , kebutuhan fungsional, non-fungsional, dan kebutuhan sistem berhasil dibuat

4) *UI/UX Designer* (Abdilah Muttaqin)

Tabel 4.4 Realisasi Tugas *UI/UX Designer*

No	Agenda Kerja	Hari/ Tanggal	Waktu	Hasil
1	Membuat <i>UI Sign In</i> dan rancangan <i>UI/UX</i>	17 Mei 2023	4 Jam	Terbentuknya <i>UI</i> bagian sign, rancangan awal bagian beranda.
2	Membuat <i>UI Front Office</i> , kas besar, dan kas kecil	4 Juni 2023	3 Jam	Menghasilkan <i>UI</i> untuk <i>Front Office</i> , kas besar, dan kas kecil

3	Membuat <i>UI</i> kas bank	12 Juni 2023	2 Jam	Pembuatan <i>UI</i> bagian kas bank.
4	Membuat <i>UI</i> cabang, <i>user</i> , dan log out	9 Juli 2023	5 Jam	Pembuatan <i>UI</i> cabang, <i>user</i> , dan <i>log out</i> kemudian melengkapi <i>UI</i> bagian <i>sign in</i> menambahkan mode tak terlihat.
5	Merapikan <i>UI</i> yang telah dibuat	10 Juli 2023	6 jam	Meluruskan dan merapikan posisi seluruh <i>UI</i> yang telah dibuat.
5	Melakukan <i>prototype</i>	16 Juli 2023	3 Jam	<i>UI</i> bagian <i>Sign In</i> , <i>Front Office</i> , dan kas besar berhasil di <i>prototype</i> .
6	Melakukan <i>prototype</i>	20 Juli 2023	5 Jam	<i>UI</i> bagian kas kecil, kas bank, cabang, <i>user</i> , dan <i>log out</i> .
7	Revisi penambahan halaman CoA dan muzaki, dan mengubah <i>UI user</i>	23 Juli 2023	4 Jam	Menambahkan <i>UI</i> bagian CoA, dan muzaki, serta mengubah <i>UI user</i> .
8	Membuat <i>form</i> untuk tambah data	25 Juli 2023	4 Jam	Pembuatan <i>UI</i> untuk <i>form</i> tambah data pada menu CoA, <i>Front Office</i> , kas besar, kas kecil, kas bank, muzaki, cabang, dan <i>user</i> .

5) *Front-End Developer* (Zakiya Nurul Baity)

Tabel 4.5 Realisasi Tugas *Front-End Developer*

No	Agenda Kerja	Hari/ Tanggal	Waktu	Hasil
1	Menambah menu muzakki	11 Juli 2023	2 Jam	Menambah menu muzakki, menu ini untuk menginputkan data muzakki dari <i>Front Office</i>
2	Membuat <i>form</i> tambah data di menu muzakki	16 Juli 2023	2 Jam	Membuat <i>form</i> tambah data, mulai dari nama muzakki, lembaga, no. telepon, NIK, dan NPWP
3	Menambahkan jumlah saldo di menu kas bank, kas besar, kas kecil	23 Juli 2023	3 Jam	Menambahkan <i>summary</i> jumlah saldo di menu kas besar, kas kecil dan kas bank yang tadinya hanya pengeluaran dan pemasukan saja
4	Menambah nomor urut bukti di <i>Front Office</i>	4 Agustus 2023	1 Jam	Menambahkan nomor urut bukti di tambah data di <i>Front Office</i>

5	Membantu <i>back end</i> memperbaiki fitur tambah data di kas besar dan kas kecil	10 Agustus 2023	2 Jam	Memperbaiki fitur tambah data di kas besar dan kas kecil di bagian nomor bukti kas
6	Memperbaiki menu CoA	17 Agustus 2023	1 Jam	Atribut laporan dihapus jadi hanya ada nomor urut, nama akun, dan posisi saldo saja dan mengganti kode menjadi nomor urut
7	Menambah fitur di CoA	5 September 2023	15 Menit	Menambah fitur <i>select all</i> di <i>dropdown</i> tipe di CoA
8	Memperbaiki fitur tambah data di menu muzakki	5 September 2023	5 Menit	Mengganti alamat menjadi <i>text area</i>
9	Menambah fitur di <i>Front Office</i>	5 September 2023	30 Menit	Menambah <i>link</i> tambah data muzakki di menu tambah data
10	Menambah fitur menu CoA	5 September 2023	30 Menit	Menambahkan <i>radio button</i> untuk memfilter nama akun CoA ya atau tidak
11	Memperbaiki fitur tambah data di menu muzakki	5 September 2023	5 Menit	Mengganti nomor telepon menjadi <i>phone</i>
12	Menambah fitur di menu muzakki	5 September 2023	1 Jam	Menambah fitur edit data di menu muzakki

6) *Back-End Developer* (Faza Husain Muhammad Zaen)

Tabel 4.6 Realisasi Tugas *Back-End Developer*

No	Kegiatan	Hari/ Tanggal	Waktu	Hasil
1	Bersama-sama menyusun ERD	4 Juni 2023	5 Jam	ERD dibuat sesuai dengan kebutuhan sistem
2	Memperbaiki <i>back-end</i> dan <i>database</i>	09 Juli 2023	5,5 Jam	<i>Back-end</i> dan <i>database</i> diperbaiki dan siap digunakan sesuai keperluan
3	Bertemu dengan Lazismu untuk mendiskusikan apa saja kebutuhan lain dari Lazismu dan bagaimana mengimplementasikannya	11 Juli 2023	4 Jam	Menambahkan kebutuhan Lazismu ke dalam daftar kebutuhan

4	Memperbaiki dan menyesuaikan fungsi CRUD pada setiap menu	11 Agustus 2023	4 Jam	Aplikasi sudah dapat menambah, mengubah, dan menghapus data pada tabel
5	Memperbaiki fitur CoA dalam sistem	17 Agustus 2023	2 Jam	Menghapus laporan pada CoA dan mengganti kode dengan nomor urut
6	Memperbaiki fitur <i>import</i>	6 September 2023	1 Jam	Fitur <i>import</i> sudah bisa berfungsi dengan baik
7	Memperbaiki nomer urut pada FO	6 September 2023	1 Jam	Nomer urut pada FO sudah bisa tampil dan diinput
8	Merubah agar menu KFO hanya menampilkan akun <i>Front Office</i> saja	6 September 2023	1 Jam	Menambah atribut pada COA sebagai pengkondisian dari akun lalu menambahkan syarat pada FO untuk akun tampil
9	Memperbaiki <i>delete</i> pada menu muzakki	6 September 2023	1 Jam	Muzakki sudah dapat dihapus dengan fitur <i>delete</i>
10	Menambahkan fitur <i>update</i> muzakki di bagian back-end	6 September 2023	1 Jam	Muzakki sudah dapat <i>diupdate</i> atribut demi atribut dengan fitur <i>update</i>

7) *Database Engineer* (Faqih 'Ainul Yaqin Nursaid)

Tabel 4.7 Realisasi Tugas Database Engineer

No	Kegiatan	Hari / Tanggal	Waktu	Hasil
1	Penyusunan <i>Entity Relationship Diagram</i> (ERD) dengan tim	4 Juni 2023	5 Jam	ERD berhasil dibuat sesuai yang dibutuhkan untuk sistem
2	Cek dan perbaikan <i>database</i> untuk sistem	09 Juli 2023	5,5 Jam	<i>Database</i> diperbaiki sesuai kebutuhan dan siap digunakan sesuai keperluan

3	Bertemu dengan klien yaitu Lazismu untuk diskusi kebutuhan lain dari Lazismu dan bagaimana mengimplementasikannya	11 Juli 2023	4 Jam	Menambahkan kebutuhan Lazismu ke dalam daftar kebutuhan
4	Memperbaiki fungsi <i>Create, Read, Update, dan Delete (CRUD) database</i> pada setiap menu sistem	11 Agustus 2023	4 Jam	Perbaikan pada fungsi CRUD <i>database</i> berhasil dibuat dan sistem sudah dapat menambah, mengubah, dan menghapus data pada tabel
5	Memperbaiki <i>database</i> untuk fitur yang ada pada menu <i>Chart of Account (CoA)</i> dalam sistem	17 Agustus 2023	2 Jam	<i>Database</i> untuk CoA berhasil diperbaiki dengan perbaikan yaitu menghapus laporan pada CoA dan mengganti kode dengan nomor urut

B. Realisasi Jadwal Pelaksanaan Proyek

Tabel 4.8 Realisasi Jadwal Pelaksanaan Proyek

No	Nama Kegiatan	Minggu Pelaksanaan																Realisasi	
		Mei				Juni				Juli				Agustus					
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
1	Penerjunan																	Ya	
2	Analisis kebutuhan <i>user</i> , analisa kebutuhan data, wawancara klien																	Ya	
3	Perancangan kebutuhan sistem																	Ya	
4	Perancangan desain <i>UI/UX</i>																	Ya	
5	Proses <i>slicing</i> desain <i>UI/UX</i> ke dalam kodingan (<i>Front End</i>)																	Ya	
6	Proses integrasi <i>Front End</i> dengan <i>database</i> dan <i>coding (Back End)</i>																	Ya	
7	Pengujian sistem																	Ya	
8	Pemasangan <i>hosting</i> dan <i>domain</i>																		Tidak
9	<i>Training</i> mitra																	Ya	
10	Penyusunan laporan																		Ya

C. Realisasi Keberlanjutan Program

Keberlanjutan Proyek Sistem Keuangan Lazismu ini tentu akan selalu memerlukan pengembangan seiring dengan berjalannya waktu serta kebutuhan mitra/klien. Oleh karena itu, jika sewaktu-waktu pihak Lazismu ingin mengubah atau menambah fungsionalitas sistem ini, proyek ini dapat dilanjutkan oleh tim Lovlace atau pihak lain.

BAB V

Penutup

A. Kesimpulan

Adapun kesimpulan yang dihasilkan dari pembuatan Manajemen Proyek Teknologi Informasi “ Sistem Keuangan Lazismu” adalah sebagai berikut :

1. Sistem Keuangan Lazismu telah berhasil dibuat dan dapat membantu pihak Lazismu dalam mengelola pemasukan dan pengeluaran keuangan
2. Fitur-fitur ini dapat membantu klien mengelola data keuangan dalam pemasukan dan pengeluaran
3. Sistem Keuangan ini dapat meminimalkan kesalahan dalam mengelola data pegawai
4. Sistem informasi kepegawaian ini dapat mengefisienkan waktu klien dalam pengelolaan data muzakki
5. Memudahkan dalam pengelolaan data muzakki

B. Saran

Aplikasi yang telah dibangun ini masih ada beberapa bagian yang harus dikembangkan lagi. Oleh karena itu, perlu adanya pengembangan sistem dengan menambahkan fitur-fitur untuk melengkapi *website* Sistem Keuangan Lazismu sehingga menjadi lebih kompleks, sebagai berikut:

1. Menu Muzakki
2. *Export* data ke format .xlsx
3. Fitur *Drop Down* untuk CoA dan Muzakki
4. Fitur Mencetak Kwitansi

DAFTAR PUSTAKA

<https://Lazismudiy.or.id/>

LAMPIRAN

i. Proposal MPTI

PROPOSAL

SISTEM KEUANGAN LAZISMU

YOGYAKARTA



PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITA AHMAD DAHLAN

2023

A. Sistem Keuangan Lazismu

Sistem Keuangan Lazismu adalah sebuah sarana atau alat yang bisa digunakan oleh Lazismu untuk mempermudah mengelola, memantau, mengarahkan, serta meningkatkan efisiensi dan efektivitas data yang disajikan secara akurat dan tepat waktu pada Lazismu. melalui sistem ini, Lazismu bisa mengelola pemasukan dan pengeluaran secara dinamis.

B. Sistem yang ditawarkan

1. Fitur

- *Login*
- Beranda
- *Front Office*
- CoA
- *Front Office*
- Kas besar
- Kas kecil
- Kas Bank

Contoh sketsa *website* :

Lazismu

Sign In

Email

Password

Sign In

Administrator

Lazismu

Menu Utama

Beranda

Akun Debit Kredit

Front Office

Kas Besar

Kas Kecil

Kas Bank

Jurnal Umum

Others

Cabang

User

Logout

Beranda

Kas Besar Front Off Kas Kecil Kas Bank Jurnal

Statistik

	Data Masuk	Data Keluar
Kas Besar	1.8	
Front Office	1.7	
Kas Besar	1.6	
Kas Kecil	1.5	
Kas Bank	1.4	
Jurnal Umum	1.3	
Others	1.2	
Cabang	1.1	
User	1	
Logout	0.9	
	0.8	
	0.7	
	0.6	
	0.5	
	0.4	
	0.3	
	0.2	
	0.1	
	0	

2. Platform Aplikasi

- Web

3. Spesifikasi Sistem Aplikasi

- a. Jenis sistem aplikasi : Web
- b. Bahasa Pemrograman : PHP, JavaScript, HTML, CSS
- c. Framework : Bootstrap, Laravel 9/Code Igniter 3, Vue
- d. *Database* : MySQL
- e. Domain : Menyesuaikan Keinginan Klien

4. Fasilitas yang didapatkan

a. Beranda

Statistik perkembangan pemasukan dan pengeluaran dalam bentuk linechart. Pada sidebar terdapat menu bernada, COA, Frontoffice, Kas Besar, Kas Kecil, Kas bank, Muzakki, dan Other yang berisi menu cabang, *user*, kemudian tombol *Logout*. Setiap menu yang diklik akan berubah warna menjadi oranye seperti pada menu beranda di atas.

b. Manajemen *user*

Pada fitur ini admin dapat melakukan pencarian dari data *user* serta melakukan CRUD selanjutnya admin dapat melakukan *import* dan ekspor berbentuk *xlsx*. Selain itu pada COA terdapat *summary* mengenai jumlah akun debit dan akun debit.

C. Kegiatan

1) Melakukan Wawancara Dan Observasi dengan Klien

Pada proses ini kami akan memberikan beberapa pertanyaan terkait pandangan terhadap *website*. dan juga melakukan observasi untuk mendapatkan data seperti fitur apa saja yang di inginkan oleh pihak klien.

2) Perancangan *User Interface* dan *user Experience*

Pada proses ini hasil dari wawancara dan observasi akan disimpulkan menjadi sebuah data. Di mana data tersebut akan dikelola oleh *User Experience* untuk dijadikan panduan dalam membentuk *User Interface* yang nyaman untuk digunakan oleh *user*.

3) Perancangan Kebutuhan Sistem

Proses ini digunakan untuk merancang sistem apa saja yang diutuhkan oleh *user* supaya dapat di implementasikan dengan jelas dan tidak ada kesalahan dalam sistem tersebut.

4) Proses Slicing Desain ke dalam *Coding* (FrontEnd)

Pada proses ini Front-End akan melakukan sebuah Implementasi Design yang mana Design yang diterima dari *UI/UX* akan di implementasikan menggunakan Bahasa pemrograman yang telah di sepakati, dan akan dibuat secara responsive untuk berbagai tampilan seperti : tampilan browser Laptop, Mobile, Dll.

5) Perancangan Detail Proyek (*Database*)

Pada perancangan *database* ini digunakan untuk merealisasikan data atau *user* yang akan berhubungan satu dengan yang lain karena setiap data pasti memiliki hubungan karena data yang digunakan saling bersangkutan antara satu dengan yang lainnya.

6) Proses Pembuatan Sistem (Menyusun Source Code) (BackEnd)

Pada proses Back-End disini akan mengimplementasikan alur dari rancangan sistem dan juga rancangan dari *database* yang akan digunakan. di mana pada proses Back-end lebih mementingkan dalam menjaga validasi supaya keamanan aplikasi lebih terjamin.

7) Uji Coba Aplikasi (Software tester)

Pada proses Uji coba aplikasi yang dilakukan oleh software tester disini aplikasi tersebut akan melalui banyak tahap untuk diuji, seperti di uji sistem aplikasi tersebut dan juga apakah aplikasi tersebut aman serta layak untuk digunakan oleh *user*. Pada software tester ini dilakukan dengan metode pengujian yang berbeda-beda, tetapi disini kami menggunakan metode pengujian *Black Box*

testing dan UAT (*User Acceptence Test*).

8) Presentasi Hasil Final Sistem

Hasil dari semua proses yang telah dijalankan akan dipresentasikan kepada pihak klien supaya klien dapat mengetahui bagaimana proses yang telah dilalui dan bagaimana cara menggunakan aplikasi tersebut. Supaya tidak adakesalahan terkait penggunaan aplikasi yang sudah dibuat.

D. Biaya Proyek

Pengeluaran	Jumlah	Satuan	Harga	Jumlah
Hosting	2	Bulan	110.999	Rp. 222.000
Domain	1	Bulan	20.884	Rp. 23.000
Operasional	7	Orang	200.000	Rp. 1.400.000
Internet	7	Orang	50.000	Rp. 350.000
Total				Rp. 1.995.000

E. Jadwal Kegiatan

No	Nama Kegiatan	Minggu Pelaksanaan															
		Mei				Juni				Juli				Agustus			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Penerjunan																
2	Analisis kebutuhan <i>user</i> , analisa kebutuhan data, wawancara klien																
3	Perancangan kebutuhan sistem																
4	Perancangan desain <i>UI/UX</i>																
5	Proses slicing desain <i>UI/UX</i> ke dalam kodingan (Front End)																

6	Proses integrasi hasil Front End dengan <i>database</i> dan <i>coding</i> (Back End)																	
7	Pengujian sistem																	
8	Pemasangan hosting dan domain																	
9	Training mitra																	
10	Penyusunan laporan																	

F. Tim Pengembang

1. Mohammad Amirul Kurniawan Putranto : PM & Software Tester
2. Dinda Putri Dwi Yanti : Sekretaris & QAEng
3. Faza Husain Muhammad Zaen : Back-End Developer
4. Sindi Rella : *Data Analyst*
5. Zakiya Nurul Baity : Front-End Developer
6. Faqih 'Ainul Yaqin Nursaid : *Database Engineer*
7. Abdilah Muttaqi : *UI/UX / Bendahara*

G. Contact Person

Informasi lebih lanjut silahkan hubungi :

Nama : Dinda Putri Dwi Yanti
 Prodi/Fakultas : Informatika/Fakultas Teknologi Industri
 Universitas : Universitas Ahmad Dahlan
 Email : dinda2000018040@webmail.uad.ac.id
 No Hp : 081277150678

H. Penutup

Demikian proposal proyek ini kami buat berdasarkan kebutuhan yang sesuai dengan kondisi di lapangan. Semoga dengan adanya proposal ini membuat kerja sama dalam membangun proyek ini terwujud dengan baik. Mampu melaksanakan serta merealisasikan proyek pembuatan Proposal sistem keuangan Lazismu berbasis *website* dengan semaksimal mungkin. Besar harapan kami kiranya proposal kami ini dapat disetujui oleh pihak Lazismu, Atas segala perhatiannya kami ucapkan terimakasih.

- ii. Surat Kontrak/MoU MPTI dengan Mitra/Klien

**NOTA KESEPAHAMAN BERSAMA
ANTARA
LOVLACE
DENGAN
LAZISMU (LEMBAGA AMIL, ZAKAT, INFAK DAN SHADAQAH)
TENTANG
PEMBUATAN *WEBSITE* FINANSIAL LAZISMU**

Pada hari ini tanggal 17 Mei 2023 pihak-pihak yang bertanda tangan di bawah ini:

1. Lovlace, sebuah kelompok yang terdaftar sebagai salah satu perwakilan dari program studi Informatika Fakultas Teknologi Industri Universitas Ahmad Dahlan, dalam hal ini diwakili oleh Mohammad Amirul Kurniawan Putranto sebagai *Project manager*/Software Tester yang bertindak dalam jabatannya untuk dan atas nama Lovlace, untuk selanjutnya disebut sebagai **PIHAK PERTAMA**.
2. Lazismu, sebagai badan amil zakat dan infak shadaqah dalam hal ini diwakili oleh Marzuki, S.E sebagai Manajer Lazismu yang beralamat Rejowinangun, Kec. Kotagede, Kota Yogyakarta, Indonesia untuk selanjutnya disebut sebagai **PIHAK KEDUA**.

PIHAK PERTAMA dan **PIHAK KEDUA** untuk selanjutnya masing-masing disebut sebagai **PIHAK** dan secara bersama-sama disebut sebagai **PARA PIHAK**.

PARA PIHAK sepakat untuk mengadakan Nota Kesepahaman Bersama dengan ketentuan sebagai berikut :

**BAB I
PENGERTIAN-PENGERTIAN
Pasal 1**

1. **PIHAK PERTAMA** bertugas untuk merancang serta menganalisa sistem *website* yang sebelumnya telah disepakati pada proposal.
2. **PIHAK PERTAMA** dalam namanya yang disebutkan di atas dan apabila diperlukan di kemudian hari untuk melakukan perwakilan terhadap perjanjian ini, maka **PIHAK PERTAMA** akan menunjuk seseorang atas nama **PIHAK PERTAMA** sebagai petugas perwakilan **PIHAK PERTAMA**.
3. **PIHAK KEDUA** adalah konsumen atau pelanggan yang berbentuk perorangan atau badan instansi yang menghendaki sistem *website* company profile tersebut untuk dipergunakan pada masa sekarang.

**BAB II
HAK DAN KEWAJIBAN**

Pasal 2

1. PIHAK PERTAMA

1. Hak :

- Menerima Informasi atau data pendukung sebagai bahan dasar pembuatan sistem.
- Mendapatkan pendapatan sesuai dengan nominal yang telah disepakati yang aturannya disebutkan di dalam Project Proposal.

b. Kewajiban :

- Merancang serta menyiapkan sistem yang dibutuhkan oleh PIHAK KEDUA.
- Memberikan tata cara dan sistem yang diterapkan di *website* tersebut dengan kenyataan yang ada dan apabila disepakati, maka tata caranya ada di dalam project proposal.
- Memberikan pelayanan project sistem yang baik kepada PIHAK KEDUA selama 3 bulan sebelum peluncuran software.
- Membantu pelaksanaan pengelola *website* sistem Informasi yang baik kepada PIHAK KEDUA agar dapat berjalan lancar selama 3 bulan sebelum peluncuran software.
- Memantau perkembangan *website* sistem informasi yang telah dirancang selama 3 bulan sebelum peluncuran software.

2. PIHAK KEDUA

a. Hak :

- Mendapatkan satu paket sistem *website* yang telah teruji serta stabil untuk dipergunakan sebagai media pengolahan data dalam lingkup perusahaan yang bersangkutan setelah semua kewajibannya terpenuhi.
- Mendapatkan pelayanan sebaik-baiknya untuk cara pengoperasian sistem tersebut sesuai dengan ketentuan yang terlampir dalam Project Proposal.
- Mendapatkan pelayanan sebaik-baiknya untuk penanganan sistem error yang diakibatkan karena kesalahan kode/script dari Program PIHAK PERTAMA sesuai dengan ketentuan yang terlampir dalam Project Proposal.
- Mendapatkan bukti pembayaran dari PIHAK PERTAMA

b. Kewajiban :

- Menyediakan perangkat hardware (Komputer dan peripheral lainnya) secara lengkap dan berfungsi baik sebagai pendukung pengolahan sistem.
- Memberikan informasi secara lengkap tentang permintaannya jika PIHAK KEDUA ingin menambahkan fasilitas yang

dikehendaknya dengan tata cara yang telah disusun di Project Proposal

- Menaati peraturan-peraturan sebagai konsumen.
- Membayar biaya seperti tertera di Project Proposal.
- Melaporkan secepatnya kepada PIHAK PERTAMA bila terjadi error secara program atau terdapat proses-proses yang tidak dimengerti dalam sistem.
- Mengikuti tata tertib secara penggunaan sistem dan administrative untuk mendukung kelancaran sistem.
- Memperoleh garansi layanan service terhadap sistem selama 3 bulan dari setelah sistem digunakan.

BAB IV JANGKA WAKTU Pasal 4

Perjanjian kerja sama ini berlaku untuk jangka waktu 14 minggu terhitung sejak ditandatangani oleh PARA PIHAK dan dapat diperpanjang atau diakhiri atas persetujuan PARA PIHAK melalui pemberitahuan tertulis dari salah satu pihak kepada pihak lainnya paling lambat 7 minggu sebelum berakhirnya perjanjian.

BAB V HUKUM YANG BERLAKU DAN PENYELESAIAN PERSELISIHAN Pasal 5

1. Semua perbedaan pendapat dan atau sengketa yang timbul dalam pelaksanaan kerja sama ini akan diselesaikan oleh PARA PIHAK secara musyawarah untuk mufakat.
2. Nota Kesepahaman ini akan ditafsirkan dan diatur oleh hukum Negara Republik Indonesia
3. Apabila terjadi perselisihan dan perbedaan pendapat antara kedua belah pihak, maka masalah tersebut akan diselesaikan melalui musyawarah mufakat sebelum menempuh jalur hukum sejauh diperlukan pada saat bertempat tinggal di Kantor Pengadilan Negeri Jakarta atau di Kantor Pengadilan Negeri Yogyakarta.

BAB VI KETENTUAN LAIN-LAIN Pasal 6

1. Nota Kesepahaman ini tidak dapat dan tidak akan dialihkan kepada pihak ketiga tanpa persetujuan tertulis terlebih dahulu dari PARA PIHAK.
2. Kesepahaman Bersama ini dilaksanakan secara kelembagaan dengan itikad baik PARA PIHAK.
3. Segala perbedaan pendapat yang terjadi dalam pelaksanaan Kesepahaman bersama ini akan diselesaikan secara musyawarah dan mufakat.
4. Kondisi darurat yang berasal dari eksternal tim, seperti bencana alam atau kerusuhan maka PIHAK KEDUA secara sukarela memberikan bantuan berupa materi.

BAB VII
PENUTUP
Pasal 7

1. Nota Kesepahaman ini dibuat dalam rangkap 2 (dua), asli masing-masing bermaterai cukup serta mempunyai kekuatan hukum yang sama setelah ditandatangani oleh PARA PIHAK, untuk masing-masing pihak dan dipergunakan sebagaimana mestinya.
2. Kesepahaman Bersama ini dibuat dan ditandatangani oleh PARA PIHAK pada hari dan tanggal tersebut pada bagian awal Kesepahaman Bersama ini dibuat.

Yogyakarta, 17 Mei 2023

PIHAK PERTAMA
Lovlace



Mohammad Amirul Kurniawan Putranto
Project Manager
NIM 2000018040

PIHAK KEDUA
Lazismu



Manager
NBM : 1211275

iii. Log book

Log book kelompok

LOGBOOK MANAJEMEN PROYEK TEKNOLOGI INFORMASI MAHASISWA PROGRAM STUDI S1 INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A 2022/2023

(WAJIB DIISI DAN MASUK DALAM PENILAIAN)

Nama Kelompok MPTI : Lovlace		
No	NIM	Nama
1	2000018425	Mohammad Amirul Kurniawa Putranto
2	2000018040	Dinda Putri Dwi Yanti
3	2000018060	Faza Husain Muhammad Zaen
4	2000018156	Sindi Rella
5	2000018414	Zakiya Nurul Baity
6	2000018415	Faqih 'Ainul Yaqin N
7	2000018418	Abdilah muttaqin
Judul Manajemen Tugas Proyek : Proyek SPI-Lazismu		
Dosen Pembimbing : Murein Miksa Mardhia, S.T., M.T.		

Petunjuk Pengisian Log Book

Logbook dibuat per minggu

Logbook ditulis tangan

Setiap kegiatan di paraf oleh Mitra atau Klien/ dosen pembimbing MPTI

Logbook per minggu di paraf oleh dosen pengampu kelas MPTI

Jumlah bimbingan minimal 4x sebelum UTS

Logbook Minggu ke 4-7 (sebelum UTS)

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan (Jika ada)	Paraf MITRA/CLIENT	Paraf Dosen Pembimbing MPTI
		Hari/Tanggal	Durasi (Jam/hari)				
1.	Rapat Internal	16 Maret 2023	13.30 - 15.40	Pembahasan tentang dosen pembimbing dan job desc untuk tiap anggota			X [Signature]
2.	Meminta bu Murein untuk menjadi dosen pembimbing SPI	04 Mei 2023	08.23 - 09.15	Menyetujui sebagai dosen pembimbing			X [Signature]
3.	Bimbingan bersama dosen pembimbing ke-1	05 Mei 2023	13.30 - 15.00	Memperkenalkan anggota, job desc masing masing anggota dan memberikan arahan tentang program SPI			X [Signature]
4.	Penerjunan tim	08 Mei 2023	09.00-12.00	Penandatanganan kontrak kerjasama dengan Lazismu dan penjelasan sistem yang akan dikerjakan			X [Signature]

X
[Signature]

5.	Bimbingan bersama dosen pembimbing ke-2	09 Mei 2023	15.30 - 16.30	Pembahasan terkait sistem apa yang akan dipilih diantara			
----	---	-------------	---------------	--	--	--	--

				2 sistem yang diajukan oleh klien			
6.	Meeting bersama klien	16 Mei 2023	09.00 - 11.00	Pendalaman materi project, pengenalan fitur dan pembagian job desc tim			Xemp-
7.	Bimbingan bersama dosen pembimbing ke-3	16 Mei 2023	19.00-19.30	Laporan terkait pertemuan dengan Lazismu sebelumnya, memberikan saran mengenai <i>interface</i> , masukan mengenai pembagian job desc logbook, MOU.			Xemp-

8.	Diskusi bersama kelompok Stardust	17 Mei 2023	10.00 - 16.00	Membahas pembagian fitur mana saja yang dikerjakan masing-masing kelompok			X
9.	Rapat Internal	4 Juni 2023	10.00-15.30	Diskusi tentang <i>interface</i> yang telah dibuat, membuat IRD <i>database</i> , dan membuat implementasi kodingan tampilan awal			X
10.	Meeting bersama klien	10 Juni 2023	13.00 - 15.30	Diskusi tentang alur data keuangan yang akan digunakan untuk dijadikan <i>website</i>			X
11.	Ketemu Lazismu	04 Juli 2023	09.30	Ketemu Lazismu dengan tim satunya			X
12.	Meeting Internal	09 Juli 2023	13.00-18.30	Memulai mencil membuat laporan			X
13.	Meeting bersama klien	11 Juli 2023	13.00-17.00	Ketemu Lazismu dengan tim satunya dan Pak Herman			X
14.	Bimbingan bersama dosen pembimbing ke-3	17 Juli 2023	10.00-13.00	Progres dari sistem yang dibuat			X
15.	Meeting internal	11 Agustus 2023	16.00-18.00	Melakukan perubahan di kodingan karena ada perubahan di UI			X

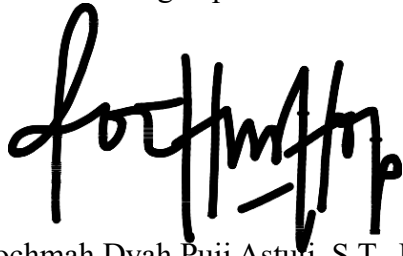
16.	Kerja kelompok dengan	17 Agustus 2023	15.00-17.00	Menambah fitur			<i>Yelp-</i>
-----	-----------------------	-----------------	-------------	----------------	--	--	--------------

	kelompok yang lain			<i>dropdown di user</i>			
17.	Meeting bersama klien	23 Agustus 2023	10.00-13.00	Melakukan evaluasi kekurangan dari sistem			<i>Yelp-</i>
18.	Meeting Internal	15 Agustus 2023	13.30-19.00	Menyelesaikan Laporan Akhir			<i>Yelp-</i>

Catatan Pembimbing Lapangan/Dosen Pembimbing MPTI/Dosen Pengampu Kelas MPTI :

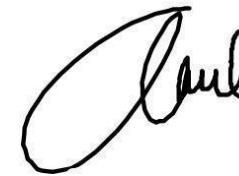
.....
.....
.....

Dosen Pengampu kelas MPTI



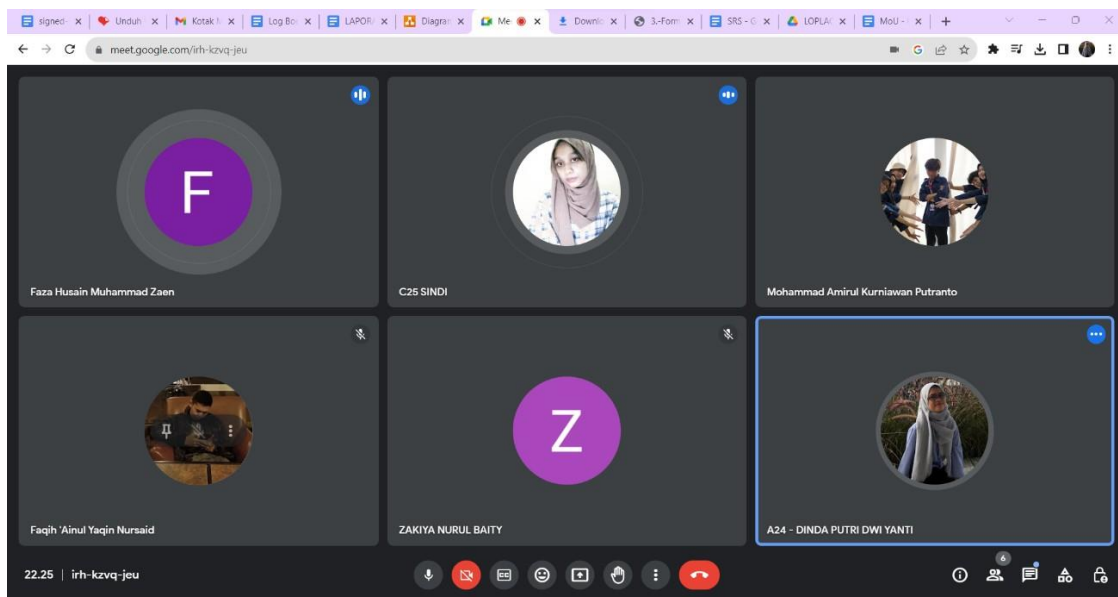
Nur Rochmah Dyah Puji Astuti, S.T., M.Kom.
NIP. 197608192005012001

Yogyakarta, 26 Agustus 2023
Ketua MPTI

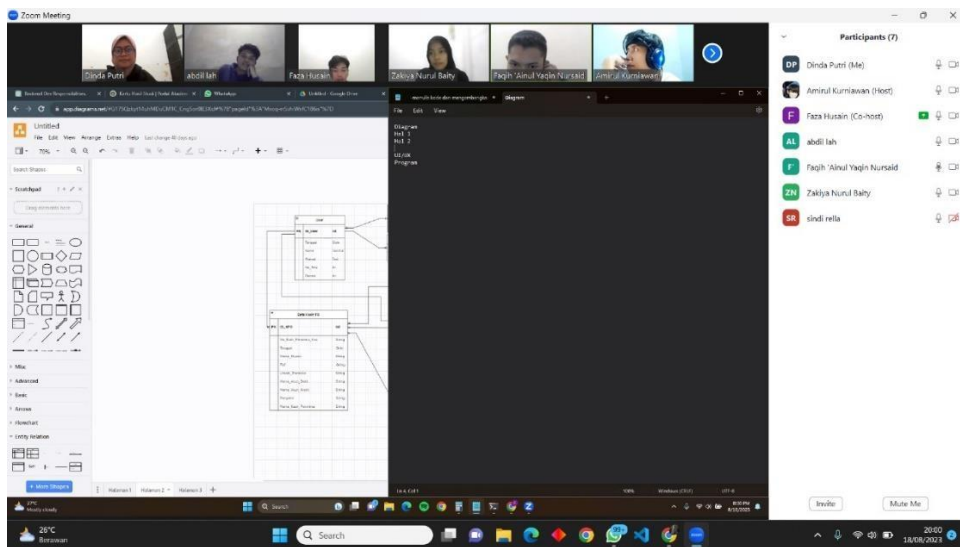


Mohammad Amirul Kurniawan Putranto
NIM. 2000018425

iv. Foto Dokumentasi Kegiatan







v. Bukti Pembiayaan

Pengeluaran



Invoice Biaya Pendidikan



Nomor IVR/20230825/XXIII/VIII/1735092286

Tanggal 25 Agu 2023 16:11

Nomor Pembayaran	2000018060
Total Tagihan	Rp51.500
ULTAHTOPED14	
Total Bayar	Rp51.500



Invoice Biaya Pendidikan



Nomor IVR/20230825/XXIII/VIII/1735426780

Tanggal 25 Agu 2023 21:28

Nomor Pembayaran	2000018415
Total Tagihan	Rp51.500
ULTAHTOPED14	
Total Bayar	Rp51.500



Invoice Biaya Pendidikan



Nomor IVR/20230825/XXIII/VIII/1735417617

Tanggal 25 Agu 2023 21:18

Nomor Pembayaran	2000018156
Total Tagihan	Rp51.500
ULTAHTOPED14	
Total Bayar	Rp51.500



Invoice Biaya Pendidikan

Invoice ini merupakan bukti pembayaran yang sah, dan diterbitkan atas nama Partner.



Nomor IVR/20230825/XXIII/VIII/1735411063

Tanggal 25 Agu 2023 21:11

Nomor Pembayaran	2000018425
Nama Institusi Pendidikan	Universitas Ahmad Dahlan
Nama	MOHAMMAD AMIRUL KURNIAWAN PUTR
Total Tagihan	Rp50.000
Biaya Admin	Rp1.500
	ULTAHTOPED14
Total Bayar	Rp51.500

Detail Tagihan

Tagihan ke-1

Jenis Tagihan	UAD-10 T. Proyek
Tagihan	Rp50.000
NIM	2000018425
Prodi	ANTOS1 Teknik Informatika



Invoice Biaya Pendidikan



Nomor IVR/20230825/XXIII/VIII/1735309429

Tanggal 25 Agu 2023 19:32

Nomor Pembayaran	2000018418
Total Tagihan	Rp51.500
Total Bayar	Rp51.500



Invoice Biaya Pendidikan



Nomor IVR/20230826/XXIII/VIII/1735671098

Tanggal 26 Agu 2023 07:38

Nomor Pembayaran	2000018040
Total Tagihan	Rp51.500
ULTAHTOPED14	
Total Bayar	Rp51.500



Invoice Biaya Pendidikan



Nomor IVR/20230825/XXIII/VIII/1735092286

Tanggal 25 Agu 2023 16:11

Nomor Pembayaran	2000018060
Total Tagihan	Rp51.500
ULTAHTOPED14	
Total Bayar	Rp51.500

vi. Tools

Source Code

index.js

Sourcecode 1 Index.js

```
import Vue from 'vue'
import Vuex from 'vuex'
import Meta from 'vue-meta'
import KlienOnly from 'vue-klien-only'
import NoSsr from 'vue-no-ssr'
import { createRouter } from './router.js'
import NuxtChild from './components/nuxt-child.js'
import NuxtError from './components/nuxt-error.vue'
import Nuxt from './components/nuxt.js'
import App from './App.js'
import { setContext, getLocation, getRouteData, normalizeError } from
'./utils'
import { createStore } from './store.js'

/* Plugins */

import nuxt_plugin_plugin_0461ebd3 from
'nuxt_plugin_plugin_0461ebd3' // Source: .\components\plugin.js
(mode: 'all')
import nuxt_plugin_vuescrollto_8afa1e5a from
'nuxt_plugin_vuescrollto_8afa1e5a' // Source: .\vue-scrollto.js (mode:
'klien')
import nuxt_plugin_axios_78537714 from
'nuxt_plugin_axios_78537714' // Source: .\axios.js (mode: 'all')
import nuxt_plugin_vClickOutside_6739adc2 from
'nuxt_plugin_vClickOutside_6739adc2' // Source:
..\plugins\vClickOutside (mode: 'klien')
import nuxt_plugin_auth_3b97fdfa from 'nuxt_plugin_auth_3b97fdfa'
// Source: .\auth.js (mode: 'all')

// Component: <KlienOnly>
Vue.component(KlienOnly.name, KlienOnly)

// TODO: Remove in Nuxt 3: <NoSsr>
Vue.component(NoSsr.name, {
  ...NoSsr,
  render (h, ctx) {
    if (process.klien && !NoSsr._warned) {
```

```

    NoSsr._warned = true

    console.warn('<no-ssr> has been deprecated and will be removed in
Nuxt 3, please use <klien-only> instead')
  }
  return NoSsr.render(h, ctx)
}
})

// Component: <NuxtChild>
Vue.component(NuxtChild.name, NuxtChild)
Vue.component('NChild', NuxtChild)

// Component NuxtLink is imported in server.js or klien.js

// Component: <Nuxt>
Vue.component(Nuxt.name, Nuxt)

Object.defineProperty(Vue.prototype, '$nuxt', {
  get() {
    const globalNuxt = this.$root ? this.$root.$options.$nuxt : null
    if (process.klien && !globalNuxt && typeof window !== 'undefined') {
      return window.$nuxt
    }
    return globalNuxt
  },
  configurable: true
})

Vue.use(Meta, {"keyName": "head", "attribute": "data-n-
head", "ssrAttribute": "data-n-head-ssr", "tagIDKeyName": "hid"})

const defaultTransition = {"name": "page", "mode": "out-
in", "appear": true, "appearClass": "appear", "appearActiveClass": "appear-
active", "appearToClass": "appear-to"}

const originalRegisterModule = Vuex.Store.prototype.registerModule

function registerModule (path, rawModule, options = {}) {
  const preserveState = process.klien && (
    Array.isArray(path)
    ? !!path.reduce((namespacedState, path) => namespacedState &&
namespacedState[path], this.state)

```

```

    : path in this.state
  )
  return originalRegisterModule.call(this, path, rawModule, {
    preserveState, ...options })
}

async function createApp(ssrContext, config = {}) {
  const store = createStore(ssrContext)
  const router = await createRouter(ssrContext, config, { store })

  // Add this.$router into store actions/mutations
  store.$router = router

  // Create Root instance

  // here we inject the router and store to all child components,
  // making them available everywhere as `this.$router` and
  `this.$store`.
  const app = {
    head: {"title": "Lazismu-
frontend", "htmlAttrs": {"lang": "en"}, "meta": [{"charset": "utf-
8"}, {"name": "viewport", "content": "width=device-width, initial-
scale=1"}, {"hid": "description", "name": "description", "content": ""}, {"na
me": "format-
detection", "content": "telephone=no"}], "link": [{"rel": "icon", "type": "ima
ge\u002F\x-ico", "href": "\u002Ffavicon.ico"}], "style": [], "script": []},

    store,
    router,
    nuxt: {
      defaultTransition,
      transitions: [defaultTransition],
      setTransitions (transitions) {
        if (!Array.isArray(transitions)) {
          transitions = [transitions]
        }
        transitions = transitions.map((transition) => {
          if (!transition) {
            transition = defaultTransition
          } else if (typeof transition === 'string') {
            transition = Object.assign({}, defaultTransition, { name: transition
          })
        }
      } else {

```

```

        transition = Object.assign({}, defaultTransition, transition)
      }
      return transition
    })
    this.$options.nuxt.transitions = transitions
    return transitions
  },

  err: null,
  dateErr: null,
  error (err) {
    err = err || null
    app.context._errored = Boolean(err)
    err = err ? normalizeError(err) : null
    let nuxt = app.nuxt // to work with @vue/composition-api, see
https://github.com/nuxt/nuxt.js/issues/6517#issuecomment-573280207
    if (this) {
      nuxt = this.nuxt || this.$options.nuxt
    }
    nuxt.dateErr = Date.now()
    nuxt.err = err
    // Used in src/server.js
    if (ssrContext) {
      ssrContext.nuxt.error = err
    }
    return err
  }
},
...App
}

// Make app available into store via this.app
store.app = app

const next = ssrContext ? ssrContext.next : location =>
app.router.push(location)
// Resolve route
let route
if (ssrContext) {
  route = router.resolve(ssrContext.url).route
} else {
  const path = getLocation(router.options.base, router.options.mode)

```

```

    route = router.resolve(path).route
  }

  // Set context to app.context
  await setContext(app, {
    store,
    route,
    next,
    error: app.nuxt.error.bind(app),
    payload: ssrContext ? ssrContext.payload : undefined,
    req: ssrContext ? ssrContext.req : undefined,
    res: ssrContext ? ssrContext.res : undefined,
    beforeRenderFns: ssrContext ? ssrContext.beforeRenderFns :
undefined,
    beforeSerializeFns: ssrContext ? ssrContext.beforeSerializeFns :
undefined,
    ssrContext
  })

  function inject(key, value) {
    if (!key) {
      throw new Error('inject(key, value) has no key provided')
    }
    if (value === undefined) {
      throw new Error(`inject('${key}', value) has no value provided`)
    }

    key = '$' + key
    // Add into app
    app[key] = value
    // Add into context
    if (!app.context[key]) {
      app.context[key] = value
    }

    // Add into store
    store[key] = app[key]

    // Check if plugin not already installed
    const installKey = '__nuxt_' + key + '_installed__'
    if (Vue[installKey]) {
      return
    }
  }

```

```

Vue[installKey] = true
// Call Vue.use() to install the plugin into vm
Vue.use(() => {
  if (!Object.prototype.hasOwnProperty.call(Vue.prototype, key)) {
    Object.defineProperty(Vue.prototype, key, {
      get () {
        return this.$root.$options[key]
      }
    })
  }
})
}

// Inject runtime config as $config
inject('config', config)

if (process.klien) {
  // Replace store state before plugins execution
  if (window.__NUXT__ && window.__NUXT__.state) {
    store.replaceState(window.__NUXT__.state)
  }
}

// Add enablePreview(previewData = {}) in context for plugins
if (process.static && process.klien) {
  app.context.enablePreview = function (previewData = {}) {
    app.previewData = Object.assign({}, previewData)
    inject('preview', previewData)
  }
}
// Plugin execution

if (typeof nuxt_plugin_plugin_0461ebd3 === 'function') {
  await nuxt_plugin_plugin_0461ebd3(app.context, inject)
}

if (process.klien && typeof nuxt_plugin_vuescrollto_8afa1e5a ===
'function') {
  await nuxt_plugin_vuescrollto_8afa1e5a(app.context, inject)
}

if (typeof nuxt_plugin_axios_78537714 === 'function') {
  await nuxt_plugin_axios_78537714(app.context, inject)
}

```



```

    }

    if (process.klien && typeof nuxt_plugin_vClickOutside_6739adc2 ===
'function') {
      await nuxt_plugin_vClickOutside_6739adc2(app.context, inject)
    }

    if (typeof nuxt_plugin_auth_3b97fdfa === 'function') {
      await nuxt_plugin_auth_3b97fdfa(app.context, inject)
    }

    // Lock enablePreview in context
    if (process.static && process.klien) {
      app.context.enablePreview = function () {
        console.warn('You cannot call enablePreview() outside a plugin.')
      }
    }

    // Wait for async component to be resolved first
    await new Promise((resolve, reject) => {
      // Ignore 404s rather than blindly replacing URL in browser
      if (process.klien) {
        const { route } = router.resolve(app.context.route.fullPath)
        if (!route.matched.length) {
          return resolve()
        }
      }
    })
    router.replace(app.context.route.fullPath, resolve, (err) => {
      // https://github.com/vuejs/vue-router/blob/v3.4.3/src/util/errors.js
      if (!err._isRouter) return reject(err)
      if (err.type !== 2 /* NavigationFailureType.redirected */) return
      resolve()

      // navigated to a different route in router guard
      const unregister = router.afterEach(async (to, from) => {
        if (process.server && ssrContext && ssrContext.url) {
          ssrContext.url = to.fullPath
        }
      })
      app.context.route = await getRouteData(to)
      app.context.params = to.params || {}
      app.context.query = to.query || {}
      unregister()
      resolve()
    })
  }
}

```

```

    })
  })
})

return {
  store,
  app,
  router
}
}

export { createApp, NuxtError }

```

sidebar.vue

Sourcecode 2 sidebar.vue

```

<template>
  <div>
    <div
      class="fixed lg:max-w-[295px] w-full overflow-y-auto h-full bg-
white z-[999]"
      :class="sidebar ? 'block' : 'hidden'"
      id="sidebarHRIS"
    >
      <div class="px-6 py-[50px] gap-y-[50px] flex flex-col">
        <div class="flex items-center justify-between">
          <a href="#" class="flex justify-center">
            
          </a>
          <a
            href="#"
            id="toggleCloseSidebar"
            class="lg:hidden"
            @click="toggleSidebar"
          >
            <svg
              class="w-6 h-6 text-dark"
              fill="none"
              stroke="currentColor"
              viewBox="0 0 24 24"
              xmlns="http://www.w3.org/2000/svg"
            >
              <path
                stroke-linecap="round"

```

```

        stroke-linejoin="round"
        stroke-width="2"
        d="M6 18L18 6M6 6l12 12"
      ></path>
    </svg>
  </a>
</div>
<div class="flex flex-col gap-4">
  <div class="text-sm text-grey">Menu utama</div>
  <NuxtLink
    v-if="role === 1 || 2 || 3"
    :to="{ name: 'cabang-id-overview', params: { id: cabang_id
  }}"
    class="nav-link"
  >
    
    Beranda
  </NuxtLink>
  <NuxtLink
    v-if="role !== 2"
    :to="{ name: 'cabang-id-CoA', params: { id: cabang_id } }"
    class="nav-link"
  >
    
    COA
  </NuxtLink>
  <NuxtLink
    v-if="role === 1 || 2 || 3"
    :to="{ name: 'cabang-id-frontoffices', params: { id:
cabang_id } }"
    class="nav-link"
  >
    
    Front Offices
  </NuxtLink>
  <NuxtLink
    v-if="role !== 2"
    :to="{ name: 'cabang-id-kasbesar', params: { id: cabang_id }
  }}"
    class="nav-link"
  >
    
    Kas Besar

```

```

</NuxtLink>
<NuxtLink
  v-if="role !== 2"
  :to="{ name: 'cabang-id-kaskecil', params: { id: cabang_id }
}"
  class="nav-link"
>
  
  Kas Kecil
</NuxtLink>
<NuxtLink
  v-if="role !== 2"
  :to="{ name: 'cabang-id-kasbank', params: { id: cabang_id }
}"
  class="nav-link"
>
  
  Kas Bank
</NuxtLink>
<NuxtLink
  v-if="role !== 2"
  :to="{ name: 'cabang-id-jurnalumum', params: { id:
cabang_id } }"
  class="nav-link"
>
  
  Jurnal Umum
</NuxtLink>
</div>
<div class="flex flex-col gap-4">
  <div class="text-sm text-grey">Others</div>
  <NuxtLink
    v-if="role === 1"
    :to="{ name: 'cabang-id-cabang', params: { id: cabang_id }
}"
    class="nav-link"
  >
    
    Cabang
  </NuxtLink>
  <NuxtLink
    v-if="role === 1"
    :to="{ name: 'cabang-id-user', params: { id: cabang_id } }"

```

```

        class="nav-link"
      >
        
        User
      </NuxtLink>

      <NuxtLink
        v-if="role === 1"
        :to="{ name: 'cabang-id-muzakki', params: { id: cabang_id }
    }"
        class="nav-link"
      >
        
        Muzakki
      </NuxtLink>

      <button @click="Logout" class="nav-link">
        
        Logout
      </button>
    </div>
  </div>
</div>
</template>

<script>
import { mapActions } from 'vuex'
export default {
  data() {
    return {
      role: JSON.parse(localStorage.getItem('role')),
      cabang_id: JSON.parse(localStorage.getItem('cabang_id')),
    }
  },
  computed: {
    sidebar() {
      return this.$store.state.sidebar
    },
  },
  methods: {
    ...mapActions(['openSidebar']),
    ...mapActions(['toggleSidebar']),
  },
}

```

```

    async Logout() {
      localStorage.removeItem('cabang_id')
      localStorage.removeItem('role')
      await this.$auth.Logout()
    },
  },
}
</script>

<style>
a.nuxt-link-active {
  @apply bg-Lazismu rounded-2xl text-white font-semibold;
}

a.nuxt-link-exact-active {
  @apply p-[11px] flex items-center gap-4;
}
.nuxt-link-active img {
  @apply brightness-0 invert;
}
</style>

```

footer.js

Sourcecode 3 Footer.js

```

<template>
  <nav
    class="flex justify-between items-center pt-4"
    aria-label="Tabel navigation"
  >
    <span class="text-sm font-normal text-gray-500"
      >Showing
      <span class="font-semibold text-gray-900">1 - {{ limit
    }}</span>
    of
    <span class="font-semibold text-gray-900">{{ total-
  rows}}</span></span>
  >
  <ul class="inline-flex items-center -space-x-px">
    <li>
      <a
        href="#"

```

```

class="relative block py-2 px-3 ml-0 leading-tight text-gray-
500 bg-white rounded-l-lg border border-gray-300 hover:bg-gray-
100 hover:text-gray-700"
>
<span class="sr-only">Previous</span>
<svg
class="w-5 h-5"
aria-hidden="true"
fill="currentColor"
viewBox="0 0 20 20"
xmlns="http://www.w3.org/2000/svg"
>
<path
fill-rule="evenodd"
d="M12.707 5.293a1 1 0 010 1.414L9.414 10l3.293
3.293a1 1 0 01-1.414 1.414l-4-4a1 1 0 010-1.414l-4-4a1 1 0
011.414 0z"
clip-rule="evenodd"
></path>
</svg>
</a>
</li>
<li>
<a
href="#"
class="py-2 px-3 text-sm leading-tight text-gray-500 bg-white
border border-gray-300 hover:bg-gray-100 hover:text-gray-700"
>1</a>
>
</li>

<li>
<a
href="#"
class="block py-2 px-3 leading-tight text-gray-500 bg-white
rounded-r-lg border border-gray-300 hover:bg-gray-100
hover:text-gray-700"
>
<span class="sr-only">Next</span>
<svg
class="w-5 h-5"
aria-hidden="true"
fill="currentColor"

```

```

        viewBox="0 0 20 20"
        xmlns="http://www.w3.org/2000/svg"
      >
        <path
          fill-rule="evenodd"
          d="M7.293 14.707a1 1 0 010-1.414L10.586 10 7.293
6.707a1 1 0 011.414-1.414l4 4a1 1 0 010 1.414l-4 4a1 1 0 01-
1.414 0z"
          clip-rule="evenodd"
        ></path>
      </svg>
    </a>
  </li>
</ul>
</nav>
</template>

```

BaseDropdown.js

Sourcecode 4 BaseDropdown.js

```

<template>
  <component
    class="dropdown"
    :is="tag"
    :class="[
      { show: isOpen },
      { dropdown: direction === 'down' },
      { dropup: direction === 'up' },
    ]"
    aria-haspopup="true"
    :aria-expanded="isOpen"
    @click="toggleDropdown"
    v-click-outside="closeDropdown"
  >
    <slot name="title">
      <a
        class="dropdown-toggle nav-link"
        :class="{ 'no-caret': hideArrow }"
        data-toggle="dropdown"
      >
        <i :class="icon"></i>

```



```

    <span class="no-icon">{{ title }}</span>
  </a>
</slot>
<ul
  class="dropdown-menu"
  ref="menu"
  :class="[
    { 'dropdown-menu-right': position === 'right' },
    { show: isOpen },
    menuClasses,
  ]"
  >
  <slot></slot>
</ul>
</component>
</template>
<script>
export default {
  name: "base-dropdown",
  props: {
    direction: {
      type: String,
      default: "down",
    },
    title: {
      type: String,
      description: "Dropdown title",
    },
    icon: {
      type: String,
      description: "Icon for dropdown title",
    },
    position: {
      type: String,
      description: "Position of dropdown menu (e.g right|left)",
    },
    menuClasses: {
      type: [String, Object],
      description: "Dropdown menu classes",
    },
    hideArrow: {
      type: Boolean,
      description: "Whether dropdown arrow should be hidden",
    },
  },
};

```

```

    },
    appendToBody: {
      type: Boolean,
      default: true,
      description: "Whether dropdown should be appended to
document body",
    },
    tag: {
      type: String,
      default: "li",
      description: "Dropdown html tag (e.g div, li etc)",
    },
  },
  data() {
    return {
      isOpen: false,
    };
  },
  methods: {
    toggleDropdown() {
      this.isOpen = !this.isOpen;
      this.$emit("change", this.isOpen);
    },
    closeDropdown() {
      this.isOpen = false;
      this.$emit("change", this.isOpen);
    },
  },
};
</script>
<style>
.dropdown {
  list-style-type: none;
}
.dropdown .dropdown-toggle {
  cursor: pointer;
}
</style>

```

Login.vue

Sourcecode 5 Login.vue

```
<template>
```

```

<section class="py-[50px] flex flex-col items-center justify-center
px-4">
  
  <div class="text-[32px] font-semibold text-dark mt-[70px]">Sign
In</div>
  <p class="mt-4 text-base leading-7 text-center mb-[50px] text-
grey">
    Kelola keuangan dengan sistem yang baru <br />
    semoga lebih membantu
  </p>
  <div
    v-if="validation != null"
    class="mb-3 bg-red-100 border border-red-400 text-red-700 px-
4 py-3 rounded relative"
    role="alert"
  >
    <strong class="font-bold">Error</strong>
    <span class="block sm:inline">{{ validation }}</span>
    <span class="absolute top-0 bottom-0 right-0 px-4 py-3">
</span>
</div>
<form class="w-full card" @submit.prevent="userLogin">
  <div class="form-group">
    <label for="" class="text-grey">Email Address</label>
    <input type="email" class="input-field" v-model="Login.email"
/>
  </div>
  <div class="form-group">
    <label for="" class="text-grey">Password</label>
    <input
      :type="show === true ? 'text' : 'password'"
      class="input-field"
      v-model="Login.password"
    />
  <svg
    v-if="!show"
    @click="show = !show"
    xmlns="http://www.w3.org/2000/svg"
    fill="none"
    viewBox="0 0 24 24"
    stroke-width="1.5"
    stroke="currentColor"
    class="show-icon w-6 h-6"
  >

```

```

>
  <path
    stroke-linecap="round"
    stroke-linejoin="round"
    d="M2.036 12.322a1.012 1.012 0 010-.639C3.423 7.51 7.36
4.5 12 4.5c4.638 0 8.573 3.007 9.963 7.178.07.207.07.431 0
.639C20.577 16.49 16.64 19.5 12 19.5c-4.638 0-8.573-3.007-9.963-
7.178z"
  />
  <path
    stroke-linecap="round"
    stroke-linejoin="round"
    d="M15 12a3 3 0 11-6 0 3 3 0 016 0z"
  />
</svg>

<svg
  v-else
  @click="show = !show"
  xmlns="http://www.w3.org/2000/svg"
  fill="none"
  viewBox="0 0 24 24"
  stroke-width="1.5"
  stroke="currentColor"
  class="hide-icon w-6 h-6"
>
  <path
    stroke-linecap="round"
    stroke-linejoin="round"
    d="M3.98 8.223A10.477 10.477 0 001.934 12C3.226 16.338
7.244 19.5 12 19.5c.993 0 1.953-.138 2.863-.395M6.228
6.228A10.45 10.45 0 0112 4.5c4.756 0 8.773 3.162 10.065
7.498a10.523 10.523 0 01-4.293 5.774M6.228 6.228L3 3m3.228
3.228l3.65 3.65m7.894 7.894L21 21m-3.228-3.228l-3.65-3.65m0
0a3 3 0 10-4.243-4.243m4.242 4.242L9.88 9.88"
  />
</svg>
</div>

<button type="submit" class="w-full btn btn-Lazismu mt-
[14px]">
  Sign In
</button>

```

```

    </form>
  </section>
</template>

<script>
export default {
  auth: 'guest',

  data() {
    return {
      show: "",
      Login: {
        email: "",
        password: "",
      },
      validation: null,
    }
  },
  methods: {
    async userLogin() {
      try {
        await this.$auth
          .LoginWith('local', { data: this.Login })
          .then((response) => {
            localStorage.setItem(
              'role',
              JSON.stringify(response.data.result.user.role)
            )
            localStorage.setItem(
              'cabang_id',
              JSON.stringify(response.data.result.user.cabang_id)
            )

            this.$router.push({
              name: 'cabang-id-overview',
              params: {
                id: response.data.result.user.cabang_id,
              },
            })
          })
        console.log(response)
      } catch (err) {
        //assign validation
      }
    }
  }
}

```

```
        this.validation = err.response.data.meta.message
        console.log(err)
      }
    },
  },
}
</script>

<style>
.show-icon {
  position: absolute;
  right: 35px;
  top: 162px;
  z-index: 11;
  cursor: pointer;
}
.hide-icon {
  position: absolute;
  right: 35px;
  top: 160px;
  z-index: 11;
  cursor: pointer;
}
</style>
```

indexcabang.vue

Sourcecode 6 Indexcabang.vue

```
<template>
  <div
    class="lg:pr-[70px] py-[50px] lg:ml-[320px] px-4 lg:pl-0"
    :class="sidebar ? 'xl:ml-[365px]' : 'xl:ml-[65px]'"
  >
    <!-- Top Section -->
    <Modal
      :onClick="deletePost"
      v-show="modal"
      :toggleModal="modal"
      :close="closeModal"
      :wrongaction="hapus"
    >Hapus akun
    <b class="text-red-700"
```

```

        ><i>{{ msg }}</i></b>
    ></Modal>
>
<!-- Top Section -->

<section
  class="flex flex-col flex-wrap justify-between gap-6 md:items-
center md:flex-row"
>
  >
  <div class="flex items-center justify-between gap-4">
    <a href="#" id="toggleOpenSidebar" @click="toggleSidebar">
      <svg
        class="w-6 h-6 text-dark"
        fill="none"
        stroke="currentColor"
        viewBox="0 0 24 24"
        xmlns="http://www.w3.org/2000/svg"
      >
        <path
          stroke-linecap="round"
          stroke-linejoin="round"
          stroke-width="2"
          d="M4 6h16M4 12h16M4 18h7"
        ></path>
      </svg>
    </a>
    <div class="text-[32px] font-semibold text-
dark">Cabang</div>
  </div>
  <div class="text-[16px] flex">
    
    <p v-if="role === 1" class="">Administrator</p>
    <p v-if="role === 2" class="">Front Office</p>
    <p v-if="role === 3" class="">Back Office</p>
  </div>
</section>

<section class="pt-[50px]">
  <!-- Section Header -->
  <div class="mb-[30px]">
    <div

```

```

class="flex flex-col justify-between gap-6 sm:items-center
sm:flex-row"
  >
  <div>
    <div class="text-xl font-medium text-dark">Statistik</div>
  </div>
</div>

<p v-if="$fetchState.pending">Data Loading</p>
<div v-else class="grid gap-6 sm:grid-cols-2 md:grid-cols-3
lg:gap-11">
  <div class="card !gap-y-10">
    <div class="flex items-center">
      <div>
        <p class="text-grey">Jumlah Cabang</p>

        <div class="text-[32px] font-bold text-dark mt-[6px]">
          {{ cabang.data.result.total }}
        </div>
      </div>
    </div>
  </div>
</div>
</section>

<section class="pt-[50px]">
  <!-- Section Header -->
  <div class="mb-[30px]">
    <div class="flex items-center justify-between gap-6">
      <div>
        <div class="text-xl font-medium text-dark">Cabang</div>
        <p class="text-grey">Daftar Cabang</p>
      </div>
    </div>
  </div>

  <div
    class="overflow-x-auto relative shadow-md sm:rounded-lg p-
2 bg-white"
  >
    <div class="pb-4 bg-white flex justify-between items-center">
      <label for="tabel-search" class="sr-only">Search</label>

```



```

<div class="relative mt-1">
  <div
    class="flex absolute inset-y-0 left-0 items-center pl-3
pointer-events-none"
  >
    <svg
      class="w-5 h-5 text-gray-500 dark:text-gray-400"
      aria-hidden="true"
      fill="currentColor"
      viewBox="0 0 20 20"
      xmlns="http://www.w3.org/2000/svg"
    >
      <path
        fill-rule="evenodd"
        d="M8 4a4 4 0 100 8 4 4 0 00-8z"
        clip-rule="evenodd"
      ></path>
    </svg>
  </div>
  <input
    type="text"
    id="tabel-search"
    class="block p-2 pl-10 w-80 text-sm text-gray-900 bg-gray-
50 rounded-lg border border-gray-300 focus:ring-blue-500
focus:border-blue-500"
    placeholder="Cari nama cabang"
    v-model="keywords"
  />
</div>
<NuxtLink
  :to="{
    name: 'cabang-id-cabang-create',
    params: { id: cabang_id },
  }"
  class="btn-sm px-4 py-2 rounded-lg bg-oranye-400 text-sm
text-white"
>
  Tambah
</NuxtLink>
</div>

```

```

<tabel
  class="w-full text-sm text-left text-gray-200 hover:border-
collapse"
  >
  <thead class="text-xs text-white uppercase bg-oranye-400">
  <tr>
  <th scope="col" class="p-2">
  <div class="flex items-center">No.</div>
  </th>
  <th scope="col" class="py-3 px-10">Nama Cabang</th>
  <th scope="col" class="py-3 px-6">Id</th>
  <th scope="col" class="py-3 px-6">
  <div class="">Aksi</div>
  </th>
  </tr>
  </thead>
  <tbody>
  <p v-if="$fetchState.pending">Fetching roles...</p>
  <tr
  class="bg-white border-b text-gray-700"
  v-else
  v-for="(item, index) in cabang.data.result.data"
  :key="index"
  >
  <td class="p-4 w-4">
  <div class="flex items-center">
  {{ cabang.data.result.current_page * 10 - 10 + index + 1
}}
  </div>
  </td>
  <td scope="row" class="py-4 px-6 font-medium text-gray-
900">
  {{ item.name }}
  </td>
  <td class="py-4 px-6">{{ item.id }}</td>
  <td class="py-4 px-6">
  <button
  id="dropdownMenuIconButton"
  data-dropdown-toggle="dropdownDots"
  class="inline-flex items-center p-2 text-sm font-medium
text-center text-gray-900 bg-white rounded-lg hover:bg-gray-100
focus:ring-4 focus:outline-none"
  type="button"

```

```

    @click="toggleDropdown(item.id)"
  >
    <svg
      class="w-6 h-6"
      aria-hidden="true"
      fill="currentColor"
      viewBox="0 0 20 20"
      xmlns="http://www.w3.org/2000/svg"
    >
      <path
        d="M10 6a2 2 0 110-4 2 2 0 010 4zM10 12a2 2 0 110-4 2 2 0 010 4zM10 18a2 2 0 110-4 2 2 0 010 4z"
      ></path>
    </svg>
  </button>

  <!-- Dropdown menu -->
  <div
    v-show="isHidden"
    id="dropdownDots"
    class="bg-white w-fit rounded divide-y divide-gray-100
shadow"
    :class="isHidden === item.id ? 'show' : 'hidden'"
  >
    <ul
      class="py-1 text-sm text-gray-700"
      aria-labelledby="dropdownMenuIconButton"
    >
      <li>
        <Nuxt-Link
          :to="{
            name: 'cabang-id-cabang-edit',
            params: { item: item.id },
          }"
          class="block py-2 px-2 w-10 hover:bg-gray-100"
        >
          Edit
        </Nuxt-Link>
      </li>
      <li>
        <button
          href="#"
          class="block py-2 px-2 hover:bg-gray-100"

```

```

        @click="openModal(item.id, index), (msg =
item.name)"
      >
        Delete
      </button>
    </li>
  </ul>
</div>
<!-- <div class="flex items-center">
  <NuxtLink :to="{name: ', params: { id: cabang_id }}"
class="bg-gray-100"
  tag="img"
  :src="require('~static/assets/svg/dot3.svg')" >

  </Nuxtlink>
</div> -->
</td>
</tr>
</tbody>
</table>
<p v-if="$fetchState.pending">Fetching roles...</p>

<nav
  v-else
  class="flex justify-between items-center pt-4"
  aria-label="Tabel navigation"
  >
  <span class="text-sm font-normal text-gray-500"
  >Showing
  <span class="font-semibold text-gray-900">1-10</span>
  of
  <span class="font-semibold text-gray-900">{{
  cabang.data.result.last_page
  }}</span></span>
  >
  <ul class="inline-flex items-center -space-x-px">
  <li>
    <button
      :value="cabang.data.result.current_page - 1"
      @click="updatePage"
      class="relative block py-2 px-3 ml-0 leading-tight text-
gray-500 bg-white rounded-l-lg border border-gray-300 hover:bg-
gray-100 hover:text-gray-700"

```

```

        :disabled="cabang.data.result.current_page == 1"
      >
        &lt;
      </button>
    </li>
    <p v-if="$fetchState.pending">Fetching roles...</p>

    <li
      v-else
      v-for="(ada, index) in cabang.data.result.links"
      :key="index"
    >
      <button
        v-if="
          ada.label != '&laquo; Sebelumnya' &&
          ada.label != 'Berikutnya &raquo;';
        "
        :value="index"
        @click="updatePage"
        class="py-2 px-3 leading-tight text-gray-500 bg-white
border border-gray-300 hover:bg-gray-100 hover:text-gray-700"
      >
        {{ ada.label }}
      </button>
    </li>

    <li>
      <button
        :value="nextUrl"
        @click="updatePage"
        class="block py-2 px-3 leading-tight text-gray-500 bg-
white rounded-r-lg border border-gray-300 hover:bg-gray-100
hover:text-gray-700"
        :disabled="pagenow == cabang.data.result.last_page"
      >
      >
    </button>
    </li>
  </ul>
</nav>
</div>
</section>
</div>

```

```

</template>

<script>
import Modal from '@/components/Modal.vue'
import { mapActions } from 'vuex'
export default {
  components: { Modal },
  layout: 'dashboard',
  middleware: 'auth',
  data() {
    return {
      cabang_id: JSON.parse(localStorage.getItem('cabang_id')),
      role: JSON.parse(localStorage.getItem('role')),
      nextUrl: null,
      prevUrl: null,
      keywords: null,
      pagenow: null,
      cabang: [],
      modal: false,
      hapus: 'Konfirmasi',
      isHidden: false,
      msg: "",
    }
  },
  computed: {
    sidebar() {
      return this.$store.state.sidebar
    },
  },
  watch: {
    keywords(after, before) {
      this.refetch()
      fetchOnServer: false
    },
  },
  async fetch() {
    this.cabang = await this.$axios.get('/cabang', {
      params: {
        name: this.keywords,
        limit: 10,
      },
    })
  },
}

```

```

methods: {
  ...mapActions(['toggleSidebar']),
  toggleDropdown(kode) {
    console.log('pencet log', this.isHidden)
    if (this.isHidden === false) {
      this.isHidden = kode
    } else {
      this.isHidden = false
    }
  },
  closeDropdown() {
    this.isHidden = false
  },
  openModal(id, index) {
    this.id = id
    this.index = index
    this.modal = !this.modal
  },
  closeModal() {
    this.modal = false
    this.modalimport = false
    this.modalExport = false
  },

  async refetch() {
    this.cabang = await this.$axios.get('/cabang', {
      params: {
        name: this.keywords,
        limit: 10,
        page: this.pagenow,
      },
    })
    // (this.jumlahCoA = await this.$axios.get('/hitung/CoA', {
    //   params: {
    //     cabang_id: this.cabang_id,
    //   },
    // }))
  },
  async deletePost() {
    //delete data post by ID
    await this.$axios.delete(`/cabang/${this.id}`).then(() => {
      this.id = null
    })
  }
}

```

```

        this.index = null
        this.modal = !this.modal
        this.refetch()
    })
  },

  updatePage(event) {
    this.pagenow = parseInt(event.target.value)
    this.$nuxt.refresh()
    this.nextUrl = 1 + parseInt(event.target.value)
  },
},
}
</script>

```

indexCoA.vue

Sourcecode 7 IndexCoA.vue

```

<template>
  <div
    class="lg:pr-[70px] py-[50px] lg:ml-[320px] px-4 lg:pl-0"
    :class="sidebar ? 'xl:ml-[365px]' : 'xl:ml-[65px]'"
  >
    <!-- Top Section -->
    <Modal
      :onClick="deletePost"
      v-show="modal"
      :toggleModal="modal"
      :close="closeModal"
      :wrongaction="hapus"
    >Hapus akun
    <b class="text-red-700"
      ><i>{{ msg }}</i>
    </b>
    ?</Modal
  >
  <Modal
    :onClick="exportExcel"
    v-show="modalExport"
    :toggleModal="modalExport"
    :close="closeModal"
    :action="upload"
  >Export dalam bentuk Excel ?</Modal
  >

```



```

<Modal
  :onClick="closeModal"
  v-show="modalimport"
  :toggleModal="modalimport"
  :close="closeModal"
  :action="upload"
>
  <input type="file" @change="onFilePicked" name="file" />
  <p class="text-red-500 mb-4">{{ messageimport }}</p>
</Modal>
<!-- Top Section -->

<section
  class="flex flex-col flex-wrap justify-between gap-6 md:items-
center md:flex-row"
>
  <div class="flex items-center justify-between gap-4">
    <a href="#" id="toggleOpenSidebar" @click="toggleSidebar">
      <svg
        class="w-6 h-6 text-dark"
        fill="none"
        stroke="currentColor"
        viewBox="0 0 24 24"
        xmlns="http://www.w3.org/2000/svg"
      >
        <path
          stroke-linecap="round"
          stroke-linejoin="round"
          stroke-width="2"
          d="M4 6h16M4 12h16M4 18h7"
        ></path>
      </svg>
    </a>
    <div class="text-[32px] font-semibold text-dark">
      Kode Akun Akutansi
    </div>
    <div class="text-[16px] flex">
      
      <p v-if="role === 1" class="">Administrator</p>
      <p v-if="role === 2" class="">Front Office</p>
      <p v-if="role === 3" class="">Back Office</p>

```

```

</div>
</section>

<section class="pt-[50px]">
  <!-- Section Header -->

  <div class="mb-[30px]">
    <div
      class="flex flex-col justify-between gap-6 sm:items-center
sm:flex-row"
    >
      <div>
        <div class="text-xl font-medium text-dark">Jumlah
Akun</div>
      </div>
    </div>
  </div>

  <p v-if="$fetchState.pending">Tunggu Sebentar ...</p>
  <div v-else class="gap-6 sm:grid-cols-2 md:grid-cols-2 lg:gap-
11">
    <div class="card !gap-y-10">
      <div class="flex items-center justify-between">
        <div>
          <p class="text-grey">CoA Debit</p>

          <div class="text-[32px] font-bold text-dark mt-[6px]">
            {{ jumlahCoA.data.CoAdebit }} Akun
          </div>
        </div>
        <div>
          <p class="text-grey">CoA Kredit</p>
          <div class="text-[32px] font-bold text-dark mt-[6px]">
            {{ jumlahCoA.data.CoAkredit }} Akun
          </div>
        </div>
      </div>
    </div>
  </div>
</section>

<section class="pt-[50px] mt-10">
  <!-- Section Header -->

```

```

<div class="mb-[30px]">
  <div class="flex items-center justify-between gap-6">
    <div>
      <div class="text-xl font-medium text-dark">
        Kode Akun (Chart of Account)
      </div>
      <p class="text-grey">Debit Kredit</p>
    </div>
    <div class="flex items-end">
      <a
        download=""
        @click="modalimport = !modalimport"
        class="button cursor-pointer py-2 px-4 bg-blue-400
rounded-lg text-white"
      >
        Import
      </a>
      <a
        download=""
        @click="modalExport = !modalExport"
        class="button cursor-pointer ml-1 py-2 px-4 bg-green-400
rounded-lg text-white"
      >
        Export to Excel
      </a>
    </div>
  </div>
</div>

<div class="md:container md:mx-auto">
  <div class="overflow-x-auto relative shadow-md sm:rounded-
lg bg-white">
    <table
      class="w-full text-sm text-left text-gray-200 hover:border-
collapse whitespace-nowrap"
    >
      <caption
        class="p-5 text-lg font-semibold text-left text-gray-900 bg-
white"
      >
        <div class="bg-white flex justify-between items-center">
          <div class="flex">

```

```

<label for="tabel-search" class="sr-
only">Search</label>
<div class="relative mt-1">
  <div
    class="flex absolute inset-y-0 left-0 items-center pl-3
pointer-events-none"
  >
    <svg
      class="w-5 h-5 text-gray-500 dark:text-gray-400"
      aria-hidden="true"
      fill="currentColor"
      viewBox="0 0 20 20"
      xmlns="http://www.w3.org/2000/svg"
    >
      <path
        fill-rule="evenodd"
        d="M8 4a4 4 0 100 8 4 4 0 000-8zM2 8a6 6 0
1110.89 3.476l4.817 4.817a1 1 0 01-1.414 1.414l-4.816-4.816A6 6
0 012 8z"
        clip-rule="evenodd"
      ></path>
    </svg>
    <input
      type="text"
      id="tabel-search"
      class="block p-2 pl-10 w-80 text-sm text-gray-900 bg-
gray-50 rounded-lg border border-gray-300 focus:ring-blue-500
focus:border-blue-500"
      placeholder="Cari nama akun"
      v-model="keywords"
    />
  </div>
  <div class="mt-1 ml-2">
    <select
      name="tipe"
      id=""
      class="bg-gray-50 border border-gray-300 text-gray-
500 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-
500 block w-full py-2 px-2.5"
      v-model="isKredit"
    >
      <option value="" selected>All</option>

```

```

        <option value="false">Debit</option>
        <option value="1">Kredit</option>
    </select>
</div>
</div>

<NuxtLink
  :to="{
    name: 'cabang-id-CoA-create',
    params: { id: cabang_id },
  }"
  class="btn-sm px-4 py-2 rounded-lg bg-oranye-400 text-
sm text-white"
  >
  Tambah
</NuxtLink>
</div>
</caption>

<thead class="text-xs text-white uppercase bg-oranye-
400">
<tr>
<th scope="col" class="p-2">
  <div class="flex items-center">No.</div>
</th>
<th scope="col" class="py-3 px-6">Nama Akun</th>
<th scope="col" class="py-3 px-6">Nomer Urut</th>
<th scope="col" class="py-3 px-6">Tipe</th>
<th scope="col" class="py-3 px-6">Front Office</th>
<th scope="col" class="py-3 px-6">Aksi</th>
</tr>
</thead>
<tbody>
<p v-if="$fetchState.pending">Fetching roles...</p>
<tr
  class="bg-white border-b text-gray-700"
  v-else
  v-for="(item, index) in CoA.data.result.data"
  :key="index"
  >
  <td class="py-4 px-6 font-medium">
    {{ CoA.data.result.current_page * 10 - 10 + index + 1 }}
  </td>

```

```

        <td class="py-4 px-6">
            {{ item.name }}
        </td>
        <td class="py-4 px-6">{{ item.kode }}</td>
        <td class="py-4 px-6">{{ item.tipe ? 'kredit' : 'debit'
    }}</td>
        <td class="py-4 px-6">{{ item.isShow ? 'Ya' : 'Tidak'
    }}</td>
        <td class="py-4 px-6">
            <button
                id="dropdownMenuIconButton"
                data-dropdown-toggle="dropdownDots"
                class="inline-flex items-center p-2 text-sm font-
                medium text-center text-gray-900 bg-white rounded-lg hover:bg-
                gray-100 focus:ring-4 focus:outline-none"
                type="button"
                @click="toggleDropdown(item.kode)"
            >
                <svg
                    class="w-6 h-6"
                    aria-hidden="true"
                    fill="currentColor"
                    viewBox="0 0 20 20"
                    xmlns="http://www.w3.org/2000/svg"
                >
                    <path
                        d="M10 6a2 2 0 110-4 2 2 0 010 4zM10 12a2 2 0
                    110-4 2 2 0 010 4zM10 18a2 2 0 110-4 2 2 0 010 4"
                    ></path>
                </svg>
            </button>

            <!-- Dropdown menu -->
            <div
                v-show="isHidden"
                refs="drop"
                id="dropdownDots"
                class="bg-white rounded divide-y divide-gray-100
                shadow absolute"
                :class="isHidden === item.kode ? 'show' : 'hidden'"
            >
                <ul
                    class="py-1 text-sm text-gray-700"

```

```

        aria-labelledby="dropdownMenuIconButton"
      >
      <li>
        <Nuxt-Link
          :to="{
            name: 'cabang-id-CoA-edit',
            params: { id: cabang_id, item: item.id },
          }"
          class="block py-2 px-4 hover:bg-gray-100"
        >Edit
        </Nuxt-Link>
      </li>
      <li>
        <button
          href="#"
          class="block py-2 px-4 hover:bg-gray-100"
          @click="openModal(item.id, index), (msg =
item.name)"
        >
          Delete
        </button>
      </li>
    </ul>
  </div>
  <!-- <div class="flex items-center">
    <NuxtLink :to="{name: ', params: { id: cabang_id } }"
class="bg-gray-100"
    tag="img"
    :src="require('~static/assets/svg/dot3.svg)" >

    </Nuxtlink>
  </div> -->
  </td>
</tr>
</tbody>
</table>
<p v-if="$fetchState.pending">Fetching</p>
<nav
  v-else
  class="flex justify-between items-center pt-4 p-2"
  aria-label="Tabel navigation"
  >
  <span class="text-sm font-normal text-gray-500"

```

```

>Showing
<span class="font-semibold text-gray-900">1-10</span>
of
<span class="font-semibold text-gray-900">{{
  CoA.data.result.last_page
}}</span></span>
>
<ul class="inline-flex items-center -space-x-px">
<li>
  <button
    :value="CoA.data.result.current_page - 1"
    @click="updatePage"
    class="block py-2 px-3 ml-0 leading-tight text-gray-500
    bg-white rounded-l-lg border border-gray-300 hover:bg-gray-100
    hover:text-gray-700"
    :disabled="CoA.data.result.current_page == 1"
  >
    &lt;
  </button>
</li>
<p v-if="$fetchState.pending">Fetching roles...</p>

<li
  v-else
  v-for="(ada, index) in CoA.data.result.links"
  :key="index"
>
  <button
    v-if="
      ada.label != '&laquo; Sebelumnya' &&
      ada.label != 'Berikutnya &raquo;'"
    :value="index"
    @click="updatePage"
    class="py-2 px-3 leading-tight text-gray-500 bg-white
    border border-gray-300 hover:bg-gray-100 hover:text-gray-700"
  >
    {{ ada.label }}
  </button>
</li>

<li>
  <button

```



```

        :value="nextUrl"
        @click="updatePage"
        class="block py-2 px-3 leading-tight text-gray-500 bg-
white rounded-r-lg border border-gray-300 hover:bg-gray-100
hover:text-gray-700"
        :disabled="pagenow == CoA.data.result.last_page"
    >
    >
</button>
</li>
</ul>
</nav>
</div>
</div>
</section>
</div>
</template>

```

```

<script>
import { mapActions } from 'vuex'
import Modal from '@components/Modal.vue'
export default {
  layout: 'dashboard',
  middleware: 'auth',

  data() {
    return {
      cabang_id: JSON.parse(localStorage.getItem('cabang_id')),
      role: JSON.parse(localStorage.getItem('role')),
      hapus: 'Konfirmasi',
      upload: 'Konfirmasi',
      modal: false,
      modalimport: false,
      modalExport: false,
      isHidden: false,
      isKredit: '',
      nextUrl: 2,
      prevUrl: null,
      keywords: null,
      pagenow: null,
      messageimport: null,
      kode: null,
      CoA: [],

```

```

    jumlahCoA: [],
    file: null,
    id: null,
    index: null,
  }
},
computed: {
  sidebar() {
    return this.$store.state.sidebar
  },
},

watch: {
  keywords(after, before) {
    this.refetch()
    fetchOnServer: false
  },

  isKredit() {
    this.$nuxt.refresh()
    console.log(this.isKredit)
  },
},
components: { Modal },

async fetch() {
  ;(this.CoA = await this.$axios.get('/CoA', {
    params: {
      name: this.keywords,
      limit: 10,
      page: this.pagenow,
      cabang_id: this.cabang_id,
      tipe: this.isKredit,
      isShow: this.isShow,
    },
  })),
  (this.jumlahCoA = await this.$axios.get('/hitung/CoA', {
    params: {
      cabang_id: this.cabang_id,
    },
  })))
},

```

```

methods: {
  ...mapActions(['toggleSidebar']),
  onFilePicked(e) {
    const formData = new FormData()
    const file = e.target.files[0]
    formData.append('file', file)
    console.log(formData)
    this.$axios
      .post('/CoA/import_excel', formData)
      .then((response) => {
        console.log(response)
        this.refetch()
      })
      .catch((error) => {
        console.log(error)
        this.modalimport = true
        this.messageimport = error.response.data.message
      })
  },

  async apa() {
    try {
      let response = await this.$axios.post('/CoA/import_excel',
      formData)

      //Redirect to my CoA page
      this.$router.push({ name: 'cabang-id-CoA' })

      console.log(response)
    } catch (error) {
      console.log(error)
    }
  },

  updatePage(event) {
    this.pagenow = parseInt(event.target.value)
    this.$nuxt.refresh()
    this.nextUrl = 1 + parseInt(event.target.value)
  },

  async refetch() {
    ;(this.CoA = await this.$axios.get('/CoA', {
      params: {
        name: this.keywords,
        limit: 10,

```

```

        page: this.pagenow,
        cabang_id: this.cabang_id,
        tipe: this.isKredit,
        isShow: this.isShow,
    },
    )),
    (this.jumlahCoA = await this.$axios.get('/hitung/CoA', {
        params: {
            cabang_id: this.cabang_id,
        },
    })),
    )),
},

async deletePost() {
    //delete data post by ID
    await this.$axios.delete(`/CoA/${this.id}`).then(() => {
        //remove item array by index
        // this.CoA.data.result.data.splice(this.index, 1)
        this.id = null
        this.index = null
        this.modal = !this.modal
        this.refetch()
    })
},

openModal(id, index) {
    this.id = id
    this.index = index
    this.modal = !this.modal
},

closeModal() {
    this.modal = false
    this.modalimport = false
    this.modalExport = false
},

//Method untuk export CoA
async exportExcel() {
    this.loading = true
    let newWindow = window.open()
    await this.$axios
        .get('/CoA/export', {
            params: {
                cabang_id: this.cabang_id,
            },
        })
        .then((response) => {
            const data = response.data
            const blob = new Blob([data], { type: 'application/vnd.ms-excel' })
            const url = window.URL.createObjectURL(blob)
            newWindow.location.href = url
        })
}

```

```

    },
    responseType: 'blob',
  })
  .then((response) => {
    this.loading = false
    const href = URL.createObjectURL(response.data)

    // create "a" HTML element with href to file & click
    const link = document.createElement('a')
    link.href = href
    link.setAttribute('download', 'CoA.xlsx') //or any other
extension
    link.setAttribute('target', '_blank') //or any other extension
    document.body.appendChild(link)
    link.click()

    // clean up "a" element & remove ObjectURL
    document.body.removeChild(link)
    URL.revokeObjectURL(href)
    this.modalExport = false
  })

  // const response = await this.$http
  // .get('http://Lazismu-backend.test/api/kasbesar/export')
  // .then((response) => {
  //   fileDownload(response.data, 'pedoman.xlsx')
  // })
},
//Method for dots three
toggleDropdown(kode) {
  if (this.isHidden === false) {
    this.isHidden = kode
    this.$emit('change', this.isHidden)
  } else {
    this.isHidden = false
  }
},
},
}
</script>

```

indexfrontoffice.vue

Sourcecode 8 Indexfrontoffice.vue

```
<template>
  <div
    class="lg:pr-[70px] py-[50px] lg:ml-[320px] px-4 xl:ml-[365px] lg:pl-
0"
    :class="sidebar ? 'xl:ml-[365px]' : 'xl:ml-[65px]'"
  >
    <!-- Top Section -->
    <Modal
      :onClick="deletePost"
      v-show="modal"
      :toggleModal="modal"
      :close="closeModal"
      :wrongaction="hapus"
    >Hapus akun
    <b class="text-red-700"
      ><i>{{ msg }}</i></b>
    </Modal>
  >
  <Modal
    :onClick="exportExcel"
    v-show="modalExport"
    :toggleModal="modalExport"
    :close="closeModal"
    :action="upload"
  >Export dalam bentuk Excel ?</Modal>
  >
  <section
    class="flex flex-col flex-wrap justify-between gap-6 md:items-
center md:flex-row"
  >
    <div class="flex items-center justify-between gap-4">
      <a href="#" id="toggleOpenSidebar" @click="toggleSidebar">
        <svg
          class="w-6 h-6 text-dark"
          fill="none"
          stroke="currentColor"
          viewBox="0 0 24 24"
          xmlns="http://www.w3.org/2000/svg"
        >
          <path
            stroke-linecap="round"
            stroke-linejoin="round"

```

```

        stroke-width="2"
        d="M4 6h16M4 12h16M4 18h7"
    ></path>
</svg>
</a>
<div class="text-[32px] font-semibold text-dark">Front
Offices</div>
</div>
<div class="text-[16px] flex">
    
    <p v-if="role === 1" class="">Administrator</p>
    <p v-if="role === 2" class="">Front Office</p>
    <p v-if="role === 3" class="">Back Office</p>
</div>
</section>

<section class="pt-[50px]">
<!-- Section Header -->
<div class="mb-[30px]">
    <div
        class="flex flex-col justify-between gap-6 sm:items-center
sm:flex-row"
    >
        <div>
            <div class="text-xl font-medium text-dark">Statistics</div>
        </div>
    </div>
</div>

<div class="grid gap-6 sm:grid-cols-2 md:grid-cols-3 lg:gap-11">
<div class="card !gap-y-10">
    <div class="flex items-center justify-between">
        <div>
            <p class="text-grey">Pemasukan</p>
            <div class="text-[32px] font-bold text-dark mt-[6px]">
                <p v-if="$fetchState.pending">Fetching roles...</p>
                <p v-else>Rp. {{ formatPrice(hitung.data.result) }}</p>
            </div>
        </div>
    </div>
</div>
</div>
<!-- <div class="card !gap-y-10">

```

```

<div class="flex items-center justify-between">
  <div>
    <p class="text-grey">Data Belum Terverifikasi</p>
    <div class="text-[32px] font-bold text-dark mt-[6px]">1</div>
  </div>
</div>
</div>
<div class="card !gap-y-10">
  <div class="flex items-center justify-between">
    <div>
      <p class="text-grey">Verifikasi</p>
      <div class="text-[32px] font-bold text-dark mt-[6px]">12</div>
    </div>
  </div>
</div> -->
</div>
</section>

<section class="pt-[50px]">
  <!-- Section Header -->
  <div class="mb-[30px]">
    <div class="flex items-center justify-between gap-6">
      <div>
        <div class="text-xl font-medium text-dark">Data Kasir</div>
        <p class="text-grey">Front Offices</p>
      </div>
      <div class="flex items-center">
        <div class="relative">
          <input
            name="start"
            type="date"
            v-model="from"
            class="border border-gray-300 text-gray-900 text-sm
rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full pl-
10 p-2.5"
          />
        </div>

        <span class="mx-4 text-gray-500">to</span>
        <div class="relative mr-4">
          <input
            v-model="to"
            name="end"

```



```

        type="date"
        class="border border-gray-300 text-gray-900 text-sm
rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full pl-
10 p-2.5"
        placeholder="Select date end"
    />
</div>

<a
download=""
@click="modalExport = !modalExport"
class="button cursor-pointer py-2 px-4 bg-green-400 rounded-
lg text-white"
>
    Export to Excel
</a>
</div>
</div>
</div>

<div class="md:container md:mx-auto">
<div class="overflow-x-auto relative shadow-md sm:rounded-lg">
<table
class="w-full text-sm text-left text-gray-500 dark:text-gray-400"
>
<caption
class="p-5 text-lg font-semibold text-left text-gray-900 bg-
white"
>
<div class="bg-white flex justify-between items-center">
<label for="tabel-search" class="sr-only">Search</label>
<div class="relative mt-1">
<div
class="flex absolute inset-y-0 left-0 items-center pl-3
pointer-events-none"
>
<svg
class="w-5 h-5 text-gray-500 dark:text-gray-400"
aria-hidden="true"
fill="currentColor"
viewBox="0 0 20 20"
xmlns="http://www.w3.org/2000/svg"
>

```

```

        <path
          fill-rule="evenodd"
          d="M8 4a4 4 0 100 8 4 4 0 000-8zM2 8a6 6 0 1110.89
3.476l4.817 4.817a1 1 0 01-1.414 1.414l-4.816-4.816A6 6 0 012 8z"
          clip-rule="evenodd"
        ></path>
      </svg>
    </div>
    <input
      type="text"
      id="tabel-search"
      v-model.lazy="keywords"
      class="block p-2 pl-10 w-80 text-sm text-gray-900 bg-gray-
50 rounded-lg border border-gray-300 focus:ring-blue-500
focus:border-blue-500"
      placeholder="Cari no. bukti, uraian transaksi"
    />
  </div>
  <NuxtLink
    :to="{
      name: 'cabang-id-frontoffices-create',
      params: { id: cabang_id },
    }"
    class="btn-sm px-4 py-2 rounded-lg bg-oranye-400 text-sm
text-white"
  >
    Tambah
  </NuxtLink>
</div>
</caption>
<thead class="text-xs text-white uppercase bg-oranye-400">
<tr>
  <th scope="col" class="py-3 px-6">No</th>
  <th scope="col" class="py-3 px-6">No. bukti Penerima</th>
  <th scope="col" class="py-3 px-6">Uraian Transaksi</th>
  <th scope="col" class="py-3 px-6">Tanggal</th>
  <th scope="col" class="py-3 px-6">Muzakki</th>
  <th scope="col" class="py-3 px-6">Penerima</th>
  <th scope="col" class="py-3 px-6">Akun Debit</th>
  <th scope="col" class="py-3 px-6">Akun Kredit</th>
  <th scope="col" class="py-3 px-6">Tempat Bayar</th>
  <th scope="col" class="py-3 px-6">Jumlah</th>
  <th scope="col" class="py-3 px-6">Aksi</th>

```

```

        <th scope="col" class="">
            <span class="sr-only">Edit</span>
        </th>
    </tr>
</thead>
<tbody>
<p v-if="$fetchState.pending">Fetching roles...</p>
<p v-else-if="$fetchState.error">An error occurred :(</p>

<tr
    class="bg-white border-b"
    v-else
    v-for="(item, index) in frontoffice.data.result.data"
    :key="index"
>
    <th scope="row" class="py-4 px-6 font-medium text-gray-
900">
        {{
            frontoffice.data.result.current_page * 10 - 10 + index + 1
        }}
    </th>
    <td class="py-4 px-1">{{ item.nobuktipenerima }}</td>
    <td class="py-4 px-6">{{ item.name }}</td>
    <td class="py-4 px-6">{{ item.tanggal }}</td>
    <td class="py-4 px-6">{{ item.penyetor }}</td>
    <td class="py-4 px-6">{{ item.penerima }}</td>
    <td class="py-4 px-6" v-if="item.CoAdebit !== null">
        {{ item.CoAdebit.name }}
    </td>
    <td v-else></td>
    <td class="py-4 px-6" v-if="item.CoAkredit !== null">
        {{ item.CoAkredit.name }}
    </td>
    <td v-else></td>
    <td class="py-4 px-6">{{ item.tempatbayar }}</td>
    <td class="py-4 px-6">{{ item.jumlah }}</td>
    <td class="py-4 px-6">
        <button
            id="dropdownMenuIconButton"
            data-dropdown-toggle="dropdownDots"
            class="inline-flex p-2 text-sm font-medium text-center
text-gray-900 bg-white rounded-lg hover:bg-gray-100 focus:ring-4
focus:outline-none"

```

```

        type="button"
        @click="toggleDropdown(item.id)"
      >
      <svg
        class="w-6 h-6"
        aria-hidden="true"
        fill="currentColor"
        viewBox="0 0 20 20"
        xmlns="http://www.w3.org/2000/svg"
      >
        <path
          d="M10 6a2 2 0 110-4 2 2 0 010 4zM10 12a2 2 0 110-4 2 2 0 010 4zM10 18a2 2 0 110-4 2 2 0 010 4z"
        ></path>
      </svg>
    </button>
    <!-- Dropdown menu -->
    <div
      v-show="isHidden"
      refs="drop"
      id="dropdownDots"
      class="bg-white rounded divide-y divide-gray-100 shadow
absolute"
      :class="isHidden === item.id ? 'show' : 'hidden'"
    >
      <ul
        class="py-1 text-sm text-gray-700"
        aria-labelledby="dropdownMenuIconButton"
      >
        <li>
          <Nuxt-Link
            :to="{
              name: 'cabang-id-frontoffices-edit',
              params: { id: cabang_id, item: item.id },
            }"
            class="block py-2 px-4 hover:bg-gray-100"
          >Edit
          </Nuxt-Link>
        </li>
        <li>
          <button
            href="#"
            class="block py-2 px-4 hover:bg-gray-100"

```

```

        @click="openModal(item.id, index), (msg =
item.name)"
      >
        Delete
      </button>
    </li>
  </ul>
</div>
</td>
</tr>
</tbody>
</table>
</div>

<p v-if="$fetchState.pending">Fetching roles...</p>
<p v-else-if="$fetchState.error">An error occurred :(</p>
<nav
  v-else-if="frontoffice.data.result.links"
  class="flex justify-between items-center pt-4"
  aria-label="Tabel navigation"
  >
  <span class="text-sm font-normal text-gray-500"
  >Showing
  <span class="font-semibold text-gray-900">1-10</span>
  of
  <span class="font-semibold text-gray-900">{{
    frontoffice.data.result.last_page
  }}</span></span>
  >

  <ul class="inline-flex items-center -space-x-px">
    <li>
      <button
        :value="frontoffice.data.result.current_page - 1"
        @click="updatePage"
        class="relative block py-2 px-3 ml-0 leading-tight text-gray-
500 bg-white rounded-l-lg border border-gray-300 hover:bg-gray-100
hover:text-gray-700"
        :disabled="frontoffice.data.result.current_page == 1"
      >
        &lt;
      </button>
    </li>

```

```

<p v-if="$fetchState.pending">Fetching roles...</p>
<p v-else-if="$fetchState.error">An error occurred :(</p>
<li
  v-else
  v-for="(ada, index) in frontoffice.data.result.links"
  :key="index"
>
  <button
    v-if="
      ada.label != '&laquo; Sebelumnya' &&
      ada.label != 'Berikutnya &raquo;';
    "
    :value="index"
    @click="updatePage"
    class="py-2 px-3 leading-tight text-gray-500 bg-white border
border-gray-300 hover:bg-gray-100 hover:text-gray-700"
  >
    {{ ada.label }}
  </button>

</li>

<li>
  <button
    :value="nextUrl"
    @click="updatePage"
    class="block py-2 px-3 leading-tight text-gray-500 bg-white
rounded-r-lg border border-gray-300 hover:bg-gray-100 hover:text-
gray-700"
    :disabled="pagenow == frontoffice.data.result.last_page"
  >
  >
  </button>
</li>
</ul>
</nav>
</div>
</section>
</div>
</template>

<script>
import Modal from '@components/Modal.vue'

```

```

import { mapActions } from 'vuex'
export default {
  components: { Modal },
  layout: 'dashboard',
  middleware: ['auth'],
  data() {
    return {
      nextUrl: 2,
      jumlahtot: [],
      pemasukan: null,
      prevUrl: null,
      keywords: null,
      now: null,
      pagenow: 1,
      frontoffice: {},
      isHidden: false,
      role: JSON.parse(localStorage.getItem('role')),
      cabang_id: JSON.parse(localStorage.getItem('cabang_id')),
      from: "",
      to: "",
      hapus: 'Konfirmasi',
      upload: 'Konfirmasi',
      modal: false,
      modalExport: false,
      msg: "",
      hitung: "",
    }
  },
  computed: {
    sidebar() {
      return this.$store.state.sidebar
    },
  },
  watch: {
    keywords(after, before) {
      this.refetch()
      fetchOnServer: false
    },
    from(after, before) {
      this.refetch()
      fetchOnServer: false
    },
    to(after, before) {

```

```

    this.refetch()
    fetchOnServer: false
  },
},

async fetch() {
  await this.$axios
    .get('/frontoffice', {
      params: {
        limit: 10,
        page: this.pagenow,
        search: this.keywords,
        cabang_id: this.cabang_id,
        from: this.from,
        to: this.to,
      },
    })
    .then((response) => {
      this.frontoffice = response
      this.now = this.frontoffice.data.result.current_page
    })
  this.hitung = await this.$axios.get('/hitung/fo', {
    params: {
      cabang_id: this.cabang_id,
    },
  })
},
methods: {
  ...mapActions(['toggleSidebar']),
  formatPrice(value) {
    let val = (value / 1).toFixed(2).replace('.', ',')
    return val.toString().replace(/\B(?=(\d{3})+(?!\d))/g, ',')
  },
  updatePage(event) {
    this.pagenow = parseInt(event.target.value)
    this.nextUrl = 1 + parseInt(event.target.value)
    this.$nuxt.refresh()
  },

  //Method for dots three
  toggleDropdown(kode) {
    console.log(this.isHidden)
  }
}

```



```

    if (this.isHidden === false) {
      this.isHidden = kode
      this.$emit('change', this.isHidden)
    } else {
      this.isHidden = false
    }
  },
  //Method untuk export frontoffice
  async exportExcel() {
    this.loading = true
    let newWindow = window.open()
    await this.$axios
      .get('/frontoffice/export', {
        params: {
          cabang_id: this.cabang_id,
          from: this.from,
          to: this.to,
        },
        responseType: 'blob',
      })
      .then((response) => {
        this.loading = false
        // newWindow.location = 'blob:http://Lazismu-backend.test/api/kasbesar/export'

        const href = URL.createObjectURL(response.data)

        // create "a" HTML element with href to file & click
        const link = document.createElement('a')
        link.href = href
        link.setAttribute('download', 'frontoffice.xlsx') //or any other
extension
        link.setAttribute('target', '_blank') //or any other extension
        document.body.appendChild(link)
        link.click()

        // clean up "a" element & remove ObjectURL
        document.body.removeChild(link)
        URL.revokeObjectURL(href)
        this.closeModal()
      })
  },
  async deletePost() {

```

```

//delete data post by ID
await this.$axios.delete(`/frontoffice/${this.id}`).then(() => {
  //remove item array by index
  // this.CoA.data.result.data.splice(this.index, 1)
  this.id = null
  this.index = null
  this.modal = !this.modal
  this.refetch()
})
},
openModal(id, index) {
  this.id = id
  this.index = index
  this.modal = !this.modal
},
closeModal() {
  this.modal = false
  this.modalimport = false
  this.modalExport = false
},
async refetch() {
  this.frontoffice = await this.$axios.get('/frontoffice', {
    params: {
      limit: 10,
      page: this.pagenow,
      search: this.keywords,
      cabang_id: this.cabang_id,
      from: this.from,
      to: this.to,
    },
  })
  this.hitung = await this.$axios.get('/hitung/fo', {
    params: {
      cabang_id: this.cabang_id,
    },
  })
},
},
}
</script>

```

indexkasbank.vue

Sourcecode 9 Indexkasbank.vue

```
<template>
  <div
    class="lg:pr-[70px] py-[50px] lg:ml-[320px] px-4 lg:pl-0"
    :class="sidebar ? 'xl:ml-[365px]' : 'xl:ml-[65px]'"
  >
    <!-- Top Section -->
    <Modal
      :onClick="deleteKasbank"
      v-show="modal"
      :toggleModal="modal"
      :close="closeModal"
      :wrongaction="hapus"
    >Hapus akun
    <b class="text-red-700"
      ><i>{{ msg }}</i></b>
    ></Modal>
  >
  <Modal
    :onClick="exportExcel"
    v-show="modalExport"
    :toggleModal="modalExport"
    :close="closeModal"
    :action="upload"
  >Export dalam bentuk Excel ?</Modal>
  >
  <section
    class="flex flex-col flex-wrap justify-between gap-6 md:items-center md:flex-row"
  >
    <!-- Responsive class jika berbentuk lebih kecil -->
    <div class="flex items-center justify-between gap-4">
      <a href="#" id="toggleOpenSidebar" @click="toggleSidebar">
        <svg
          class="w-6 h-6 text-dark"
          fill="none"
          stroke="currentColor"
          viewBox="0 0 24 24"
          xmlns="http://www.w3.org/2000/svg"
        >
          <path
            stroke-linecap="round"
            stroke-linejoin="round"

```

```

        stroke-width="2"
        d="M4 6h16M4 12h16M4 18h7"
    ></path>
</svg>
</a>
<div class="text-[32px] font-semibold text-dark">Kas
Bank</div>
</div>
<div class="text-[16px] flex">
    
    <p v-if="role === 1" class="">Administrator</p>
    <p v-if="role === 2" class="">Front Office</p>
    <p v-if="role === 3" class="">Back Office</p>
</div>
</section>

<section class="pt-[50px]">
<!-- Section Header -->
<div class="mb-[30px]">
    <div
        class="flex flex-col justify-between gap-6 sm:items-center
sm:flex-row"
    >
        <div>
            <div class="text-xl font-medium text-dark">Statistics</div>
        </div>
    </div>
</div>

<p v-if="$fetchState.pending">Pemasukan Loading ...</p>
<p v-else-if="$fetchState.error">An error occurred :(</p>
<div v-else class="grid gap-6 sm:grid-cols-2 md:grid-cols-3
lg:gap-11">
    <div class="card !gap-y-10">
        <div class="flex items-center justify-between">
            <div>
                <p class="text-grey">Pemasukan</p>
                <div class="text-[32px] font-bold text-dark mt-[6px]">
                    Rp. {{ formatPrice(hitung.data.pemasukan) }}
                </div>
            </div>
        </div>
    </div>
</div>

```

```

</div>
<div class="card !gap-y-10">
  <div class="flex items-center justify-between">
    <div>
      <p class="text-grey">Pengeluaran</p>
      <div class="text-[32px] font-bold text-dark mt-[6px]">
        Rp. {{ formatPrice(hitung.data.pengeluaran) }}
      </div>
    </div>
  </div>
</div>
<div class="card !gap-y-10">
  <div class="flex items-center justify-between">
    <div>
      <p class="text-grey">Jumlah Saldo</p>
      <div class="text-[32px] font-bold text-dark mt-[6px]">
        Rp. {{ formatPrice(hitung.data.pemasukan -
hitung.data.pengeluaran) }}
      </div>
    </div>
  </div>
</div>
</div>
</div>
</div>
</section>
<!-- Top Section -->

<section class="pt-[100px]">
  <!-- Section Header -->
  <div class="mb-[30px]">
    <div class="flex items-center justify-between gap-6">
      <div>
        <div class="text-xl font-medium text-dark">Data
Kasir</div>
        <p class="text-grey">Kas Bank</p>
      </div>
      <div class="flex items-center">
        <div class="relative">
          <input
            name="start"
            type="date"
            v-model="from"

```

```

        class="border border-gray-300 text-gray-900 text-sm
rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full
pl-10 p-2.5"
    />
</div>

<span class="mx-4 text-gray-500">to</span>
<div class="relative mr-4">
    <input
        v-model="to"
        name="end"
        type="date"
        class="border border-gray-300 text-gray-900 text-sm
rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full
pl-10 p-2.5"
        placeholder="Select date end"
    />
</div>

<a
    download=""
    @click="modalExport = !modalExport"
    class="button cursor-pointer py-2 px-4 bg-green-400
rounded-lg text-white"
    >
        Export to Excel
    </a>
</div>
</div>
</div>

<div class="md:container md:mx-auto">
    <div class="overflow-x-auto relative shadow-md sm:rounded-
lg">
        <table class="w-full text-sm text-left text-gray-500">
            <caption
                class="p-5 text-lg font-semibold text-left text-gray-900 bg-
white"
            >
                <div class="bg-white flex justify-between items-center">
                    <div class="flex items-center mt-3">
                        <label for="tabel-search" class="sr-
only">Search</label>

```

```

<div class="relative">
  <div
    class="flex absolute inset-y-0 left-0 items-center pl-3
pointer-events-none"
  >
    <svg
      class="w-5 h-5 text-gray-500"
      aria-hidden="true"
      fill="currentColor"
      viewBox="0 0 20 20"
      xmlns="http://www.w3.org/2000/svg"
    >
      <path
        fill-rule="evenodd"
        d="M8 4a4 4 0 100 8 4 4 0 00-8zM2 8a6 6 0
1110.89 3.476l4.817 4.817a1 1 0 01-1.414 1.414l-4.816-4.816A6 6
0 012 8z"
        clip-rule="evenodd"
      ></path>
    </svg>
  </div>
  <input
    type="text"
    id="tabel-search"
    v-model.lazy="keywords"
    class="block p-2 pl-10 w-80 text-sm text-gray-900 bg-
gray-50 rounded-lg border border-gray-300 focus:ring-blue-500
focus:border-blue-500"
    placeholder="Cari Uraian transaksi, no. bukti"
  />
</div>
</div>
<NuxtLink
  :to="{
    name: 'cabang-id-kasbank-create',
    params: { id: cabang_id },
  }"
  class="btn-sm px-4 py-2 rounded-lg bg-oranye-400 text-
sm text-white"
  >
  Tambah
</NuxtLink>
</div>

```

```

</caption>
<thead class="text-xs text-white uppercase bg-oranye-
400">
  <tr>
    <th scope="col" class="py-3 px-6">No</th>
    <th scope="col" class="py-3 px-6">No. bukti Kas</th>
    <th scope="col" class="py-3 px-6">Tanggal</th>
    <th scope="col" class="py-3 px-6">ref</th>
    <th scope="col" class="py-3 px-6">Uraian Transaksi</th>
    <th scope="col" class="py-3 px-6">Nama Akun
Debit</th>
    <th scope="col" class="py-3 px-6">Nama Akun
Kredit</th>
    <th scope="col" class="py-3 px-6">Jumlah</th>
    <th scope="col" class="py-3 px-6">Nama Kasir</th>
    <th scope="col" class="py-3 px-6">Aksi</th>
    <th scope="col" class="">
      <span class="sr-only">Edit</span>
    </th>
  </tr>
</thead>
<tbody>
<p v-if="$fetchState.pending">Fetching roles...</p>
<p v-else-if="$fetchState.error">An error occurred :(</p>

<tr
class="bg-white border-b"
v-else
v-for="(item, index) in kasbank.data.result.data"
:key="index"
>
  <th scope="row" class="py-4 px-6 font-medium text-gray-
900">
    {{ kasbank.data.result.current_page * 10 - 10 + index + 1
}}
  </th>
  <td class="py-4 px-6">{{ item.nobuktikas }}</td>
  <td class="py-4 px-6">{{ item.tanggal }}</td>
  <td class="py-4 px-6">{{ item.ref }}</td>
  <td class="py-4 px-6">{{ item.name }}</td>
  <td class="py-4 px-6" v-if="item.CoAdebit !== null">
    {{ item.CoAdebit.name }}
  </td>
</tr>

```



```

<td v-else></td>
<td class="py-4 px-6" v-if="item.CoAkredit !== null">
  {{ item.CoAkredit.name }}
</td>
<td v-else></td>
<td class="py-4 px-6">{{ item.jumlah }}</td>
<td class="py-4 px-6">{{ item.penerima }}</td>
<td class="py-4 px-10">
  <button
    id="dropdownMenuIconButton"
    data-dropdown-toggle="dropdownDots"
    class="inline-flex p-2 text-sm font-medium text-center
text-gray-900 bg-white rounded-lg hover:bg-gray-100 focus:ring-4
focus:outline-none"
    type="button"
    @click="toggleDropdown(item.id)"
  >
    <svg
      class="w-6 h-6"
      aria-hidden="true"
      fill="currentColor"
      viewBox="0 0 20 20"
      xmlns="http://www.w3.org/2000/svg"
    >
      <path
        d="M10 6a2 2 0 110-4 2 2 0 010 4zM10 12a2 2 0
110-4 2 2 0 010 4zM10 18a2 2 0 110-4 2 2 0 010 4z"
      ></path>
    </svg>
  </button>
  <!-- Dropdown menu -->
  <div
    v-show="isHidden"
    refs="drop"
    id="dropdownDots"
    class="bg-white rounded divide-y divide-gray-100
shadow absolute"
    :class="isHidden === item.id ? 'show' : 'hidden'"
  >
    <ul
      class="py-1 text-sm text-gray-700"
      aria-labelledby="dropdownMenuIconButton"
    >

```

```

        <li>
          <Nuxt-Link
            :to="{
              name: 'cabang-id-kasbank-edit',
              params: { id: cabang_id, item: item.id },
            }"
            class="block py-2 px-4 hover:bg-gray-100"
            >Edit
          </Nuxt-Link>
        </li>
        <li>
          <button
            href="#"
            class="block py-2 px-4 hover:bg-gray-100"
            @click="openModal(item.id, index), (msg =
item.name)"
          >
            <!-- deleteKasbank(item, index) -->
            Delete
          </button>
        </li>
      </ul>
    </div>
  </td>
</tr>
<tr class="p-4"></tr>
</tbody>
</table>
</div>

<p v-if="$fetchState.pending">Fetching roles...</p>
<p v-else-if="$fetchState.error">An error occurred :(</p>
<nav
  v-else-if="kasbank.data.result.links"
  class="flex justify-between items-center pt-4"
  aria-label="Tabel navigation"
>
  <span class="text-sm font-normal text-gray-500"
  >Showing
  <span class="font-semibold text-gray-900">1-10</span>
  of
  <span class="font-semibold text-gray-900">{{
kasbank.data.result.last_page

```

```

    }}</span></span>
  >
  <ul class="inline-flex items-center -space-x-px">
    <li>
      <button
        :value="kasbank.data.result.current_page - 1"
        @click="updatePage"
        class="relative block py-2 px-3 ml-0 leading-tight text-
gray-500 bg-white rounded-l-lg border border-gray-300 hover:bg-
gray-100 hover:text-gray-700"
        :disabled="kasbank.data.result.current_page == 1"
      >
        &lt;
      </button>
    </li>
    <p v-if="$fetchState.pending">Fetching roles...</p>
    <p v-else-if="$fetchState.error">An error occurred :(</p>
    <li>
      v-else
      v-for="(ada, index) in kasbank.data.result.links"
      :key="index"
    >
      <button
        v-if="
          ada.label != '&laquo; Sebelumnya' &&
          ada.label != 'Berikutnya &raquo;'"
        :value="index"
        @click="updatePage"
        class="py-2 px-3 leading-tight text-gray-500 bg-white
border border-gray-300 hover:bg-gray-100 hover:text-gray-700"
      >
        {{ ada.label }}
      </button>
    </li>
    <li>
      <button
        :value="nextUrl"
        @click="updatePage"
        class="block py-2 px-3 leading-tight text-gray-500 bg-
white rounded-r-lg border border-gray-300 hover:bg-gray-100
hover:text-gray-700"

```

```

        :disabled="pagenow == kasbank.data.result.last_page"
      >
      >
    </button>
  </li>
</ul>
</nav>
</div>
</section>
</div>
</template>

<script>
import Modal from '@components/Modal.vue'
import { mapActions } from 'vuex'

export default {
  components: { Modal },
  layout: 'dashboard',
  middleware: 'auth',
  data() {
    return {
      isHidden: false,
      role: JSON.parse(localStorage.getItem('role')),
      cabang_id: JSON.parse(localStorage.getItem('cabang_id')),
      lastPage: 1,
      nextUrl: 2,
      prevUrl: "",
      keywords: null,
      pagenow: null,
      kasbank: {},
      msg: "",
      from: "",
      to: "",

      hapus: 'Konfirmasi',
      upload: 'Konfirmasi',
      modal: false,
      modalExport: false,
      hitung: {},
    }
  },
  computed: {

```

```

sidebar() {
  return this.$store.state.sidebar
},
},
watch: {
  keywords(after, before) {
    this.refetch()
    fetchOnServer: false
  },

  from(after, before) {
    this.refetch()
    fetchOnServer: false
  },
  to(after, before) {
    this.refetch()
    fetchOnServer: false
  },
},
},
async fetch() {
  this.kasbank = await this.$axios.get('/kasbank', {
    params: {
      limit: 10,
      page: this.pagenow,
      search: this.keywords,
      cabang_id: this.cabang_id,
      from: this.from,
      to: this.to,
    },
  })
  this.hitung = await this.$axios.get('/hitung/kasbank', {
    params: {
      cabang_id: this.cabang_id,
    },
  })
},
methods: {
  ...mapActions(['toggleSidebar']),

  formatPrice(value) {
    let val = (value / 1).toFixed(2).replace('.', ',')
    return val.toString().replace(/\B(?=(\d{3})+(?!\d))/g, ',')
  },
},

```

```

updatePage(event) {
  this.pagenow = parseInt(event.target.value)
  this.refetch()
  this.nextUrl = 1 + parseInt(event.target.value)
},

async refetch() {
  this.kasbank = await this.$axios.get('/kasbank', {
    params: {
      limit: 10,
      page: this.pagenow,
      search: this.keywords,
      cabang_id: this.cabang_id,
      from: this.from,
      to: this.to,
    },
  })
  this.hitung = await this.$axios.get('/hitung/kasbank', {
    params: {
      cabang_id: this.cabang_id,
    },
  })
},

//this for delete
async deleteKasbank() {
  await this.$axios.delete(`/kasbank/${this.id}`).then(() => {
    //remove item array by index
    // this.CoA.data.result.data.splice(this.index, 1)
    this.id = null
    this.index = null
    this.modal = !this.modal
    this.refetch()
  })
},

async exportExcel() {
  this.loading = true
  let newWindow = window.open()
  await this.$axios
    .get('/kasbank/export', {
      params: {
        cabang_id: this.cabang_id,

```

```

        from: this.from,
        to: this.to,
    },
    responseType: 'blob',
  })
  .then((response) => {
    this.modalExport = false
    this.loading = false
    // newWindow.location = 'blob:http://Lazismu-
backend.test/api/kasbank/export'

    const href = URL.createObjectURL(response.data)

    // console.log(href);

    // create "a" HTML element with href to file & click
    const link = document.createElement('a')
    link.href = href
    link.setAttribute('download', 'kasbank.xlsx') //or any other
extension
    link.setAttribute('target', '_blank') //or any other extension
    document.body.appendChild(link)
    link.click()

    // clean up "a" element & remove ObjectURL
    document.body.removeChild(link)
    URL.revokeObjectURL(href)
    this.closeModal()
  })

  // const response = await this.$http
  // .get('http://Lazismu-backend.test/api/kasbank/export')
  // .then((response) => {
  //   fileDownload(response.data, 'pedoman.xlsx')
  // })
},

//Method for dots three
toggleDropdown(kode) {
  if (this.isHidden === false) {
    this.isHidden = kode
    this.$emit('change', this.isHidden)
  } else {

```

```

        this.isHidden = false
    }
},
openModal(id, index) {
    this.id = id
    this.index = index
    this.modal = !this.modal
},
closeModal() {
    this.modal = false
    this.modalimport = false
    this.modalExport = false
},
},
}
</script>

```

indexkasbesar.vue

Sourcecode 10 Indexkasbesar.vue

```

<template>
  <div
    class="lg:pr-[70px] py-[50px] lg:ml-[320px] px-4 lg:pl-0"
    :class="sidebar ? 'xl:ml-[365px]' : 'xl:ml-[65px]'"
  >
    <!-- Top Section -->
    <Modal
      :onClick="deleteKasbesar"
      v-show="modal"
      :toggleModal="modal"
      :close="closeModal"
      :wrongaction="hapus"
    >Hapus akun
    <b class="text-red-700"
      ><i>{{ msg }}</i></b>
    </Modal>
  >
  <Modal
    :onClick="exportExcel"
    v-show="modalExport"
    :toggleModal="modalExport"
    :close="closeModal"
    :action="upload"
  >Export dalam bentuk Excel ?</Modal>

```



```

>
<section
  class="flex flex-col flex-wrap justify-between gap-6 md:items-
center md:flex-row"
  >
  <!-- Responsive class jika berbentuk lebih kecil -->
  <div class="flex items-center justify-between gap-4">
    <a href="#" id="toggleOpenSidebar" @click="toggleSidebar">
      <svg
        class="w-6 h-6 text-dark"
        fill="none"
        stroke="currentColor"
        viewBox="0 0 24 24"
        xmlns="http://www.w3.org/2000/svg"
      >
        <path
          stroke-linecap="round"
          stroke-linejoin="round"
          stroke-width="2"
          d="M4 6h16M4 12h16M4 18h7"
        ></path>
      </svg>
    </a>
    <div class="text-[32px] font-semibold text-dark">Kas
  Besar</div>
  </div>
  <div class="text-[16px] flex">
    
    <p v-if="role === 1" class="">Administrator</p>
    <p v-if="role === 2" class="">Front Office</p>
    <p v-if="role === 3" class="">Back Office</p>
  </div>
</section>

<section class="pt-[50px]">
  <!-- Section Header -->
  <div class="mb-[30px]">
    <div
      class="flex flex-col justify-between gap-6 sm:items-center
sm:flex-row"
    >
    <div>

```

```

        <div class="text-xl font-medium text-dark">Statistics</div>
    </div>
</div>
</div>

<p v-if="$fetchState.pending">Pemasukan Loading ...</p>
<p v-else-if="$fetchState.error">An error occurred :(</p>
<div v-else class="grid gap-6 sm:grid-cols-2 md:grid-cols-3
lg:gap-11">
    <div class="card !gap-y-10">
        <div class="flex items-center justify-between">
            <div>
                <p class="text-grey">Pemasukan</p>
                <div class="text-[32px] font-bold text-dark mt-[6px]">
                    Rp. {{ formatPrice(hitung.data.pemasukan) }}
                </div>
            </div>
        </div>
    </div>
    <div class="card !gap-y-10">
        <div class="flex items-center justify-between">
            <div>
                <p class="text-grey">Pengeluaran</p>
                <div class="text-[32px] font-bold text-dark mt-[6px]">
                    Rp. {{ formatPrice(hitung.data.pengeluaran) }}
                </div>
            </div>
        </div>
    </div>
    <div class="card !gap-y-10">
        <div class="flex items-center justify-between">
            <div>
                <p class="text-grey">Jumlah Saldo</p>
                <div class="text-[32px] font-bold text-dark mt-[6px]">
                    Rp. {{ formatPrice(hitung.data.pemasukan -
hitung.data.pengeluaran) }}
                </div>
            </div>
        </div>
    </div>
</div>
</div>
</section>
<!-- Top Section -->

```

```

<section class="pt-[100px]">
  <!-- Section Header -->
  <div class="mb-[30px]">
    <div class="flex items-center justify-between gap-6">
      <div>
        <div class="text-xl font-medium text-dark">Data
Kasir</div>
        <p class="text-grey">Kas Besar</p>
      </div>
      <div class="flex items-center">
        <div class="relative">
          <input
            name="start"
            type="date"
            v-model="from"
            class="border border-gray-300 text-gray-900 text-sm
rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full
pl-10 p-2.5"
          />
        </div>

        <span class="mx-4 text-gray-500">to</span>
        <div class="relative mr-4">
          <input
            v-model="to"
            name="end"
            type="date"
            class="border border-gray-300 text-gray-900 text-sm
rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full
pl-10 p-2.5"
            placeholder="Select date end"
          />
        </div>

        <a
          download=""
          @click="modalExport = !modalExport"
          class="button cursor-pointer py-2 px-4 bg-green-400
rounded-lg text-white"
          >
          Export to Excel
        </a>

```

```

    </div>
  </div>
</div>

<div class="md:container md:mx-auto">
  <div class="overflow-x-auto relative shadow-md sm:rounded-
lg">
    <table class="w-full text-sm text-left text-gray-500">
      <caption
        class="p-5 text-lg font-semibold text-left text-gray-900 bg-
white"
      >
        <div class="bg-white flex justify-between items-center">
          <div class="flex items-center mt-3">
            <label for="tabel-search" class="sr-
only">Search</label>
            <div class="relative">
              <div
                class="flex absolute inset-y-0 left-0 items-center pl-3
pointer-events-none"
              >
                <svg
                  class="w-5 h-5 text-gray-500"
                  aria-hidden="true"
                  fill="currentColor"
                  viewBox="0 0 20 20"
                  xmlns="http://www.w3.org/2000/svg"
                >
                  <path
                    fill-rule="evenodd"
                    d="M8 4a4 4 0 10 8 4 4 0 00-8 4M2 8a6 6 0
1110.89 3.476l4.817 4.817a1 1 0 01-1.414 1.414l-4.816-4.816A6 6
0 012 8z"
                    clip-rule="evenodd"
                  ></path>
                </svg>
              </div>
              <input
                type="text"
                id="tabel-search"
                v-model.lazy="keywords"

```

```

        class="block p-2 pl-10 w-80 text-sm text-gray-900 bg-
gray-50 rounded-lg border border-gray-300 focus:ring-blue-500
focus:border-blue-500"
        placeholder="Cari Uraian transaksi, no. bukti"
    />
</div>
</div>
<NuxtLink
:to="{
name: 'cabang-id-kasbesar-create',
params: { id: cabang_id },
}"
class="btn-sm px-4 py-2 rounded-lg bg-oranye-400 text-
sm text-white"
>
    Tambah
</NuxtLink>
</div>
</caption>
<thead class="text-xs text-white uppercase bg-oranye-
400">
<tr>
<th scope="col" class="py-3 px-6">No</th>
<th scope="col" class="py-3 px-6">No. bukti Kas</th>
<th scope="col" class="py-3 px-6">Tanggal</th>
<th scope="col" class="py-3 px-6">ref</th>
<th scope="col" class="py-3 px-6">Uraian Transaksi</th>
Debit</th>
<th scope="col" class="py-3 px-6">Nama Akun
Kredit</th>
<th scope="col" class="py-3 px-6">Jumlah</th>
<th scope="col" class="py-3 px-6">Nama Kasir</th>
<th scope="col" class="py-3 px-6">Aksi</th>
<th scope="col" class="">
    <span class="sr-only">Edit</span>
</th>
</tr>
</thead>
<tbody>
<p v-if="$fetchState.pending">Fetching roles...</p>
<p v-else-if="$fetchState.error">An error occurred :(</p>

```

```

<tr
  class="bg-white border-b"
  v-else
  v-for="(item, index) in kasbesar.data.result.data"
  :key="index"
>
  <th scope="row" class="py-4 px-6 font-medium text-gray-
900">
    {{ kasbesar.data.result.current_page * 10 - 10 + index +
1 }}
  </th>
  <td class="py-4 px-6">{{ item.nobuktikas }}</td>
  <td class="py-4 px-6">{{ item.tanggal }}</td>
  <td class="py-4 px-6">{{ item.ref }}</td>
  <td class="py-4 px-6">{{ item.name }}</td>
  <td class="py-4 px-6" v-if="item.CoAdebit !== null">
    {{ item.CoAdebit.name }}
  </td>
  <td v-else></td>
  <td class="py-4 px-6" v-if="item.CoAkredit !== null">
    {{ item.CoAkredit.name }}
  </td>
  <td v-else></td>
  <td class="py-4 px-6">{{ item.jumlah }}</td>
  <td class="py-4 px-6">{{ item.penerima }}</td>
  <td class="py-4 px-10">
    <button
      id="dropdownMenuIconButton"
      data-dropdown-toggle="dropdownDots"
      class="inline-flex p-2 text-sm font-medium text-center
text-gray-900 bg-white rounded-lg hover:bg-gray-100 focus:ring-4
focus:outline-none"
      type="button"
      @click="toggleDropdown(item.id)"
    >
      <svg
        class="w-6 h-6"
        aria-hidden="true"
        fill="currentColor"
        viewBox="0 0 20 20"
        xmlns="http://www.w3.org/2000/svg"
      >
        <path

```

```

        d="M10 6a2 2 0 110-4 2 2 0 010 4zM10 12a2 2 0
110-4 2 2 0 010 4zM10 18a2 2 0 110-4 2 2 0 010 4z"
      ></path>
    </svg>
  </button>
  <!-- Dropdown menu -->
  <div
    v-show="isHidden"
    refs="drop"
    id="dropdownDots"
    class="bg-white rounded divide-y divide-gray-100
shadow absolute"
    :class="isHidden === item.id ? 'show' : 'hidden'"
  >
    <ul
      class="py-1 text-sm text-gray-700"
      aria-labelledby="dropdownMenuIconButton"
    >
      <li>
        <Nuxt-Link
          :to="{
            name: 'cabang-id-kasbesar-edit',
            params: { id: cabang_id, item: item.id },
          }"
          class="block py-2 px-4 hover:bg-gray-100"
        >Edit
        </Nuxt-Link>
      </li>
      <li>
        <button
          href="#"
          class="block py-2 px-4 hover:bg-gray-100"
          @click="openModal(item.id, index), (msg =
item.name)"
        >
          <!-- deleteKasbesar(item, index) -->
          Delete
        </button>
      </li>
    </ul>
  </div>
</td>
</tr>

```

```

        <tr class="p-4"></tr>
    </tbody>
</tabel>
</div>

<p v-if="$fetchState.pending">Fetching roles...</p>
<p v-else-if="$fetchState.error">An error occurred :(</p>
<nav
  v-else-if="kasbesar.data.result.links"
  class="flex justify-between items-center pt-4"
  aria-label="Tabel navigation"
>
  <span class="text-sm font-normal text-gray-500"
  >Showing
  <span class="font-semibold text-gray-900">1-10</span>
  of
  <span class="font-semibold text-gray-900">{{
    kasbesar.data.result.last_page
  }}</span></span>
  >
  <ul class="inline-flex items-center -space-x-px">
    <li>
      <button
        :value="kasbesar.data.result.current_page - 1"
        @click="updatePage"
        class="relative block py-2 px-3 ml-0 leading-tight text-
        gray-500 bg-white rounded-l-lg border border-gray-300 hover:bg-
        gray-100 hover:text-gray-700"
        :disabled="kasbesar.data.result.current_page == 1"
      >
        &lt;
      </button>
    </li>
    <p v-if="$fetchState.pending">Fetching roles...</p>
    <p v-else-if="$fetchState.error">An error occurred :(</p>
    <li
      v-else
      v-for="(ada, index) in kasbesar.data.result.links"
      :key="index"
    >
      <button
        v-if="
          ada.label != '&laquo; Sebelumnya' &&

```



```

        ada.label != 'Berikutnya &raquo;'
        "
        :value="index"
        @click="updatePage"
        class="py-2 px-3 leading-tight text-gray-500 bg-white
border border-gray-300 hover:bg-gray-100 hover:text-gray-700"
    >
        {{ ada.label }}
    </button>
</li>

<li>
    <button
        :value="nextUrl"
        @click="updatePage"
        class="block py-2 px-3 leading-tight text-gray-500 bg-
white rounded-r-lg border border-gray-300 hover:bg-gray-100
hover:text-gray-700"
        :disabled="pagenow == kasbesar.data.result.last_page"
    >
        >
    </button>
</li>
</ul>
</nav>
</div>
</section>
</div>
</template>

<script>
import Modal from '@components/Modal.vue'
import { mapActions } from 'vuex'

export default {
  components: { Modal },
  layout: 'dashboard',
  middleware: 'auth',
  data() {
    return {
      isHidden: false,
      cabang_id: JSON.parse(localStorage.getItem('cabang_id')),
      role: JSON.parse(localStorage.getItem('role')),
    }
  }
}

```

```

    lastPage: 1,
    nextUrl: 2,
    prevUrl: "",
    keywords: null,
    pagenow: null,
    kasbesar: {},
    msg: "",
    from: "",
    to: "",
    activeGrade: null,
    data1: null,
    data2: null,

    hapus: 'Konfirmasi',
    upload: 'Konfirmasi',
    modal: false,
    modalExport: false,
    hitung: {},
  }
},
computed: {
  sidebar() {
    return this.$store.state.sidebar
  },
},
watch: {
  keywords(after, before) {
    this.refetch()
    fetchOnServer: false
  },

  from(after, before) {
    this.refetch()
    fetchOnServer: false
  },
  to(after, before) {
    this.refetch()
    fetchOnServer: false
  },
},
async fetch() {
  this.kasbesar = await this.$axios.get('/kasbesar', {
    params: {

```

```

        limit: 10,
        page: this.pagenow,
        search: this.keywords,
        cabang_id: this.cabang_id,
        from: this.from,
        to: this.to,
    },
  })
  this.hitung = await this.$axios.get('/hitung/kasbesar', {
    params: {
      cabang_id: this.cabang_id,
    },
  })
},
methods: {
  ...mapActions(['toggleSidebar']),
  formatPrice(value) {
    let val = (value / 1).toFixed(2).replace('.', ',')
    return val.toString().replace(/\B(?=(\d{3})+(?!\d))/g, ',')
  },

  updatePage(event) {
    this.pagenow = parseInt(event.target.value)
    this.refetch()
    this.nextUrl = 1 + parseInt(event.target.value)
  },

  async refetch() {
    this.kasbesar = await this.$axios.get('/kasbesar', {
      params: {
        limit: 10,
        page: this.pagenow,
        search: this.keywords,
        cabang_id: this.cabang_id,
        from: this.from,
        to: this.to,
      },
    })
    this.hitung = await this.$axios.get('/hitung/kasbesar', {
      params: {
        cabang_id: this.cabang_id,
      },
    })
  })
}

```

```

    },

    //this for delete
    async deleteKasbesar() {
        //delete data post by ID
        await this.$axios.delete(`/kasbesar/${this.id}`).then(() => {
            //remove item array by index
            // this.kasbesar.data.result.data.splice(index, 1)
            this.id = null
            this.index = null
            this.modal = !this.modal
            this.refetch()
        })
    },
    async exportExcel() {
        this.loading = true
        let newWindow = window.open()
        await this.$axios
            .get('/kasbesar/export', {
                params: {
                    cabang_id: this.cabang_id,
                    from: this.from,
                    to: this.to,
                },
                responseType: 'blob',
            })
            .then((response) => {
                this.loading = false
                // newWindow.location = 'blob:http://Lazismu-backend.test/api/kasbesar/export'

                const href = URL.createObjectURL(response.data)

                // console.log(href);

                // create "a" HTML element with href to file & click
                const link = document.createElement('a')
                link.href = href
                link.setAttribute('download', 'kasbesar.xlsx') //or any other
                extension
                link.setAttribute('target', '_blank') //or any other extension
                document.body.appendChild(link)
                link.click()
            })
    },

```

```

        // clean up "a" element & remove ObjectURL
        document.body.removeChild(link)
        URL.revokeObjectURL(href)
    })

    // const response = await this.$http
    // .get('http://Lazismu-backend.test/api/kasbesar/export')
    // .then((response) => {
    //     fileDownload(response.data, 'pedoman.xlsx')
    // })
    },

    //Method for dots three
    toggleDropdown(kode) {
        if (this.isHidden === false) {
            this.isHidden = kode
            this.$emit('change', this.isHidden)
        } else {
            this.isHidden = false
        }
    },
    openModal(id, index) {
        this.id = id
        this.index = index
        this.modal = !this.modal
    },
    closeModal() {
        this.modal = false
        this.modalimport = false
        this.modalExport = false
    },
    },
}
</script>

```

indexkaskecil.vue

Sourcecode 11 Indexkaskecil.vue

```

<template>
  <div
    class="lg:pr-[70px] py-[50px] lg:ml-[320px] px-4 lg:pl-0"
    :class="sidebar ? 'xl:ml-[365px]' : 'xl:ml-[65px]'"
  >

```

```

<!-- Top Section -->
<Modal
  :onClick="deletePost"
  v-show="modal"
  :toggleModal="modal"
  :close="closeModal"
  :wrongaction="hapus"
  >Hapus uraian transaksi
  <b class="text-red-700"
    ><i>{{ msg }}</i></b>
  ></Modal
>
<Modal
  :onClick="exportExcel"
  v-show="modalExport"
  :toggleModal="modalExport"
  :close="closeModal"
  :action="upload"
  >Export dalam bentuk Excel ?</Modal
>
<!-- Top Section -->
<section
  class="flex flex-col flex-wrap justify-between gap-6 md:items-
center md:flex-row"
>
  <div class="flex items-center justify-between gap-4">
    <a href="#" id="toggleOpenSidebar" @click="toggleSidebar">
      <svg
        class="w-6 h-6 text-dark"
        fill="none"
        stroke="currentColor"
        viewBox="0 0 24 24"
        xmlns="http://www.w3.org/2000/svg"
      >
        <path
          stroke-linecap="round"
          stroke-linejoin="round"
          stroke-width="2"
          d="M4 6h16M4 12h16M4 18h7"
        ></path>
      </svg>
    </a>

```

```

    <div class="text-[32px] font-semibold text-dark">Kas
    Kecil</div>
  </div>
  <div class="text-[16px] flex">
    
    <p v-if="role === 1" class="">Administrator</p>
    <p v-if="role === 2" class="">Front Office</p>
    <p v-if="role === 3" class="">Back Office</p>
  </div>
</section>

<section class="pt-[50px]">
  <!-- Section Header -->
  <div class="mb-[30px]">
    <div
      class="flex flex-col justify-between gap-6 sm:items-center
      sm:flex-row"
    >
      <div>
        <div class="text-xl font-medium text-dark">Statistics</div>
      </div>
      </div>
    </div>
    <p v-if="$fetchState.pending">Pemasukan Loading ...</p>
    <p v-else-if="$fetchState.error">An error occurred :(</p>
    <div v-else class="grid gap-6 sm:grid-cols-2 md:grid-cols-3
    lg:gap-11">
      <div class="card !gap-y-10">
        <div class="flex items-center justify-between">
          <div>
            <p class="text-grey">Pemasukan</p>

            <div class="text-[32px] font-bold text-dark mt-[6px]">
              Rp. {{ formatPrice(hitung.data.pemasukan) }}
            </div>
          </div>
        </div>
      </div>
      <div class="card !gap-y-10">
        <div class="flex items-center justify-between">
          <div>
            <p class="text-grey">Pengeluaran</p>

```

```

        <div class="text-[32px] font-bold text-dark mt-[6px]">
            Rp. {{ formatPrice(hitung.data.pengeluaran) }}
        </div>
    </div>
</div>
<div class="card !gap-y-10">
    <div class="flex items-center justify-between">
        <div>
            <p class="text-grey">Jumlah Saldo</p>
            <div class="text-[32px] font-bold text-dark mt-[6px]">
                Rp. {{ formatPrice(hitung.data.pemasukan -
                hitung.data.pengeluaran) }}
            </div>
        </div>
    </div>
</div>
</div>
</div>
</div>
</section>

<section class="pt-[50px]">
    <!-- Section Header -->
    <div class="mb-[30px]">
        <div class="flex items-center justify-between gap-6">
            <div>
                <div class="text-xl font-medium text-dark">Data
                Kasir</div>
                <p class="text-grey">Kas Kecil</p>
            </div>
            <div class="flex items-center">
                <div class="relative">
                    <input
                        name="start"
                        type="date"
                        v-model="from"
                        class="border border-gray-300 text-gray-900 text-sm
                        rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full
                        pl-10 p-2.5"
                    />
                </div>

                <span class="mx-4 text-gray-500">to</span>
                <div class="relative mr-4">

```



```

        <input
          v-model="to"
          name="end"
          type="date"
          class="border border-gray-300 text-gray-900 text-sm
rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full
pl-10 p-2.5"
          placeholder="Select date end"
        />
      </div>

      <a
        download=""
        @click="modalExport = !modalExport"
        class="button cursor-pointer py-2 px-4 bg-green-400
rounded-lg text-white"
      >
        Export to Excel
      </a>
    </div>
  </div>

  <div class="md:container md:mx-auto">
    <div class="overflow-x-auto relative shadow-md sm:rounded-
lg">
      <table
        class="w-full text-sm text-left text-gray-500 dark:text-gray-
400 whitespace-nowrap"
      >
        <caption
          class="p-5 text-lg font-semibold text-left text-gray-900 bg-
white"
        >
          <div class="bg-white flex justify-between items-center">
            <label for="tabel-search" class="sr-only">Search</label>
            <div class="relative mt-1">
              <div
                class="flex absolute inset-y-0 left-0 items-center pl-3
pointer-events-none"
              >
                <svg
                  class="w-5 h-5 text-gray-500 dark:text-gray-400"

```

```

        aria-hidden="true"
        fill="currentColor"
        viewBox="0 0 20 20"
        xmlns="http://www.w3.org/2000/svg"
    >
    <path
        fill-rule="evenodd"
        d="M8 4a4 4 0 100 8 4 4 0 000-8zM2 8a6 6 0
1110.89 3.476l4.817 4.817a1 1 0 01-1.414 1.414l-4.816-4.816A6 6
0 012 8z"
        clip-rule="evenodd"
    ></path>
</svg>
</div>
<input
    type="text"
    id="tabel-search"
    v-model.lazy="keywords"
    class="block p-2 pl-10 w-80 text-sm text-gray-900 bg-
gray-50 rounded-lg border border-gray-300 focus:ring-blue-500
focus:border-blue-500"
    placeholder="Cari uraian transaksi, no. bukti"
/>
</div>
<NuxtLink
    :to="{
        name: 'cabang-id-kaskecil-create',
        params: { id: cabang_id },
    }"
    class="btn-sm px-4 py-2 rounded-lg bg-oranye-400 text-
sm text-white"
    >
    Tambah
</NuxtLink>
</div>
</caption>
<thead class="text-xs text-white uppercase bg-oranye-
400">
<tr>
<th scope="col" class="py-3 px-6">No</th>
<th scope="col" class="py-3 px-6">No. bukti Kas</th>
<th scope="col" class="py-3 px-6">Uraian Transaksi</th>
<th scope="col" class="py-3 px-6">Tanggal</th>

```

```

<th scope="col" class="py-3 px-6">Penerima</th>
<th scope="col" class="py-3 px-6">Akun Debit</th>
<th scope="col" class="py-3 px-6">Akun Kredit</th>
<th scope="col" class="py-3 px-6">Jumlah</th>
<th scope="col" class="py-3 px-6">Aksi</th>
<th scope="col" class="">
  <span class="sr-only">Edit</span>
</th>
</tr>
</thead>
<tbody>
<p v-if="$fetchState.pending">Fetching roles...</p>
<p v-else-if="$fetchState.error">An error occurred :(</p>

<tr
  class="bg-white border-b"
  v-else
  v-for="(item, index) in kaskecil.data.result.data"
  :key="index"
>
  <th scope="row" class="py-4 px-6 font-medium text-gray-
900">
    {{ kaskecil.data.result.current_page * 10 - 10 + index + 1
}}
  </th>
  <td class="py-4 px-6">{{ item.nobuktikas }}</td>
  <td class="py-4 px-6">{{ item.name }}</td>
  <td class="py-4 px-6">{{ item.tanggal }}</td>
  <td class="py-4 px-6">{{ item.penerima }}</td>
  <td class="py-4 px-6" v-if="item.CoAdebit !== null">
    {{ item.CoAdebit.name }}
  </td>
  <td v-else></td>
  <td class="py-4 px-6" v-if="item.CoAkredit !== null">
    {{ item.CoAkredit.name }}
  </td>
  <td v-else></td>
  <td class="py-4 px-6">{{ formatPrice(item.jumlah) }}</td>
  <td class="py-4 px-6">
    <button
      id="dropdownMenuIconButton"
      data-dropdown-toggle="dropdownDots"

```

```

        class="inline-flex items-center p-2 text-sm font-
medium text-center text-gray-900 bg-white rounded-lg hover:bg-
gray-100 focus:ring-4 focus:outline-none"
        type="button"
        @click="toggleDropdown(item.id)"
    >
    <svg
        class="w-6 h-6"
        aria-hidden="true"
        fill="currentColor"
        viewBox="0 0 20 20"
        xmlns="http://www.w3.org/2000/svg"
    >
        <path
            d="M10 6a2 2 0 110-4 2 2 0 010 4zM10 12a2 2 0
110-4 2 2 0 010 4zM10 18a2 2 0 110-4 2 2 0 010 4z"
        ></path>
    </svg>
</button>

<!-- Dropdown menu -->
<div
    v-show="isHidden"
    refs="drop"
    id="dropdownDots"
    class="bg-white rounded divide-y divide-gray-100
shadow absolute"
    :class="isHidden === item.id ? 'show' : 'hidden'"
    >
    <ul
        class="py-1 text-sm text-gray-700"
        aria-labelledby="dropdownMenuIconButton"
    >
        <li>
            <Nuxt-Link
                :to="{
                    name: 'cabang-id-kaskecil-edit',
                    params: { id: cabang_id, item: item.id },
                }"
                class="block py-2 px-4 hover:bg-gray-100"
            >Edit
            </Nuxt-Link>
        </li>

```

```

        <li>
          <button
            href="#"
            class="block py-2 px-4 hover:bg-gray-100"
            @click="openModal(item.id, index), (msg =
item.name)"
          >
            Delete
          </button>
        </li>
      </ul>
    </div>
    <!-- <div class="flex items-center">
      <NuxtLink :to="{name: ', params: { id: cabang_id }}"
class="bg-gray-100"
      tag="img"
      :src="require('~static/assets/svg/dot3.svg')" >

        </Nuxtlink>
      </div> -->
    </td>
  </tr>
</tbody>
</table>
</div>

<p v-if="$fetchState.pending">Fetching roles...</p>
<p v-else-if="$fetchState.error">An error occurred :(</p>
<nav
  v-else-if="kaskecil.data.result.links"
  class="flex justify-between items-center pt-4"
  aria-label="Tabel navigation"
  >
  <span class="text-sm font-normal text-gray-500"
  >Showing
  <span class="font-semibold text-gray-900">1-10</span>
  of
  <span class="font-semibold text-gray-900">{{
  kaskecil.data.result.last_page
  }}</span></span>
  >
  <ul class="inline-flex items-center -space-x-px">
    <li>

```

```

    <button
      :value="kaskecil.data.result.current_page - 1"
      @click="updatePage"
      class="relative block py-2 px-3 ml-0 leading-tight text-
gray-500 bg-white rounded-l-lg border border-gray-300 hover:bg-
gray-100 hover:text-gray-700"
      :disabled="kaskecil.data.result.current_page == 1"
    >
    <
  </button>
</li>
<p v-if="$fetchState.pending">Fetching roles...</p>
<p v-else-if="$fetchState.error">An error occurred :(</p>
<li
  v-else
  v-for="(ada, index) in kaskecil.data.result.links"
  :key="index"
>
  <button
    v-if="
      ada.label != '&laquo; Sebelumnya' &&
      ada.label != 'Berikutnya &raquo;'"
    :value="index"
    @click="updatePage"
    class="py-2 px-3 leading-tight text-gray-500 bg-white
border border-gray-300 hover:bg-gray-100 hover:text-gray-700"
  >
    {{ ada.label }}
  </button>
</li>

<li>
  <button
    :value="nextUrl"
    @click="updatePage"
    class="block py-2 px-3 leading-tight text-gray-500 bg-
white rounded-r-lg border border-gray-300 hover:bg-gray-100
hover:text-gray-700"
    :disabled="pagenow == kaskecil.data.result.last_page"
  >
  >
  </button>

```

```

        </li>
      </ul>
    </nav>
  </div>
</section>
</div>
</template>

<script>
import Modal from '@/components/Modal.vue'
import { mapActions } from 'vuex'
export default {
  components: { Modal },
  layout: 'dashboard',
  middleware: 'auth',
  data() {
    return {
      isHidden: false,
      lastPage: "",
      nextUrl: 2,
      prevUrl: "",
      keywords: null,
      pagenow: null,
      kaskecil: {},
      msg: "",
      upload: 'Konfirmasi',
      modalExport: false,
      hapus: 'Konfirmasi',
      modal: false,
      hitung: {},
      from: "",
      to: "",
      cabang_id: JSON.parse(localStorage.getItem('cabang_id')),
      role: JSON.parse(localStorage.getItem('role')),
    }
  },
  computed: {
    sidebar() {
      return this.$store.state.sidebar
    },
  },
  watch: {
    keywords(after, before) {

```

```

    this.refetch()
    fetchOnServer: false
  },
  from(after, before) {
    this.refetch()
    fetchOnServer: false
  },
  to(after, before) {
    this.refetch()
    fetchOnServer: false
  },
},
async fetch() {
  this.kaskecil = await this.$axios.get('/kaskecil', {
    params: {
      limit: 10,
      page: this.pagenow,
      search: this.keywords,
      cabang_id: this.cabang_id,
      from: this.from,
      to: this.to,
    },
  })
  this.hitung = await this.$axios.get('/hitung/kaskecil', {
    params: {
      cabang_id: this.cabang_id,
    },
  })
},
methods: {
  ...mapActions(['toggleSidebar']),

  formatPrice(value) {
    const rupiah =
value.toString().replace(/(\d)(?=(\d\d\d)+(?!\d))/g, '$1.')
    // const rupiah = value.toString().toLocaleString('en-GB')

    return rupiah
  },
  updatePage(event) {
    this.pagenow = parseInt(event.target.value)
    this.$nuxt.refresh()
    window.onNuxtReady(() => {

```



```

        window.$nuxt.$router.push('/kaskecil')
    })
  },
  async deletePost() {
    //delete data post by ID
    await this.$axios.delete(`/kaskecil/${this.id}`).then(() => {
      //remove item array by index
      // this.CoA.data.result.data.splice(this.index, 1)
      this.id = null
      this.index = null
      this.modal = !this.modal
      this.refetch()
    })
  },
  openModal(id, index) {
    this.id = id
    this.index = index
    this.modal = !this.modal
  },
  closeModal() {
    this.modal = false
    this.modalimport = false
    this.modalExport = false
  },
  async refetch() {
    this.kaskecil = await this.$axios.get('/kaskecil', {
      params: {
        limit: 10,
        page: this.pagenow,
        search: this.keywords,
        cabang_id: this.cabang_id,
        from: this.from,
        to: this.to,
      },
    })
    this.hitung = await this.$axios.get('/hitung/kaskecil', {
      params: {
        cabang_id: this.cabang_id,
      },
    })
  },
  async exportExcel() {
    this.loading = true
  }
}

```

```

let newWindow = window.open()
await this.$axios
  .get('/kaskecil/export', {
    params: {
      cabang_id: this.cabang_id,
      from: this.from,
      to: this.to,
    },
    responseType: 'blob',
  })
  .then((response) => {
    this.loading = false
    // newWindow.location = 'blob:http://Lazismu-backend.test/api/kasbesar/export'

    const href = URL.createObjectURL(response.data)

    // console.log(href);

    // create "a" HTML element with href to file & click
    const link = document.createElement('a')
    link.href = href
    link.setAttribute('download', 'kaskecil.xlsx') //or any other
extension
    link.setAttribute('target', '_blank') //or any other extension
    document.body.appendChild(link)
    link.click()

    // clean up "a" element & remove ObjectURL
    document.body.removeChild(link)
    URL.revokeObjectURL(href)
    this.closeModal()
  })

// const response = await this.$http
// .get('http://Lazismu-backend.test/api/kasbesar/export')
// .then((response) => {
//   fileDownload(response.data, 'pedoman.xlsx')
// })
// },
toggleDropdown(kode) {
  if (this.isHidden === false) {
    this.isHidden = kode
  }
}

```

```

        this.$emit('change', this.isHidden)
      } else {
        this.isHidden = false
      }
    },
  },
}
</script>

```

indexoverview.vue

Sourcecode 12 Indexoverview.vue

```

<template>
  <div>
    <!-- Main Content -->
    <div
      class="lg:pr-[70px] py-[50px] lg:ml-[320px] px-4 lg:pl-0"
      :class="sidebar ? 'xl:ml-[365px]' : 'xl:ml-[65px]'"
    >
      <!-- Top Section -->
      <section
        class="flex flex-col flex-wrap justify-between gap-6 md:items-center
        md:flex-row"
      >
        <div class="flex items-center justify-between gap-4">
          <a href="#" id="toggleOpenSidebar" @click="toggleSidebar">
            <svg
              class="w-6 h-6 text-dark"
              fill="none"
              stroke="currentColor"
              viewBox="0 0 24 24"
              xmlns="http://www.w3.org/2000/svg"
            >
              <path
                stroke-linecap="round"
                stroke-linejoin="round"
                stroke-width="2"
                d="M4 6h16M4 12h16M4 18h7"
              ></path>
            </svg>
          </a>
          <div class="text-[32px] font-semibold text-dark">Beranda</div>
        </div>
        <div class="text-[16px] flex">

```

```


<p v-if="role === 1" class="">Administrator</p>
<p v-if="role === 2" class="">Front Office</p>
<p v-if="role === 3" class="">Back Office</p>
</div>
</section>
<section class="flex my-8 justify-start gap-2">
  <div
    :v-if="role === 1 || role === 3"
    v-for="(menu, id) in menus"
    :key="id"
  >
    <button
      class="text-white px-4 py-2 mr-0 rounded-lg"
      :class="[
        isActive == menu.statistik
          ? ['border-2 border-blue-500', menu.color]
          : menu.color,
      ]"
      @click.prevent="
        changeMenu(menu.statistik), changeActive(menu.statistik)
      "
    >
      {{ menu.item }}
    </button>
  </div>

  <div v-if="role === 2">
    <button
      class="text-white px-4 py-2 mr-0 rounded-lg bg-red-500
      active:border-cyan-500"
      @click.prevent="changeMenu('frontoffice')"
    >
      Front Office
    </button>
  </div>
</section>

```

```

<section>
  <!-- Section Header -->
  <div class="mb-[30px]">
    <div class="flex items-center">
      <div>
        <div class="text-xl font-medium text-dark">Statistik</div>
        <p class="text-grey">
          Perkembangan Pemasukan dan Pengeluaran dalam satu bulan
        </p>
      </div>
    </div>
  </div>
  <div class="card-chart min-h-[468px] items-center">
    <div class="bg-white flex">
      <line-chart-generator v-if="menu" :menu="menu" />
    </div>
  </div>
</section>
</div>
</div>
</template>

<script>
import { mapActions } from 'vuex'
import bar from '@components/chart/bar.vue'
import LineChartGenerator from '@components/chart/LineChart.vue'
import Doughnut from '@components/chart/DoughnutChart.vue'
import role from '~/middleware/role'
export default {
  layout: 'dashboard',
  components: { bar, LineChartGenerator, Doughnut },
  data() {
    return {
      role: JSON.parse(localStorage.getItem('role')),
      menu: 'frontoffice',
      isActive: 'frontoffice',
      menus: [
        {
          color: 'bg-red-700',
          item: 'Kas Besar',
          statistik: 'kasbesar',
        },
      ],
    }
  }
}

```

```
    color: 'bg-green-700',
    item: 'Front Office',
    statistik: 'frontoffice',
  },
  {
    color: 'bg-blue-700',
    item: 'Kas Kecil',
    statistik: 'kaskecil',
  },
  {
    color: 'bg-yellow-700',
    item: 'Kas Bank',
    statistik: 'kasbank',
  },
  {
    color: 'bg-gray-700',
    item: 'Jurnal Umum',
    statistik: 'jurnalumum',
  },
],
}
},
computed: {
  sidebar() {
    return this.$store.state.sidebar
  },
},
methods: {
  ...mapActions(['toggleSidebar']),
  changeMenu(menu) {
    this.menu = menu
  },
  changeActive(menu) {
    this.isActive = menu
  },
},
}
</script>
```

indexuser.vue

Sourcecode 13 Indexuser.vue

```
<template>
  <div
    class="lg:pr-[70px] py-[50px] lg:ml-[320px] px-4 lg:pl-0"
    :class="sidebar ? 'xl:ml-[365px]' : 'xl:ml-[65px]'"
  >
    <!-- Top Section -->
    <Modal
      :onClick="deletePost"
      v-show="modal"
      :toggleModal="modal"
      :close="closeModal"
      :wrongaction="hapus"
    >Hapus akun
    <b class="text-red-700"
      ><i>{{ msg }}</i></b>
    </Modal>
  >
  <section
    class="flex flex-col flex-wrap justify-between gap-6 md:items-center
    md:flex-row"
  >
    <div class="flex items-center justify-between gap-4">
      <a href="#" id="toggleOpenSidebar" @click="toggleSidebar">
        <svg
          class="w-6 h-6 text-dark"
          fill="none"
          stroke="currentColor"
          viewBox="0 0 24 24"
          xmlns="http://www.w3.org/2000/svg"
        >
          <path
            stroke-linecap="round"
            stroke-linejoin="round"
            stroke-width="2"
            d="M4 6h16M4 12h16M4 18h7"
          ></path>
        </svg>
      </a>
      <div class="text-[32px] font-semibold text-dark">User
    Manajemen</div>
    </div>
    <div class="text-[16px] flex">
```

```

    
  <p v-if="role === 1" class="">Administrator</p>
  <p v-if="role === 2" class="">Front Office</p>
  <p v-if="role === 3" class="">Back Office</p>
</div>
</section>

<section class="pt-[50px]"></section>

<section class="pt-[50px]">
  <!-- Section Header -->
  <div class="mb-[30px]">
    <div class="flex items-center justify-between gap-6">
      <div>
        <div class="text-xl font-medium text-dark">User</div>
        <p class="text-grey">Pengaturan user</p>
      </div>
    </div>
  </div>

  <div
    class="overflow-x-auto relative shadow-md sm:rounded-lg p-2 bg-
white"
  >
    <div class="pb-4 bg-white flex justify-between items-center">
      <label for="tabel-search" class="sr-only">Search</label>

      <div class="flex">
        <div class="relative mt-1">
          <div
            class="flex absolute inset-y-0 left-0 items-center pl-3 pointer-
events-none"
          >
            <svg
              class="w-5 h-5 text-gray-500 dark:text-gray-400"
              aria-hidden="true"
              fill="currentColor"
              viewBox="0 0 20 20"
              xmlns="http://www.w3.org/2000/svg"
            >
              <path
                fill-rule="evenodd"

```



```

        d="M8 4a4 4 0 100 8 4 4 0 000-8zM2 8a6 6 0 1110.89
3.476l4.817 4.817a1 1 0 01-1.414 1.414l-4.816-4.816A6 6 0 012 8z"
        clip-rule="evenodd"
    ></path>
</svg>
</div>
<input
  type="text"
  id="tabel-search"
  class="block p-2 pl-10 w-80 text-sm text-gray-900 bg-gray-50
rounded-lg border border-gray-300 focus:ring-blue-500 focus:border-blue-
500"
  placeholder="Cari nama, email"
  v-model="keywords"
/>
</div>
<div class="mt-1 ml-2">
  <select
    name="role"
    id=""
    class="bg-gray-50 border border-gray-300 text-gray-500 text-sm
rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full py-2 px-
2.5"
    v-model="roleFilter"
  >
    <option value="0" selected>Pilih Role </option>
    <option value="1">Admin</option>
    <option value="2">Front Office</option>
    <option value="3">Back Office</option>
  </select>
</div>
</div>
<NuxtLink
  :to="{
    name: 'cabang-id-user-create',
    params: { id: cabang_id },
  }"
  class="btn-sm px-4 py-2 rounded-lg bg-oranye-400 text-sm text-
white"
  >
  Tambah
</NuxtLink>
</div>

```

```

<tabel
  class="w-full text-sm text-left text-gray-200 hover:border-collapse"
>
<thead class="text-xs text-white uppercase bg-oranye-400">
<tr>
  <th scope="col" class="p-2">
    <div class="flex items-center">No.</div>
  </th>
  <th scope="col" class="py-3 px-6">Nama</th>
  <th scope="col" class="py-3 px-6">Email</th>
  <th scope="col" class="py-3 px-6">Role</th>
  <th scope="col" class="py-3 px-6">Cabang</th>
  <th scope="col" class="py-3 px-6">
    <div class="flex items-center">Aksi</div>
  </th>
</tr>
</thead>
<tbody>
<p v-if="$fetchState.pending">Fetching roles...</p>
<tr
  class="bg-white border-b text-gray-700"
  v-else
  v-for="(item, index) in user.data.result.data"
  :key="index"
>
  <td class="p-4 w-4">
    <div class="flex items-center">
      {{ user.data.result.current_page * 10 - 10 + index + 1 }}
    </div>
  </td>
  <td scope="row" class="py-4 px-6 font-medium text-gray-900">
    {{ item.name }}
  </td>
  <td class="py-4 px-6">{{ item.email }}</td>
  <td class="py-4 px-6">
    {{ rolename[item.role - 1] }}
  </td>

  <td class="py-4 px-6">
    <p v-if="item.cabang">{{ item.cabang.name }}</p>
    <p v-else class="text-red-500">terhapus</p>
  </td>

```

```

<td class="py-4 px-6">
  <button
    id="dropdownMenuIconButton"
    data-dropdown-toggle="dropdownDots"
    class="inline-flex items-center p-2 text-sm font-medium text-center text-gray-900 bg-white rounded-lg hover:bg-gray-100 focus:ring-4 focus:outline-none"
    type="button"
    @click="toggleDropdown(item.id)"
  >
    <svg
      class="w-6 h-6"
      aria-hidden="true"
      fill="currentColor"
      viewBox="0 0 20 20"
      xmlns="http://www.w3.org/2000/svg"
    >
      <path
        d="M10 6a2 2 0 11-4 2 2 0 010 4zM10 12a2 2 0 11-4 2 2 0 010 4zM10 18a2 2 0 11-4 2 2 0 010 4z"
      ></path>
    </svg>
  </button>

  <!-- Dropdown menu -->
  <div
    v-show="isHidden"
    refs="drop"
    id="dropdownDots"
    class="bg-white rounded divide-y divide-gray-100 shadow absolute"
    :class="isHidden === item.id ? 'show' : 'hidden'"
  >
    <ul
      class="py-1 text-sm text-gray-700"
      aria-labelledby="dropdownMenuIconButton"
    >
      <li>
        <Nuxt-Link
          :to="{
            name: 'cabang-id-user-edit',
            params: { item: item.id },

```

```

        }"
        class="block py-2 px-4 hover:bg-gray-100"
        >Edit
    </Nuxt-Link>
</li>
<li>
    <button
        href="#"
        class="block py-2 px-4 hover:bg-gray-100"
        @click="openModal(item.id, index), (msg = item.name)"
    >
        Delete
    </button>
</li>
</ul>
</div>
<!-- <div class="flex items-center">
    <NuxtLink :to="{name: ', params: { id: cabang_id }}" class="bg-
gray-100"
        tag="img"
        :src="require('~/static/assets/svg/dot3.svg')" >

    </Nuxtlink>
</div> -->
</td>
</tr>
</tbody>
</table>
<p v-if="$fetchState.pending">Fetching</p>
<nav
    v-else
    class="flex justify-between items-center pt-4"
    aria-label="Tabel navigation"
>
    <span class="text-sm font-normal text-gray-500"
    >Showing
    <span class="font-semibold text-gray-900">1-10</span>
    of
    <span class="font-semibold text-gray-900">{{
        user.data.result.last_page
    }}</span></span>
    >
    <ul class="inline-flex items-center -space-x-px">

```

```

</li>
  <button
    :value="user.data.result.current_page - 1"
    @click="updatePage"
    class="block py-2 px-3 ml-0 leading-tight text-gray-500 bg-white
rounded-l-lg border border-gray-300 hover:bg-gray-100 hover:text-gray-
700"
    :disabled="user.data.result.current_page == 1"
  >
    &lt;
  </button>
</li>
<p v-if="$fetchState.pending">Fetching roles...</p>

<li
  v-else
  v-for="(ada, index) in user.data.result.links"
  :key="index"
  >
  <button
    v-if="
      ada.label != '&laquo; Sebelumnya' &&
      ada.label != 'Berikutnya &raquo;'"
    :value="index"
    @click="updatePage"
    class="py-2 px-3 text-sm leading-tight text-gray-500 bg-white
border border-gray-300 hover:bg-gray-100 hover:text-gray-700"
  >
    {{ ada.label }}
  </button>
</li>

<li>
  <button
    :value="nextUrl"
    @click="updatePage"
    class="block py-2 px-3 leading-tight text-gray-500 bg-white
rounded-r-lg border border-gray-300 hover:bg-gray-100 hover:text-gray-
700"
    :disabled="pagenow == user.data.result.last_page"
  >
  >

```

```

        </button>
      </li>
    </ul>
  </nav>
</div>
</section>
</div>
</template>

<script>
import vClickOutside from 'v-click-outside'
import { mapActions } from 'vuex'
export default {
  layout: 'dashboard',
  middleware: 'auth',
  directives: {
    clickOutside: vClickOutside.directive,
  },
  data() {
    return {
      cabang_id: JSON.parse(localStorage.getItem('cabang_id')),
      role: JSON.parse(localStorage.getItem('role')),
      nextUrl: 2,
      prevUrl: null,
      keywords: null,
      pagenow: null,
      isHidden: false,
      rolename: ['Admin', 'Front Office', 'Back Office'],
      roleFilter: '0',
      user: {},
      hapus: 'Konfirmasi',
      modal: false,
      msg: "",
    }
  },
  computed: {
    sidebar() {
      return this.$store.state.sidebar
    },
  },
  watch: {
    keywords(after, before) {
      this.refetch()
    }
  }
}

```

```

    fetchOnServer: false
  },
  roleFilter() {
    console.log(this.roleFilter)
    this.$nuxt.refresh()
    this.pagenow = 1
    this.nextUrl = 2
    this.prevUrl = null
  },
},
mounted() {
  console.log(this.isHidden)
},
async fetch() {
  this.user = await this.$axios.get('/userall', {
    params: {
      search: this.keywords,
      limit: 10,
      page: this.pagenow,
      role: this.roleFilter,
    },
  })
},
methods: {
  ...mapActions(['toggleSidebar']),

  updatePage(event) {
    this.pagenow = parseInt(event.target.value)
    this.$nuxt.refresh()
    this.nextUrl = 1 + parseInt(event.target.value)
  },
  async refetch() {
    this.user = await this.$axios.get('/userall', {
      params: {
        limit: 10,
        page: this.pagenow,
        search: this.keywords,
        role: this.roleFilter,
      },
    })
  },
},

  async deletePost(item, index) {

```

```

//delete data post by ID
await this.$axios.delete(`/user/${item.id}`).then(() => {
  //remove item array by index
  this.user.data.result.data.splice(index, 1)
})
},

//Method for dots three
toggleDropdown(id) {
  if (this.isHidden === false) {
    this.isHidden = id
    this.$emit('change', this.isHidden)
  } else {
    this.isHidden = false
  }
},

async deletePost() {
  //delete data post by ID
  await this.$axios.delete(`/CoA/${this.id}`).then(() => {
    //remove item array by index
    // this.CoA.data.result.data.splice(this.index, 1)
    this.id = null
    this.index = null
    this.modal = !this.modal
    this.refetch()
  })
},

openModal(id, index) {
  this.id = id
  this.index = index
  this.modal = !this.modal
},

closeModal() {
  this.modal = false
  this.modalimport = false
  this.modalExport = false
},
},
}
</script>

```

createcabang.vue

Sourcecode 14 Createcabang.vue

```
<template>
  <section class="flex flex-col items-center justify-center px-4">
    <div class="text-[32px] font-semibold text-dark">Tambah
    Data</div>
    <p class="mt-4 text-base leading-7 text-center mb-[50px] text-
    grey">
      Cabang Untuk Login <br />
    </p>
    <form class="w-full card" @submit.prevent="createCabang">
      <p v-if="!listerror" class="m-3">
        <b>Tolong Di Cek Kembali :</b>
        <ul class="list-disc">
          <li v-for="error in errors" :key="error" class="text-red-500">
            {{ error[0] }}
          </li>
        </ul>
      </p>
      <div class="form-group">
        <label for="" class="text-grey">Nama Cabang</label>
        <input
          type="text"
          class="input-field"
          value="cabang.name"
          v-model="cabang.name"
          required
        />
      </div>

      <button type="submit" class="w-full btn btn-Lazismu mt-
      [20px]">
        Create
      </button>
    </form>
  </section>
</template>

<script>
export default {
  layout: 'form',
  // middleware: 'auth',
  data() {
    return {
```

```

    cabang: {
      name: "",
    },
    /listerror : false,
    errors: [],
    cabang_id: this.$route.params.id,
  }
},
methods: {
  async createCabang() {
    try {
      //send registration data to server
      let response = await this.$axios.post('/cabang', this.cabang)

      //Redirect to my cabang page
      this.$router.push({
        name: 'cabang-id-cabang',
        params: {
          id: this.cabang.cabang_id,
        },
      })

      console.log(response)
    } catch (error) {
      console.log(error)
    }
  },
},
}
</script>

```

createCoA.vue

Sourcecode 15 CreateCoA.vue

```

<template>
  <section class="flex flex-col items-center justify-center px-4">
    <div class="text-[32px] font-semibold text-dark">Tambah
    Data</div>
    <p class="mt-4 text-base leading-7 text-center mb-[50px] text-
    grey">
      CoA Kredit<i>(Chart of Account)</i> <br />
    </p>
    <form class="w-full card" @submit.prevent="createCoA">
      <p v-if="/listerror" class="m-3">

```

```

<b>Tolong Di Cek Kembali :</b>
<ul class="list-disc">
  <li v-for="error in errors" :key="error[0]" class="text-red-500">
    {{ error[0] }}
  </li>
</ul>
</p>
<div class="form-group">
  <label for="" class="text-grey">No Urut</label>
  <input
    type="text"
    class="input-field"
    v-model="CoA.kode"
    value="CoA.kode"
    required
  />
</div>
<div class="form-group">
  <label for="" class="text-grey">Nama Akun</label>
  <input
    type="text"
    class="input-field"
    value="CoA.name"
    v-model="CoA.name"
    required
  />
</div>
<div class="form-group">
  <label for="" class="text-grey">Tipe</label>
  <select
    name="tipe"
    id=""
    v-model="CoA.tipe"
    class="input-field rounded-lg block p-2.5 text-sm"
  >
    <option value="0">Debit</option>
    <option value="1">Kredit</option>
  </select>
</div>
<button type="submit" class="w-full btn btn-Lazismu mt-[14px]">

```

```

        Create
      </button>
    </form>
  </section>
</template>

<script>
export default {
  layout: 'form',
  // middleware: 'auth',
  data() {
    return {
      CoA: {
        name: "",
        kode: "",
        cabang_id: this.$route.params.id,
        tipe: "",
      },
      //listerror: false,
      errors: [],
    }
  },
  methods: {
    async createCoA() {
      try {
        //send registration data to server
        let response = await this.$axios.post('/CoA', this.CoA)

        //Redirect to my CoA page
        this.$router.push({
          name: 'cabang-id-CoA',
          params: {
            id: this.CoA.cabang_id,
          },
        })

        console.log(response)
      } catch (error) {
        console.log(error)
        this.errors = error.response.data.errors
        this.listerror = true
      }
    },
  },
}

```

```

    },
  }
</script>

```

createfrontoffice.vue

Sourcecode 16 Createfrontoffice.vue

```

<template>
  <section class="flex flex-col items-center justify-center px-4 mb-8">
    <div class="text-[24px] font-semibold text-dark">Tambah Data</div>
    <p class="mt-2 text-base leading-7 text-center mb-[50px] text-grey">
      Front Office <br />
    </p>
    <form class="w-full card-baru"
    @submit.prevent="createFrontoffice">
      <p v-if="listerror" class="m-3">
        <b>Tolong Di Cek Kembali :</b>
        <ul class="list-disc">
          <li v-for="error in errors" :key="error[0]" class="text-red-500">
            {{ error[0] }}
          </li>
        </ul>
      </p>
      <div class="grid grid-cols-3 gap-4">
        <div class="form-group">
          <label for="" class="text-grey">Tanggal</label>
          <input
            type="date"
            class="input-field2"
            v-model="frontoffice.tanggal"
          />
        </div>
        <div class="form-group col-span-2">
          <label for="" class="text-grey">No. bukti Penerima</label>
          <input
            type="text"
            class="input-field2"
            v-model="frontoffice.nobuktipenerima"
          />
        </div>
      </div>
    </form>
  </div>

```

```

</div>
<div class="form-group">
  <label for="" class="text-grey">Uraian Transaksi</label>
  <input type="text" class="input-field2" v-
model="frontoffice.name" />
</div>

<div class="grid grid-cols-2 gap-4">
  <div class="form-group">
    <label for="" class="text-grey">Muzakki</label>
    <input
      type="text"
      class="input-field2"
      v-model="frontoffice.penyetor"
    />
  </div>
  <div class="form-group">
    <label for="" class="text-grey">Penerima</label>
    <input
      type="text"
      class="input-field2"
      v-model="frontoffice.penerima"
    />
  </div>
</div>
<div class="form-group">
  <label for="" class="text-grey">Pilih Akun Debit</label>
  <p v-if="$fetchState.pending">Fetching debits...</p>
  <select
    v-else
    v-model="frontoffice.CoAdebit_id"
    name="debit"
    id=""
    class="appearance-none input-field2 form-icon-
chevron_down w-9/12"
  >
    <option
      :value="item.id"
      v-for="item in debit.data.result.data"
      :key="item"
    >
      {{ item.name }}
    </option>
  </select>

```

```

    </select>
  </div>
  <div class="form-group">
    <label for="" class="text-grey">Pilih Akun Kredit</label>
    <p v-if="$fetchState.pending">Fetching roles...</p>
    <select
      v-else
      v-model="frontoffice.CoAkredit_id"
      name="kredit"
      id=""
      class="appearance-none input-field2 form-icon-
chevron_down w-9/12"
    >
      <option
        :value="item.id"
        v-for="item in kredit.data.result.data"
        :key="item"
      >
        {{ item.name }}
      </option>
    </select>
  </div>
  <div class="grid grid-cols-3 gap-4">
    <div class="form-group">
      <label for="" class="text-grey">Jumlah</label>
      <input
        type="text"
        class="input-field2"
        v-model="frontoffice.jumlah"
      />
    </div>
    <div class="form-group">
      <label for="" class="text-grey">Tempat Bayar</label>
      <input
        type="text"
        class="input-field2"
        v-model="frontoffice.tempatbayar"
      />
    </div>
    <div class="form-group">
      <label for="" class="text-grey">Ref</label>
      <input type="text" class="input-field2" v-
model="frontoffice.ref" />

```

```

    </div>
  </div>

  <button type="submit" class="w-full btn btn-Lazismu mt-
[14px]">
    Create
  </button>
</form>
</section>
</template>

<script>
export default {
  layout: 'form',
  middleware: 'auth',
  data() {
    return {
      debit: [],
      kredit: [],
      frontoffice: {
        name: "",
        penyetor: "",
        penerima: "",
        nobuktipenerima: "",
        tanggal: "",
        ref: "",
        jumlah: "",
        tempatbayar: "",
        CoAdebit_id: "",
        CoAkredit_id: "",
        cabang_id: this.$route.params.id,
      },
      /listerror: false,
      errors: [],
      cabang_id: this.$route.params.id,
    }
  },
  async fetch() {
    ;(this.debit = await this.$axios.get('/CoA', {
      params: {
        limit: 120,
        cabang_id: this.cabang_id,
        tipe: false,

```



```

    },
  })),
  (this.kredit = await this.$axios.get('/CoA', {
    params: {
      limit: 120,
      cabang_id: this.cabang_id,
      tipe: 1,
    },
  })),
},
methods: {
  async createFrontoffice() {
    try {
      //send registration data to server
      let response = await this.$axios.post('/frontoffice',
this.frontoffice)

      //Redirect to my frontoffice page
      this.$router.push({
        name: 'cabang-id-frontoffices',
        params: {
          id: this.cabang_id,
        },
      })

      console.log(response)
    } catch (error) {
      this.errors = error.response.data.errors
      this.$toast.error = true
    }
  },
},
}
</script>

```

createkasbank.vue

Sourcecode 17 Createkasbank.vue

```

<template>
  <section class="flex flex-col items-center justify-center px-4 mb-4">
    <div class="text-[24px] font-semibold text-dark">Tambah Data</div>

```

```

<p class="mt-2 text-base leading-7 text-center mb-[50px] text-
grey">
  Kas Bank<br />
</p>

<div class="w-full card-baru">
  <div class="flex mb-8 justify-center">
    <button
      class="bg-red-700 text-white px-4 py-2 mr-0 rounded-l-lg
hover:bg-red-800"
      :class="{ 'bg-opacity-50': inputType }"
      @click.prevent="changeDataPengeluaran"
    >
      Pengeluaran
    </button>
    <button
      class="bg-green-700 text-white px-4 py-2 ml-0 rounded-r-lg
hover:bg-green-800"
      :class="{ 'bg-opacity-50': inputType }"
      @click="changeDataPemasukan"
    >
      Pemasukan
    </button>

  </div>
  <form class="w-full card-baru"
@submit.prevent="createKasbank">
    <p v-if="!listerror" class="m-3">
      <b>Tolong Di Cek Kembali :</b>
      <ul class="list-disc">
        <li v-for="error in errors" :key="error[0]" class="text-red-
500">
          {{ error[0] }}
        </li>
      </ul>
    </p>
    <div class="grid grid-cols-3 gap-4">
      <div class="form-group">
        <label for="" class="text-grey">Tanggal</label>
        <input
          type="date"
          class="input-field2"
          required

```

```

        v-model="kasbank.tanggal"
      />
    </div>
    <div class="form-group col-span-2">
      <label for="" class="text-grey">No. bukti Penerima</label>
      <input
        type="text"
        class="input-field2"
        v-model="kasbank.nobuktikas"
        required
      />
    </div>
  </div>
  <div class="form-group">
    <label for="" class="text-grey">Uraian Transaksi</label>
    <input type="text" class="input-field2" v-
model="kasbank.name" />
  </div>

  <div class="grid grid-cols-2 gap-4">
    <div class="form-group">
      <label for="" class="text-grey">Penerima</label>
      <input
        type="text"
        class="input-field2"
        v-model="kasbank.penerima"
      />
    </div>
  </div>
  <div class="form-group">
    <label for="" class="text-grey">Pilih Akun Debit</label>
    <p v-if="$fetchState.pending">Fetching debits...</p>
    <select
      v-else
      v-model="kasbank.CoAdebit_id"
      name="debit"
      id=""
      class="appearance-none input-field2 form-icon-
chevron_down w-9/12"
    >
      <option
        :value="item.id"
        v-for="item in debit.data.result.data"

```

```

        :key="'B' + item.id"
      >
        {{ item.name }}
      </option>
    </select>
  </div>
  <div class="form-group">
    <label for="" class="text-grey">Pilih Akun Kredit</label>
    <p v-if="$fetchState.pending">Fetching roles...</p>
    <select
      v-else
      v-model="kasbank.CoAkredit_id"
      name="kredit"
      id=""
      class="appearance-none input-field2 form-icon-
chevron_down w-9/12"
    >
      <option
        :value="item2.id"
        v-for="item2 in kredit.data.result.data"
        :key="'A' + item2.id"
      >
        {{ item2.name }}
      </option>
    </select>
  </div>
  <div class="grid grid-cols-2 gap-4">
    <div class="form-group">
      <label for="" class="text-grey">Jumlah</label>
      <input type="text" class="input-field2" v-
model="kasbank.jumlah" />
    </div>
    <div class="form-group">
      <label for="" class="text-grey">Ref</label>
      <input
        type="text"
        class="input-field2"
        v-model="kasbank.ref"
      />
    </div>
  </div>
</div>

```

```

    <button type="submit" class="w-full btn btn-Lazismu mt-
[14px]">
      Create
    </button>
  </form>
</div>
</section>
</template>

```

```

<script>
export default {
  layout: 'form',
  middleware: 'auth',
  data() {
    return {
      debit: [],
      kredit: [],
      temp: [],
      kasbank: {
        name: "",
        penerima: "",
        nobuktikas: "",
        tanggal: "",
        ref: "",
        jumlah: "",
        CoAdebit_id: "",
        CoAkredit_id: "",
        cabang_id: this.$route.params.id,
        jenis_data: 0,
      },
      cabang_id: this.$route.params.id,
      inputType: false,
      cabang: [],
      listerror: false,
      errors: [],
    }
  },
  async fetch() {
    ;(this.debit = await this.$axios.get('/CoA', {
      params: {
        limit: 100,
        cabang_id: this.cabang_id,
        tipe: false,

```

```

    },
  })),
  (this.kredit = await this.$axios.get('/CoA', {
    params: {
      limit: 100,
      cabang_id: this.cabang_id,
      tipe: 1,
    },
  })),
},
methods: {
  changeDataPengeluaran() {
    if (this.inputType === false) {
      console.log('ganti ke pengeluaran')
      this.kasbank.jenis_data = 1
      this.inputType = true
      this.temp = this.debit
      this.debit = this.kredit
      this.kredit = this.temp
    }
    console.log('tetap')
  },
  changeDataPemasukan() {
    if (this.inputType === true) {
      this.inputType = false
      this.kasbank.jenis_data = 0
      this.temp = this.debit
      this.debit = this.kredit
      this.kredit = this.temp
      console.log('ganti ke pemasukan')
    }
  },
  async createKasbank() {
    try {
      //send registration data to server
      let response = await this.$axios.post('/kasbank', this.kasbank)
      //Redirect to my kasbank page
      this.$router.push({
        name: 'cabang-id-kasbank',
        params: {
          id: this.cabang_id,
        },
      })
    }
  }
}

```

```

        console.log(response)
      } catch (error) {
        this.errors = error.response.data.errors
        this.$toast.error = true
      }
    },
  },
}
</script>

```

createkasbesar.vue

Sourcecode 18 Createkasbesar.vue

```

<template>
  <section class="flex flex-col items-center justify-center px-4 mb-4">
    <div class="text-[24px] font-semibold text-dark">Tambah
    Data</div>
    <p class="mt-2 text-base leading-7 text-center mb-[50px] text-
    grey">
      Kas Besar<br />
    </p>

    <div class="w-full card-baru">
      <div class="flex mb-8 justify-center">
        <button
          class="bg-green-700 text-white px-4 py-2 mr-0 rounded-l-lg
          hover:bg-green-800"
          :class="{ 'bg-opacity-50': !inputType }"
          @click.prevent="changeDataPemasukan"
        >
          Pemasukan
        </button>
        <button
          class="bg-red-700 text-white px-4 py-2 ml-0 rounded-r-lg
          hover:bg-red-800"
          :class="{ 'bg-opacity-50': inputType }"
          @click="changeDataPengeluaran"
        >
          Pengeluaran
        </button>

      </div>
      <form class="w-full card-baru"
        @submit.prevent="createKasbesar">

```

```

<p v-if="!isterror" class="m-3">
<b>Tolong Di Cek Kembali :</b>
<ul class="list-disc">
  <li v-for="error in errors" :key="error[0]" class="text-red-500">
    {{ error[0] }}
  </li>
</ul>
</p>
<div class="grid grid-cols-3 gap-4">
  <div class="form-group">
    <label for="" class="text-grey">Tanggal</label>
    <input
      type="date"
      class="input-field2"
      required
      v-model="kasbesar.tanggal"
    />
  </div>
  <div class="form-group col-span-2">
    <label for="" class="text-grey">No. bukti Penerima</label>
    <input
      type="text"
      class="input-field2"
      v-model="kasbesar.nobuktikas"
      required
    />
  </div>
</div>
<div class="form-group">
  <label for="" class="text-grey">Uraian Transaksi</label>
  <input type="text" class="input-field2" v-
model="kasbesar.name" />
</div>

<div class="grid grid-cols-2 gap-4">
  <div class="form-group">
    <label for="" class="text-grey">Penerima</label>
    <input
      type="text"
      class="input-field2"
      v-model="kasbesar.penerima"
    />
  </div>
</div>

```



```

</div>
<div class="form-group">
  <label for="" class="text-grey">Pilih Akun Debit</label>
  <p v-if="$fetchState.pending">Fetching debits...</p>
  <select
    v-else
    v-model="kasbesar.CoAdebit_id"
    name="debit"
    id=""
    class="appearance-none input-field2 form-icon-chevron_down
w-9/12"
  >
    <option
      :value="item.id"
      v-for="item in debit.data.result.data"
      :key="'B' + item.id"
    >
      {{ item.name }}
    </option>
  </select>
</div>
<div class="form-group">
  <label for="" class="text-grey">Pilih Akun Kredit</label>
  <p v-if="$fetchState.pending">Fetching roles...</p>
  <select
    v-else
    v-model="kasbesar.CoAkredit_id"
    name="kredit"
    id=""
    class="appearance-none input-field2 form-icon-chevron_down
w-9/12"
  >
    <option
      :value="item2.id"
      v-for="item2 in kredit.data.result.data"
      :key="'A' + item2.id"
    >
      {{ item2.name }}
    </option>
  </select>
</div>
<div class="grid grid-cols-2 gap-4">
  <div class="form-group">

```

```

        <label for="" class="text-grey">Jumlah</label>
        <input type="text" class="input-field2" v-
model="kasbesar.jumlah" />
    </div>
    <div class="form-group">
        <label for="" class="text-grey">Ref</label>
        <input
            type="text"
            class="input-field2"
            v-model="kasbesar.ref"
            disabled
        />
    </div>
</div>

    <button type="submit" class="w-full btn btn-Lazismu mt-[14px]">
        Create
    </button>
</form>
</div>
</section>
</template>

<script>
export default {
    layout: 'form',
    middleware: 'auth',
    data() {
        return {
            debit: [],
            kredit: [],
            temp: [],
            kasbesar: {
                name: "",
                penerima: "",
                nobuktikas: "",
                tanggal: "",
                ref: 'JKBB',
                jumlah: "",
                CoAdebit_id: "",
                CoAkredit_id: "",
                cabang_id: this.$route.params.id,
                jenis_data: 0,
            }
        }
    }
}

```

```

    },
    cabang_id: this.$route.params.id,
    inputType: false,
    cabang: [],
    listerror: false,
    errors: [],
  }
},
async fetch() {
;(this.debit = await this.$axios.get('/CoA', {
  params: {
    limit: 100,
    cabang_id: this.cabang_id,
    tipe: false,
  },
})),
(this.kredit = await this.$axios.get('/CoA', {
  params: {
    limit: 100,
    cabang_id: this.cabang_id,
    tipe: 1,
  },
})))
},
methods: {
  changeDataPengeluaran() {
    if (this.inputType === false) {
      console.log('ganti ke pengeluaran')
      this.kasbesar.jenis_data = 1
      this.inputType = true
      this.temp = this.debit
      this.debit = this.kredit
      this.kredit = this.temp
    }
    console.log('tetap')
  },
  changeDataPemasukan() {
    if (this.inputType === true) {
      this.inputType = false
      this.kasbesar.jenis_data = 0
      this.temp = this.debit
      this.debit = this.kredit
      this.kredit = this.temp
    }
  }
}
}

```

```

        console.log('ganti ke pemasukan')
      }
    },
    async createKasbesar() {
      try {
        //send registration data to server
        let response = await this.$axios.post('/kasbesar', this.kasbesar)
        //Redirect to my kasbesar page
        this.$router.push({
          name: 'cabang-id-kasbesar',
          params: {
            id: this.cabang_id,
          },
        })
        console.log(response)
      } catch (error) {
        this.errors = error.response.data.errors
        this.$listerror = true
      }
    },
  },
}
</script>

```

createkaskacil.vue

Sourcecode 19 Createkaskacil.vue

```

<template>
  <section class="flex flex-col items-center justify-center px-4 mb-4">
    <div class="text-[24px] font-semibold text-dark">Tambah
    Data</div>
    <p class="mt-2 text-base leading-7 text-center mb-[50px] text-
    grey">
      Kas Besar<br />
    </p>

    <div class="w-full card-baru">
      <div class="flex mb-8 justify-center">
        <button
          class="bg-green-700 text-white px-4 py-2 mr-0 rounded-l-lg
          hover:bg-green-800"
          :class="{ 'bg-opacity-50': !inputType }"
          @click.prevent="changeDataPemasukan"
        >

```

```

        Pemasukan
    </button>
    <button
        class="bg-red-700 text-white px-4 py-2 ml-0 rounded-r-lg
hover:bg-red-800"
        :class="{ 'bg-opacity-50': inputType }"
        @click="changeDataPengeluaran"
    >
        Pengeluaran
    </button>

</div>
<form class="w-full card-baru"
@submit.prevent="createKasbesar">
    <p v-if="!listerror" class="m-3">
    <b>Tolong Di Cek Kembali :</b>
    <ul class="list-disc">
        <li v-for="error in errors" :key="error[0]" class="text-red-500">
            {{ error[0] }}
        </li>
    </ul>
</p>
<div class="grid grid-cols-3 gap-4">
    <div class="form-group">
        <label for="" class="text-grey">Tanggal</label>
        <input
            type="date"
            class="input-field2"
            required
            v-model="kasbesar.tanggal"
        />
    </div>
    <div class="form-group col-span-2">
        <label for="" class="text-grey">No. bukti Penerima</label>
        <input
            type="text"
            class="input-field2"
            v-model="kasbesar.nobuktikas"
            required
        />
    </div>
</div>
<div class="form-group">

```

```

    <label for="" class="text-grey">Uraian Transaksi</label>
    <input type="text" class="input-field2" v-
model="kasbesar.name" />
  </div>

  <div class="grid grid-cols-2 gap-4">
    <div class="form-group">
      <label for="" class="text-grey">Penerima</label>
      <input
        type="text"
        class="input-field2"
        v-model="kasbesar.penerima"
      />
    </div>
  </div>
  <div class="form-group">
    <label for="" class="text-grey">Pilih Akun Debit</label>
    <p v-if="$fetchState.pending">Fetching debits...</p>
    <select
      v-else
      v-model="kasbesar.CoAdebit_id"
      name="debit"
      id=""
      class="appearance-none input-field2 form-icon-chevron_down
w-9/12"
    >
      <option
        :value="item.id"
        v-for="item in debit.data.result.data"
        :key="'B' + item.id"
      >
        {{ item.name }}
      </option>
    </select>
  </div>
  <div class="form-group">
    <label for="" class="text-grey">Pilih Akun Kredit</label>
    <p v-if="$fetchState.pending">Fetching roles...</p>
    <select
      v-else
      v-model="kasbesar.CoAkredit_id"
      name="kredit"
      id=""

```

```

        class="appearance-none input-field2 form-icon-chevron_down
w-9/12"
      >
        <option
          :value="item2.id"
          v-for="item2 in kredit.data.result.data"
          :key="'A' + item2.id"
        >
          {{ item2.name }}
        </option>
      </select>
    </div>
    <div class="grid grid-cols-2 gap-4">
      <div class="form-group">
        <label for="" class="text-grey">Jumlah</label>
        <input type="text" class="input-field2" v-
model="kasbesar.jumlah" />
      </div>
      <div class="form-group">
        <label for="" class="text-grey">Ref</label>
        <input
          type="text"
          class="input-field2"
          v-model="kasbesar.ref"
          disabled
        />
      </div>
    </div>

    <button type="submit" class="w-full btn btn-Lazismu mt-[14px]">
      Create
    </button>
  </form>
</div>
</section>
</template>

<script>
export default {
  layout: 'form',
  middleware: 'auth',
  data() {
    return {

```

```

    debit: [],
    kredit: [],
    temp: [],
    kasbesar: {
      name: "",
      penerima: "",
      nobuktikas: "",
      tanggal: "",
      ref: 'JKBB',
      jumlah: "",
      CoAdebit_id: "",
      CoAkredit_id: "",
      cabang_id: this.$route.params.id,
      jenis_data: 0,
    },
    cabang_id: this.$route.params.id,
    inputType: false,
    cabang: [],
    /listerror: false,
    errors: [],
  }
},
async fetch() {
  ;(this.debit = await this.$axios.get('/CoA', {
    params: {
      limit: 100,
      cabang_id: this.cabang_id,
      tipe: false,
    },
  })),
  (this.kredit = await this.$axios.get('/CoA', {
    params: {
      limit: 100,
      cabang_id: this.cabang_id,
      tipe: 1,
    },
  })),
},
methods: {
  changeDataPengeluaran() {
    if (this.inputType === false) {
      console.log('ganti ke pengeluaran')
      this.kasbesar.jenis_data = 1
    }
  }
}

```



```

        this.inputType = true
        this.temp = this.debit
        this.debit = this.kredit
        this.kredit = this.temp
    }
    console.log('tetap')
  },
  changeDataPemasukan() {
    if (this.inputType === true) {
      this.inputType = false
      this.kasbesar.jenis_data = 0
      this.temp = this.debit
      this.debit = this.kredit
      this.kredit = this.temp
      console.log('ganti ke pemasukan')
    }
  },
  async createKasbesar() {
    try {
      //send registration data to server
      let response = await this.$axios.post('/kasbesar', this.kasbesar)
      //Redirect to my kasbesar page
      this.$router.push({
        name: 'cabang-id-kasbesar',
        params: {
          id: this.cabang_id,
        },
      })
      console.log(response)
    } catch (error) {
      this.errors = error.response.data.errors
      this.$listerror = true
    }
  },
},
}
</script>

```

createuser.vue

Sourcecode 20 Createuser.vue

```

<template>
  <section class="flex flex-col items-center justify-center px-4">

```

```

<div class="text-[32px] font-semibold text-dark">Tambah
Data</div>
<p class="mt-4 text-base leading-7 text-center mb-[50px] text-
grey">
  User Registrasi <br />
</p>
<form class="w-full card-baru" @submit.prevent="createUser">
<p v-if="/listerror" class="m-3">
  <b>Tolong Di Cek Kembali :</b>
  <ul class="list-disc">
    <p class="text-red-500">
      {{ errors }}
    </p>
  </ul>
</p>

<div class="grid gap-4 grid-cols-3">
  <div class="form-group">
    <label for="" class="text-grey">Nama</label>
    <input
      type="text"
      class="input-field"
      value="user.name"
      v-model="user.name"
      required
    />
  </div>
  <div class="col-span-2">
    <div class="form-group">
      <label for="" class="text-grey">Email</label>
      <input
        type="email"
        class="input-field"
        v-model="user.email"
        value="user.email"
        required
      />
    </div>
  </div>
  <div class="form-group">
    <label for="" class="text-grey">Password</label>
    <input

```

```

        type="password"
        class="input-field"
        value="user.password"
        v-model="user.password"
        required
    />
</div>
<div class="form-group">
    <label for="" class="text-grey">Role</label>
    <select
        name="role"
        id=""
        v-model="user.role"
        class="input-field2 form-icon-chevron_down appearance-
none w-9/12"
        :value="user.role"
    >
        <option value="2">Front Office</option>
        <option value="3">Back Office</option>
    </select>
</div>
<div class="form-group">
    <label for="" class="text-grey">Pilih Cabang </label>
    <p v-if="$fetchState.pending">Fetching cabang...</p>
    <select
        v-else
        v-model="user.cabang_id"
        name="debit"
        id=""
        class="appearance-none input-field2 form-icon-
chevron_down w-9/12"
    >
        <option
            :value="item.id"
            v-for="item in cabang.data.result.data"
            :key="'B' + item.id"
        >
            {{ item.name }}
        </option>
    </select>
</div>
<button type="submit" class="w-full btn btn-Lazismu mt-
[14px]">

```

```

        Create
      </button>
    </form>
  </section>
</template>

<script>
export default {
  layout: 'form',
  // middleware: 'auth',
  data() {
    return {
      user: {
        name: "",
        email: "",
        password: "",
        cabang_id: this.$route.params.id,
        role: "",
      },
      cabang: [],
      listerror: false,
      errors: [],
    }
  },
  async fetch() {
    ;(this.cabang = await this.$axios.get('/cabang', {
      params: {
        limit: 100,
        cabang_id: this.cabang_id,
      },
    })))
  },
  watch: {
    errors(){
      this.listerror = true
    }
  },
  methods: {
    async createUser() {
      try {
        //send registration data to server
        let response = await this.$axios.post('/register', this.user)
      }
    }
  }
}

```

```

//Redirect to my user page
this.$router.push({
  name: 'cabang-id-user',
  params: {
    id: this.user.cabang_id,
  },
})

console.log(response)
} catch (error) {
  console.log(error.response.data.meta.message);
  this.errors = error.response.data.meta.message
}
},
}
}
</script>

```

Back-End

CreateNewUser.php

Sourcecode 21 CreateNewUser.php

```

<?php

namespace App\Actions\Fortify;

use App\Models\User;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;
use Laravel\Fortify\Contracts\CreatesNewUsers;
use Laravel\Jetstream\Jetstream;

class CreateNewUser implements CreatesNewUsers
{
    use PasswordValidationRules;

    /**
     * Validate and create a newly registered user.
     *
     * @param array $input
     * @return \App\Models\User
     */
    public function create(array $input)

```

```

    {
        Validator::make($input, [
            'name' => ['required', 'string', 'max:255'],
            'email' => ['required', 'string', 'email', 'max:255',
'unique:users'],
            'password' => $this->passwordRules(),
            'terms' => Jetstream::hasTermsAndPrivacyPolicyFeature() ?
['accepted', 'required'] : '',
        ]->validate());

        return User::create([
            'name' => $input['name'],
            'email' => $input['email'],
            'password' => Hash::make($input['password']),
        ]);
    }
}

```

UpdateUserProfile.php

Sourcecode 22 UpdateUserProfile.php

```

<?php

namespace App\Actions\Fortify;

use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Support\Facades\Validator;
use Illuminate\Validation\Rule;
use Laravel\Fortify\Contracts\UpdatesUserProfileInformation;

class UpdateUserProfileInformation implements
UpdatesUserProfileInformation
{
    /**
     * Validate and update the given user's profile information.
     *
     * @param mixed $user
     * @param array $input
     * @return void
     */
    public function update($user, array $input)
    {
        Validator::make($input, [
            'name' => ['required', 'string', 'max:255'],

```

```

        'email' => ['required', 'email', 'max:255',
Rule::unique('users')->ignore($user->id)],
        'photo' => ['nullable', 'mimes:jpg,jpeg,png', 'max:1024'],
    ]->validateWithBag('updateProfileInformation');

    if (isset($input['photo'])) {
        $user->updateProfilePhoto($input['photo']);
    }

    if ($input['email'] !== $user->email &&
        $user instanceof MustVerifyEmail) {
        $this->updateVerifiedUser($user, $input);
    } else {
        $user->forceFill([
            'name' => $input['name'],
            'email' => $input['email'],
        ]->save());
    }
}

/**
 * Update the given verified user's profile information.
 *
 * @param mixed $user
 * @param array $input
 * @return void
 */
protected function updateVerifiedUser($user, array $input)
{
    $user->forceFill([
        'name' => $input['name'],
        'email' => $input['email'],
        'email_verified_at' => null,
    ]->save());

    $user->sendEmailVerificationNotification();
}
}

```

CabangController.php

Sourcecode 23 CabangController.php

```
<?php
```

```

namespace App\Http\Controllers\API;

use Exception;
use App\Models\User;
use App\Models\Cabang;
use Illuminate\Http\Request;
use App\Helpers\ResponseFormatter;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Auth;
use App\Http\Requests\CreateCabangRequest;
use App\Http\Requests\UpdateCabangRequest;

class CabangController extends Controller
{
    public function fetch(Request $request)
    {
        $id = $request->input('id');
        $name = $request->input('name');
        $limit = $request->input('limit',10);

        // $cabangQuery = Cabang::with(['users'])->whereHas('users',
function ($query) {
    // $query->where('user_id', Auth::id());
    // });
    $cabangQuery = Cabang::with(['users']);

    //get single data
    if($id)
    {
        $cabang = $cabangQuery->find($id);

        if($cabang)
        {
            return ResponseFormatter::success($cabang);
        }
        return ResponseFormatter::error('Cabang not found');
    }

    //get multiple data
    $cabangs = $cabangQuery;

    if($name){

```



```

        $cabangs->where('name','like','%'.$name . '%');
    }
    return ResponseFormatter::success($cabangs-
>paginate($limit),'cabangs Found');
}

public function create(CreateCabangRequest $request)
{
    try {
        $cabang = Cabang::create([
            'name' => $request->name,
        ]);

        if(!$cabang)
        {
            throw new Exception('Cabang not created');
        }

        return ResponseFormatter::success($cabang, 'Cabang
created');
    } catch (Exception $e) {
        return ResponseFormatter::error($e->getMessage(), 500);
    }
}

public function update(UpdateCabangRequest $request, $id)
{
    try {
        $cabang = Cabang::find($id);

        if(!$cabang)
        {
            throw new Exception('Cabang not created');
        }

        //update cabang
        $cabang->update([
            'name' => $request->name,
        ]);

        return ResponseFormatter::success($cabang, 'Cabang
updated');
    }
}

```

```

        } catch (Exception $e) {
            return ResponseFormatter::error($e->getMessage(), 500);
        }
    }

    public function destroy($id)
    {
        try {
            //Get Cabang
            $cabang = Cabang::find($id);

            //check if cabang exists
            if (!$cabang) {
                throw new Exception('Cabang not Found');
            }

            //Delete Cabang
            $cabang->delete();

            return ResponseFormatter::success('Cabang deleted');
        } catch (Exception $e) {
            return ResponseFormatter::error($e->getMessage(),500);
        }
    }
}

```

CoAController.php

Sourcecode 24 CoAController.php

```

<?php

namespace App\Http\Controllers\API;

use App\Exports\CoAExport;
use Exception;
use App\Models\CoA;
use Illuminate\Http\Request;
use App\Helpers\ResponseFormatter;
use App\Http\Controllers\Controller;
use App\Http\Requests\CreateCoARequest;
use App\Http\Requests\UpdateCoARequest;
use App\Imports\CoAImport;
use Maatwebsite\Excel\Facades\Excel;

```

```

class CoAController extends Controller
{
  public function fetch(Request $request)
  {
    $id = $request->input('id');
    $name = $request->input('name');
    $limit = $request->input('limit',10);
    $cabang_id = $request->input('cabang_id');
    $tipe = $request->input('tipe');

    //get single data
    if($id)
    {
      $CoA = CoA::where('cabang_id',$request->cabang_id)-
>find($id);

      if($CoA)
      {
        return ResponseFormatter::success($CoA);
      }
      return ResponseFormatter::error('CoA not found');
    }
    //get multiple data
    $CoAs = CoA::query()->where('cabang_id',$request-
>cabang_id);

    if($name){
      $CoAs->where('name','like','%'.$name . '%');
    }
    if($tipe){
      $CoAs->where('tipe', intval($request->tipe));
    }

    return ResponseFormatter::success($CoAs-
>orderBy('name','asc')->paginate($limit),'CoAs Found');
  }

  public function create(CreateCoARequest $request)
  {
    try {
      $CoA = CoA::create([

```

```

        'name' => $request->name,
        'kode' => $request->kode,
        'cabang_id' => $request->cabang_id,
        'tipe' => $request->tipe,
    ]]);

    if(!$CoA)
    {
        throw new Exception('CoA not created');
    }

    return ResponseFormatter::success($CoA, 'CoA created');
} catch (Exception $e) {
    return ResponseFormatter::error($e->getMessage(), 500);
}
}

public function update(UpdateCoARequest $request, $id)
{
    try {
        $CoA = CoA::find($id);

        if(!$CoA)
        {
            throw new Exception('CoA not created');
        }

        //update CoA
        $CoA->update([
            'name' => $request->name,
            'kode' => $request->kode,
            'cabang_id' => $request->cabang_id,
            'tipe' => $request->tipe,
        ]]);

        return ResponseFormatter::success($CoA, 'CoA updated');
    } catch (Exception $e) {
        return ResponseFormatter::error($e->getMessage(), 500);
    }
}
}

```

```

public function destroy($id)
{
    try {
        //Get CoA
        $CoA = CoA::find($id);

        //check if CoA exists
        if (!$CoA) {
            throw new Exception('CoA not Found');
        }

        //Delete CoA
        $CoA->delete();

        return ResponseFormatter::success('CoA deleted');
    } catch (Exception $e) {
        return ResponseFormatter::error($e->getMessage(),500);
    }
}

public function importCoA(Request $request)
{
    $request->validate([
        'file' => 'required|mimes:xls,xlsx',
    ]);

    // import data
    Excel::import(new CoAImport, $request->file('file')-
>store('temp'));

    return ResponseFormatter::success('Import Success');
}

public function export(Request $request)
{
    $cabang_id = $request->input('cabang_id');
    return Excel::download(new CoAExport($cabang_id),
'kliens.xlsx');
}
}

```

CoAdebitController.php

Sourcecode 25 CoAdebitController.php

```
<?php
```

```

namespace App\Http\Controllers\API;

use Exception;
use App\Models\CoAdebit;
use Illuminate\Http\Request;
use App\Helpers\ResponseFormatter;
use App\Http\Controllers\Controller;
use App\Http\Requests\CreateCoAdebitRequest;
use App\Http\Requests\UpdateCoAdebitRequest;

class CoAdebitController extends Controller
{
    public function fetch(Request $request)
    {
        $id = $request->input('id');
        $name = $request->input('name');
        $cabang_id = $request->input('cabang_id');
        $limit = $request->input('limit',10);

        $CoAdebitQuery = CoAdebit::with(['cabang'])-
>where('cabang_id',$request->cabang_id);
        //get single data
        if($id)
        {
            $CoAdebit = $CoAdebitQuery->find($id);

            if($CoAdebit)
            {
                return ResponseFormatter::success($CoAdebit);
            }
            return ResponseFormatter::error('CoAdebit not found');
        }
        //get multiple data
        $CoAdebits = $CoAdebitQuery;

        if($name){
            $CoAdebits->where('name','like','%'.$name . '%');
        }
        return ResponseFormatter::success($CoAdebits-
>paginate($limit),'CoAdebits Found');
    }
}

```

```

public function create(CreateCoAdebitRequest $request)
{
    try {
        $CoAdebit = CoAdebit::create([
            'name' => $request->name,
            'kode' => $request->kode,
            'cabang_id' => $request->cabang_id,
        ]);

        if(!$CoAdebit)
        {
            throw new Exception('CoAdebit not created');
        }

        return ResponseFormatter::success($CoAdebit, 'CoAdebit
created');
    } catch (Exception $e) {
        return ResponseFormatter::error($e->getMessage(), 500);
    }
}

public function update(UpdateCoAdebitRequest $request, $id)
{
    try {
        $CoAdebit = CoAdebit::find($id);

        if(!$CoAdebit)
        {
            throw new Exception('CoAdebit not created');
        }

        //update CoAdebit
        $CoAdebit->update([
            'name' => $request->name,
            'kode' => $request->kode,
            'cabang_id' => $request->cabang_id,
        ]);

        return ResponseFormatter::success($CoAdebit, 'CoAdebit
updated');
    }
}

```

```

        } catch (Exception $e) {
            return ResponseFormatter::error($e->getMessage(), 500);
        }
    }

    public function destroy($id)
    {
        try {
            //Get CoAdebit
            $CoAdebit = CoAdebit::find($id);

            //check if CoAdebit exists
            if (!$CoAdebit) {
                throw new Exception('CoAdebit not Found');
            }

            //Delete CoAdebit
            $CoAdebit->delete();

            return ResponseFormatter::success('CoAdebit deleted');
        } catch (Exception $e) {
            return ResponseFormatter::error($e->getMessage(),500);
        }
    }
}

```

CoAKreditController.php

Sourcecode 26 CoAKreditController.php

```

<?php

namespace App\Http\Controllers\API;

use Exception;
use App\Models\CoAkredit;
use Illuminate\Http\Request;
use App\Helpers\ResponseFormatter;
use App\Http\Controllers\Controller;

```



```

use App\Http\Requests\CreateCoAkreditRequest;
use App\Http\Requests\UpdateCoAkreditRequest;
use App\Imports\CoAkreditImport;
use Maatwebsite\Excel\Facades\Excel;

class CoAkreditController extends Controller
{
    public function fetch(Request $request)
    {
        $id = $request->input('id');
        $name = $request->input('name');
        $limit = $request->input('limit',10);
        $cabang_id = $request->input('cabang_id');

        //get single data
        if($id)
        {
            $CoAkredit = CoAkredit::where('cabang_id',$request->cabang_id)->find($id);

            if($CoAkredit)
            {
                return ResponseFormatter::success($CoAkredit);
            }
            return ResponseFormatter::error('CoAkredit not found');
        }
        //get multiple data
        $CoAkredits = CoAkredit::query()->where('cabang_id',$request->cabang_id);

        if($name){
            $CoAkredits->where('name','like','%'.$name . '%')->where('cabang_id',$request->cabang_id);
        }
        return ResponseFormatter::success($CoAkredits->paginate($limit),'CoAkredits Found');
    }

    public function create(CreateCoAkreditRequest $request)
    {
        try {
            $CoAkredit = CoAkredit::create([
                'name' => $request->name,

```

```

        'kode' => $request->kode,
        'cabang_id' => $request->cabang_id,
    ]]);

    if(!$CoAkredit)
    {
        throw new Exception('CoAkredit not created');
    }

    return ResponseFormatter::success($CoAkredit, 'CoAkredit
created');
    } catch (Exception $e) {
        return ResponseFormatter::error($e->getMessage(), 500);
    }
}

public function update(UpdateCoAkreditRequest $request, $id)
{
    try {
        $CoAkredit = CoAkredit::find($id);

        if(!$CoAkredit)
        {
            throw new Exception('CoAkredit not created');
        }

        //update CoAkredit
        $CoAkredit->update([
            'name' => $request->name,
            'kode' => $request->kode,
            'cabang_id' => $request->cabang_id

        ]]);

        return ResponseFormatter::success($CoAkredit, 'CoAkredit
updated');

    } catch (Exception $e) {
        return ResponseFormatter::error($e->getMessage(), 500);
    }
}

```

```

public function destroy($id)
{
    try {
        //Get CoAkredit
        $CoAkredit = CoAkredit::find($id);

        //check if CoAkredit exists
        if (!$CoAkredit) {
            throw new Exception('CoAkredit not Found');
        }

        //Delete CoAkredit
        $CoAkredit->delete();

        return ResponseFormatter::success('CoAkredit deleted');
    } catch (Exception $e) {
        return ResponseFormatter::error($e->getMessage(),500);
    }
}

public function import_excel(Request $request)
{
    // import data
    Excel::import(new CoAkreditImport, $request->file('file')->store('temp'));

    return ResponseFormatter::success('Import Success');
}
}

```

FrontOfficeController.php

Sourcecode 27 FrontOfficeController.php

```

<?php

namespace App\Http\Controllers\API;

use Exception;
use App\Models\Frontoffice;
use Illuminate\Http\Request;
use App\Exports\FrontofficeExport;
use App\Helpers\ResponseFormatter;

```

```

use App\Http\Controllers\Controller;
use Maatwebsite\Excel\Facades\Excel;
use App\Http\Requests\CreateFrontofficeRequest;
use App\Http\Requests\UpdateFrontofficeRequest;

class FrontofficeController extends Controller
{
    public function fetch(Request $request)
    {
        $id = $request->input('id');
        $name = $request->input('name');
        $penyetor = $request->input('penyetor');
        $penerima = $request->input('penerima');
        $nobuktipenerima = $request->input('nobuktipenerima');
        $tanggal = $request->input('tanggal');
        $from = $request->input('from');
        $to = $request->input('to');
        $tempatbayar = $request->input('tempatbayar');
        $CoAdebit_id = $request->input('CoAdebit_id');
        $CoAkredit_id = $request->input('CoAkredit_id');
        $cabang_id = $request->input('cabang_id');
        $limit = $request->input('limit');
        $search = $request->input('search');
        $jenis_data = $request->input('jenis_data');

        $frontofficeQuery =
Frontoffice::with(['CoAdebit','CoAkredit'])-
>where('cabang_id',$cabang_id);

        //get single data
        if($id)
        {
            $frontoffice = $frontofficeQuery->find($id);

            if($frontoffice)
            {
                return ResponseFormatter::success($frontoffice,
"Frontoffice found");
            }
            return ResponseFormatter::error('Frontoffice not found');
        }

        //get multiple data

```

```

$frontoffices = $frontofficeQuery;

if($search){
    $frontoffices->where('name','like','%'.$search.'%')-
>orWhere('nobuktipenerima','like','%'.$search.'%');
}

if($penyeter){
    $frontoffices->where('penyeter',$penyeter);
}

if($penerima){
    $frontoffices->where('penerima',$penerima);
}
// if($nobuktipenerima){
//     $frontoffices-
>where('nobuktipenerima','like','%'.$nobuktipenerima.'%');
// }
if($tanggal){
    $frontoffices->where('tanggal', $tanggal);
}
if($tempatbayar){
    $frontoffices->where('tempatbayar',$tempatbayar);
}
if($CoAdebit_id){
    $frontoffices->where('CoAdebit_id',$CoAdebit_id);
}
if($CoAkredit_id){
    $frontoffices->where('CoAkredit_id',$CoAkredit_id);
}

if($from && $to){
    $frontoffices->whereBetween('tanggal', [$from, $to]);
}
if($limit)
{
    return ResponseFormatter::success($frontoffices-
>orderBy('tanggal','desc')->paginate($limit),'Frontoffices Found');
}

return ResponseFormatter::success($frontoffices-
>get(),'Frontoffices Found');
}

```

```

    public function create(CreateFrontofficeRequest $request,
Frontoffice $fo)
    {
        try {
            $frontoffice = Frontoffice::create([
                'name' => $request->name,
                'penyetor' => $request->penyetor,
                'penerima' => $request->penerima,
                'nobuktipenerima' => $request->nobuktipenerima,
                'tanggal' => $request->tanggal,
                'ref' => $request->ref,
                'jumlah' => $request->jumlah,
                'tempatbayar' => $request->tempatbayar,
                'CoAdebit_id' => $request->CoAdebit_id,
                'CoAkredit_id' => $request->CoAkredit_id,
                'cabang_id' => $request->cabang_id,

                ]);

            if(!$frontoffice)
            {
                throw new Exception('Frontoffice not created');
            }

            return ResponseFormatter::success($frontoffice,
'Frontoffice created');
        } catch (Exception $e) {
            return ResponseFormatter::error($e->getMessage(), 500);
        }
    }

    public function update(UpdateFrontofficeRequest $request,
$id)
    {
        try {
            $frontoffice = Frontoffice::find($id);

            if(!$frontoffice)
            {

```

```

        throw new Exception('Frontoffice not created');
    }

    //update frontoffice
    $frontoffice->update([
        'name' => $request->name,
        'penyetor' => $request->penyetor,
        'penerima' => $request->penerima,
        'nobuktipenerima' => $request->nobuktipenerima,
        'tanggal' => $request->tanggal,
        'ref' => $request->ref,
        'jumlah' => $request->jumlah,
        'tempatbayar' => $request->tempatbayar,
        'CoAdebit_id' => $request->CoAdebit_id,
        'CoAkredit_id' => $request->CoAkredit_id,
        'jenis_data' => $request->jenis_data

    ]);

    return ResponseFormatter::success($frontoffice,
'Frontoffice updated');

    } catch (Exception $e) {
        return ResponseFormatter::error($e->getMessage(), 500);
    }
}

public function destroy($id)
{
    try {
        //Get Frontoffice
        $frontoffice = Frontoffice::find($id);

        //check if frontoffice exists
        if (!$frontoffice) {
            throw new Exception('Frontoffice not Found');
        }

        //Delete Frontoffice
        $frontoffice->delete();

        return ResponseFormatter::success('Frontoffice deleted');
    } catch (Exception $e) {

```

```

        return ResponseFormatter::error($e->getMessage(),500);
    }
}

public function export(Request $request)
{
    $cabang_id = $request->input('cabang_id');
    $from = $request->input('from');
    $to = $request->input('to');

    return Excel::download(new FrontofficeExport($cabang_id,
    $from, $to), 'FrontOffice.xlsx');

}
}

```

HitungController.php

Sourcecode 28 HitungController.php

```

<?php

namespace App\Http\Controllers\API;

use App\Models\CoA;
use App\Models\Kasbesar;
use App\Models\Frontoffice;
use Illuminate\Http\Request;
use Illuminate\Support\Carbon;
use App\Helpers\ResponseFormatter;
use Illuminate\Support\Facades\DB;
use App\Http\Controllers\Controller;
use App\Models\Jurnalumum;
use App\Models\Kasbank;
use App\Models\Kaskecil;

class HitungController extends Controller
{

    public function statisAll(Request $request)
    {
        $cabang_id = $request->input('cabang_id');
        $menu = $request->input('menu');

        $month = Carbon::now()->addMonth(0)->format('m');
    }
}

```



```

$year = Carbon::now()->format('Y');

switch ($menu) {
    case 'kasbesar':
        $pemasukan = Kasbesar::where('cabang_id',
        $cabang_id)->whereYear('tanggal', $year)-
        >whereMonth('tanggal', $month)->where('jenis_data', 0)-
        >orderBy('tanggal', 'ASC')->get();
        $pengeluaran = Kasbesar::where('cabang_id',
        $cabang_id)->whereYear('tanggal', $year)-
        >whereMonth('tanggal', $month)->where('jenis_data', 1)-
        >orderBy('tanggal', 'ASC')->get();
        break;
    case 'frontoffice':
        $pemasukan = Frontoffice::where('cabang_id',
        $cabang_id)->whereYear('tanggal', $year)-
        >whereMonth('tanggal', $month)->orderBy('tanggal', 'ASC')-
        >get();
        $pengeluaran = [];
        break;
    case 'kaskecil':
        $pemasukan = Kaskecil::where('cabang_id', $cabang_id)-
        >whereYear('tanggal', $year)->whereMonth('tanggal', $month)-
        >orderBy('tanggal', 'ASC')->get();
        $pengeluaran = Kaskecil::where('cabang_id',
        $cabang_id)->whereYear('tanggal', $year)-
        >whereMonth('tanggal', $month)->where('jenis_data', 1)-
        >orderBy('tanggal', 'ASC')->get();
        break;
    case 'kasbank':
        $pemasukan = Kasbank::where('cabang_id', $cabang_id)-
        >whereYear('tanggal', $year)->whereMonth('tanggal', $month)-
        >orderBy('tanggal', 'ASC')->get();
        $pengeluaran = Kasbank::where('cabang_id',
        $cabang_id)->whereYear('tanggal', $year)-
        >whereMonth('tanggal', $month)->where('jenis_data', 1)-
        >orderBy('tanggal', 'ASC')->get();
        break;
    case 'jurnalumum':
        $pemasukan = Jurnalumum::where('cabang_id',
        $cabang_id)->whereYear('tanggal', $year)-

```

```

>whereMonth('tanggal', $month)->orderBy('tanggal', 'ASC')-
>get();
    $pengeluaran = [];
    break;

    default:
        # code...
        break;
    }

    // $pemasukan = $statis->where('jenis_data', 0)->get();
    // $pengeluaran = $statis->where('jenis_data', 1)->get();

    return [
        'kasbesarIn' => $pemasukan,
        'kasbesarOut' => $pengeluaran,
    ];
}
public function CoA(Request $request)
{
    $cabang_id = $request->input('cabang_id');

    return [
        'CoAkredit' => CoA::where('tipe', 1)->where('cabang_id',
        $cabang_id)->count(),
        'CoAdebit' => CoA::where('tipe', false)->where('cabang_id',
        $cabang_id)->count()
    ];
}

public function fo(Request $request)
{
    $cabang_id = $request->input('cabang_id');

    $frontoffices = DB::tabel('frontoffices')->where('cabang_id',
    $cabang_id)->sum('jumlah');

    return ResponseFormatter::success($frontoffices,
    'Frontoffices Found');
}

```

```

public function kasbesar(Request $request)
{
    $id = $request->input('id');
    $cabang_id = $request->input('cabang_id');
    $kasbesarpemasukan = DB::tabel('kasbesars')-
>where('cabang_id', $cabang_id)->where('jenis_data', 0)-
>sum('jumlah');
    $kasbesarpengeluaran = DB::tabel('kasbesars')-
>where('cabang_id', $cabang_id)->where('jenis_data', 1)-
>sum('jumlah');

    $total = $kasbesarpemasukan - $kasbesarpengeluaran;

    return [
        'pemasukan' => $kasbesarpemasukan,
        'pengeluaran' => $kasbesarpengeluaran,
        'total' => $total,
    ];
}

public function kaskecil(Request $request)
{
    $id = $request->input('id');
    $cabang_id = $request->input('cabang_id');
    $kaskecilpemasukan = DB::tabel('kaskecils')-
>where('cabang_id', $cabang_id)->where('jenis_data', 0)-
>sum('jumlah');
    $kaskecilpengeluaran = DB::tabel('kaskecils')-
>where('cabang_id', $cabang_id)->where('jenis_data', 1)-
>sum('jumlah');

    $total = $kaskecilpemasukan - $kaskecilpengeluaran;

    return [
        'pemasukan' => $kaskecilpemasukan,
        'pengeluaran' => $kaskecilpengeluaran,
        'total' => $total,
    ];
}

public function kasbank(Request $request)
{
    $id = $request->input('id');
    $cabang_id = $request->input('cabang_id');

```

```

        $kasbankpemasukan = DB::tabel('kasbanks')-
        >where('cabang_id', $cabang_id)->where('jenis_data', 0)-
        >sum('jumlah');
        $kasbankpengeluaran = DB::tabel('kasbanks')-
        >where('cabang_id', $cabang_id)->where('jenis_data', 1)-
        >sum('jumlah');

        $total = $kasbankpemasukan - $kasbankpengeluaran;

        return [
            'pemasukan' => $kasbankpemasukan,
            'pengeluaran' => $kasbankpengeluaran,
            'total' => $total,
        ];
    }
    public function jurnalumum(Request $request)
    {
        $cabang_id = $request->input('cabang_id');

        $jurnalumums = DB::tabel('jurnalumums')-
        >where('cabang_id', $cabang_id)->sum('jumlah');
        return ResponseFormatter::success($jurnalumums, 'Hitung
        Jurnal Umum Found');
    }
}

```

KasBankController.php

Sourcecode 29 KasBankController.php

```

<?php

namespace App\Http\Controllers\API;

use App\Exports\KasbankExport;
use Exception;
use App\Models\Kasbank;
use Illuminate\Http\Request;
use App\Exports\KasbesarExport;
use App\Helpers\ResponseFormatter;
use App\Http\Controllers\Controller;
use Maatwebsite\Excel\Facades\Excel;
use App\Http\Requests\CreateKasbankRequest;
use App\Http\Requests\UpdateKasbankRequest;

```

```

class KasbankController extends Controller
{
    public function fetch(Request $request)
    {
        $id = $request->input('id');
        $name = $request->input('name');
        $penerima = $request->input('penerima');
        $nobuktikas = $request->input('nobuktikas');
        $tanggal = $request->input('tanggal');
        $from = $request->input('from');
        $to = $request->input('to');
        $CoAdebit_id = $request->input('CoAdebit_id');
        $CoAkredit_id = $request->input('CoAkredit_id');
        $limit = $request->input('limit',10);
        $cabang_id = $request->input('cabang_id');
        $search = $request->input('search');

        $kasbankQuery = Kasbank::with(['CoAdebit','CoAkredit'])-
>where('cabang_id', $cabang_id);

        //get single data
        if($id)
        {
            $kasbank = $kasbankQuery->find($id);

            if($kasbank)
            {
                return ResponseFormatter::success($kasbank, "Kasbank
found");
            }
            return ResponseFormatter::error('Kasbank not found');
        }

        //get multiple data
        $kasbank = $kasbankQuery;

        if($search || ($from && $to)){
            $kasbank->where(function ($query) use ($search, $from,
$to) {
                $query->where('name', 'like', '%' . $search . '%')
                ->orWhere('nobuktikas', 'like', '%' . $search . '%');

                if ($from && $to) {

```

```

        $query->whereBetween('tanggal', [$from, $to]);
    }
});

}

if($name){
    $kasbank->where('name','like','%'.$name . '%');
}
if($penerima){
    $kasbank->where('penerima',$penerima);
}
if($nobuktikas){
    $kasbank->where('nobuktikas','like','%'.$nobuktikas . '%');
}
if($tanggal){
    $kasbank->where('tanggal', $tanggal);
}
if($CoAdebit_id){
    $kasbank->where('CoAdebit_id',$CoAdebit_id);
}
if($CoAkredit_id){
    $kasbank->where('CoAkredit_id',$CoAkredit_id);
}
if($from && $to){
    $kasbank->whereBetween('tanggal', [$from, $to]);
}

return ResponseFormatter::success($kasbank-
>orderBy('tanggal','desc')->paginate($limit,'Kasbank Found');
}

public function create(CreateKasbankRequest $request)
{
    try {
        $kasbank = Kasbank::create([
            'name' => $request->name,
            'penerima' => $request->penerima,
            'nobuktikas' => $request->nobuktikas,
            'tanggal' => $request->tanggal,
            'ref' => $request->ref,
            'jumlah' => $request->jumlah,

```

```

        'CoAdebit_id' => $request->CoAdebit_id,
        'CoAkredit_id' => $request->CoAkredit_id,
        'cabang_id' => $request->cabang_id,
        'jenis_data' => $request->jenis_data
    ]]);

    if(!$kasbank)
    {
        throw new Exception('Kasbank not created');
    }

    return ResponseFormatter::success($kasbank, 'Kasbank
created');
    } catch (Exception $e) {
        return ResponseFormatter::error($e->getMessage(), 500);
    }
}

public function update(UpdateKasbankRequest $request, $id)
{
    try {
        $kasbank = Kasbank::find($id);

        if(!$kasbank)
        {
            throw new Exception('Kasbank not created');
        }

        //update kasbank
        $kasbank->update([
            'name' => $request->name,
            'penerima' => $request->penerima,
            'nobuktikas' => $request->nobuktikas,
            'tanggal' => $request->tanggal,
            'ref' => $request->ref,
            'jumlah' => $request->jumlah,
            'CoAdebit_id' => $request->CoAdebit_id,
            'CoAkredit_id' => $request->CoAkredit_id,
            'cabang_id' => $request->cabang_id,
            'jenis_data' => $request->jenis_data
        ]]);
    }
}

```

```

        return ResponseFormatter::success($kasbank, 'Kasbank
        updated');

        } catch (Exception $e) {
            return ResponseFormatter::error($e->getMessage(), 500);
        }
    }

    public function destroy($id)
    {
        try {
            //Get Kasbank
            $kasbank = Kasbank::find($id);

            //check if kasbank exists
            if (!$kasbank) {
                throw new Exception('Kasbank not Found');
            }

            //Delete Kasbank
            $kasbank->delete();

            return ResponseFormatter::success('Kasbank deleted');
        } catch (Exception $e) {
            return ResponseFormatter::error($e->getMessage(),500);
        }
    }

    public function export(Request $request){
        $cabang_id = $request->input('cabang_id');
        $from = $request->input('from');
        $to = $request->input('to');
        return Excel::download(new KasbankExport($cabang_id,
        $from, $to), 'kasbank.xlsx');
    }
}

```

KasbesarController.php

Sourcecode 30 KasbesarController.php

```

<?php

namespace App\Http\Controllers\API;

use Exception;

```



```

use App\Models\Kasbesar;
use Illuminate\Http\Request;
use App\Exports\KasbesarExport;
use App\Helpers\ResponseFormatter;
use App\Http\Controllers\Controller;
use Maatwebsite\Excel\Facades\Excel;
use App\Http\Requests\CreateKasbesarRequest;
use App\Http\Requests\UpdateKasbesarRequest;

class KasbesarController extends Controller
{
    public function fetch(Request $request)
    {
        $id = $request->input('id');
        $name = $request->input('name');
        $penerima = $request->input('penerima');
        $nobuktikas = $request->input('nobuktikas');
        $tanggal = $request->input('tanggal');
        $from = $request->input('from');
        $to = $request->input('to');
        $CoAdebit_id = $request->input('CoAdebit_id');
        $CoAkredit_id = $request->input('CoAkredit_id');
        $limit = $request->input('limit',10);
        $cabang_id = $request->input('cabang_id');
        $search = $request->input('search');
        $jenis_data = $request->input('jenis_data');

        $kasbesarQuery = Kasbesar::with(['CoAdebit','CoAkredit'])-
        >where('cabang_id', $cabang_id);

        //get single data
        if($id)
        {
            $kasbesar = $kasbesarQuery->find($id);

            if($kasbesar)
            {
                return ResponseFormatter::success($kasbesar, "Kasbesar
found");
            }
            return ResponseFormatter::error('Kasbesar not found');
        }
    }
}

```

```

//get multiple data
$kasbesars = $kasbesarQuery;

if($search || ($from && $to)){
    $kasbesars->where(function ($query) use ($search, $from,
    $to) {
        $query->where('name', 'like', '%' . $search . '%')
        ->orWhere('nobuktikas', 'like', '%' . $search . '%');

        if ($from && $to) {
            $query->whereBetween('tanggal', [$from, $to]);
        }
    });
}

if($penerima){
    $kasbesars->where('penerima',$penerima);
}
if($nobuktikas){
    $kasbesars->where('nobuktikas','like','%'.$nobuktikas . '%');
}
if($tanggal){
    $kasbesars->where('tanggal', $tanggal);
}
if($CoAdebit_id){
    $kasbesars->where('CoAdebit_id',$CoAdebit_id);
}
if($CoAkredit_id){
    $kasbesars->where('CoAkredit_id',$CoAkredit_id);
}
if($cabang_id){
    $kasbesars->where('cabang_id',$cabang_id);
}
if($from && $to){
    $kasbesars->whereBetween('tanggal', [$from, $to]);
}

return ResponseFormatter::success($kasbesars-
>orderBy('tanggal','desc')->paginate($limit),'Kasbesar Found');

```

```

    }

    public function create(CreateKasbesarRequest $request)
    {
        try {
            $kasbesar = Kasbesar::create([
                'name' => $request->name,
                'penerima' => $request->penerima,
                'nobuktikas' => $request->nobuktikas,
                'tanggal' => $request->tanggal,
                'ref' => $request->ref,
                'jumlah' => $request->jumlah,
                'CoAdebit_id' => $request->CoAdebit_id,
                'CoAkredit_id' => $request->CoAkredit_id,
                'cabang_id' => $request->cabang_id,
                'jenis_data' => $request->jenis_data

            ]);

            if(!$kasbesar)
            {
                throw new Exception('Kasbesar not created');
            }

            return ResponseFormatter::success($kasbesar, 'Kasbesar
created');
        } catch (Exception $e) {
            return ResponseFormatter::error($e->getMessage(), 500);
        }
    }

    public function update(UpdateKasbesarRequest $request, $id)
    {
        try {
            $kasbesar = Kasbesar::find($id);

            if(!$kasbesar)
            {
                throw new Exception('Kasbesar not created');
            }
        }
    }

```

```

//update kasbesar
$kasbesar->update([
    'name' => $request->name,
    'penerima' => $request->penerima,
    'nobuktikas' => $request->nobuktikas,
    'tanggal' => $request->tanggal,
    'ref' => $request->ref,
    'jumlah' => $request->jumlah,
    'CoAdebit_id' => $request->CoAdebit_id,
    'CoAkredit_id' => $request->CoAkredit_id,
    'cabang_id' => $request->cabang_id,
    'jenis_data' => $request->jenis_data,
]);

return ResponseFormatter::success($kasbesar, 'Kasbesar
updated');

} catch (Exception $e) {
    return ResponseFormatter::error($e->getMessage(), 500);
}
}

public function destroy($id)
{
    try {
        //Get Kasbesar
        $kasbesar = Kasbesar::find($id);

        //check if kasbesar exists
        if (!$kasbesar) {
            throw new Exception('Kasbesar not Found');
        }

        //Delete Kasbesar
        $kasbesar->delete();

        return ResponseFormatter::success('Kasbesar deleted');
    } catch (Exception $e) {
        return ResponseFormatter::error($e->getMessage(),500);
    }
}

public function export(Request $request)

```

```

    {
        $cabang_id = $request->input('cabang_id');
        $from = $request->input('from');
        $to = $request->input('to');
        return Excel::download(new KasbesarExport($cabang_id,
        $from, $to), 'kasbesar.xlsx');
    }
}

```

KaskecilController.php

Sourcecode 31 KaskecilController.php

```

<?php

namespace App\Http\Controllers\API;

use App\Exports\KaskecilExport;
use Exception;
use App\Models\Kaskecil;
use Illuminate\Http\Request;
use App\Helpers\ResponseFormatter;
use App\Http\Controllers\Controller;
use Maatwebsite\Excel\Facades\Excel;
use App\Http\Requests\CreateKaskecilRequest;
use App\Http\Requests\UpdateKaskecilRequest;

class KaskecilController extends Controller
{
    public function fetch(Request $request)
    {
        $id = $request->input('id');
        $name = $request->input('name');
        $penerima = $request->input('penerima');
        $nobuktikas = $request->input('nobuktikas');
        $tanggal = $request->input('tanggal');
        $from = $request->input('from');
        $to = $request->input('to');
        $CoAdebit_id = $request->input('CoAdebit_id');
        $CoAkredit_id = $request->input('CoAkredit_id');
        $limit = $request->input('limit',10);
        $search = $request -> input('search');
        $cabang_id = $request->input('cabang_id');
    }
}

```

```

    $kaskecilQuery = Kaskecil::with(['CoAdebit','CoAkredit'])-
>where('cabang_id', $cabang_id);

    //get single data
    if($id)
    {
        $kaskecil = $kaskecilQuery->find($id);

        if($kaskecil)
        {
            return ResponseFormatter::success($kaskecil, "Kaskecil found");
        }
        return ResponseFormatter::error('Kaskecil not found');
    }

    //get multiple data
    $kaskecil = $kaskecilQuery;

    if($search || ($from && $to)){
        $kaskecil->where(function ($query) use ($search, $from, $to) {
            $query->where('name', 'like', '%'. $search . '%')
                ->orWhere('nobuktikas', 'like', '%'. $search . '%');

            if ($from && $to) {
                $query->whereBetween('tanggal', [$from, $to]);
            }
        });
    }

    if($penerima){
        $kaskecil->where('penerima',$penerima);
    }
    if($nobuktikas){
        $kaskecil->where('nobuktikas','like','%'.$nobuktikas . '%');
    }
    if($tanggal){
        $kaskecil->where('tanggal', $tanggal);
    }
    if($CoAdebit_id){
        $kaskecil->where('CoAdebit_id',$CoAdebit_id);
    }
    if($CoAkredit_id){

```

```

    $kaskecil->where('CoAkredit_id',$CoAkredit_id);
}
if($from && $to){
    $kaskecil->whereBetween('tanggal', [$from, $to]);
}

return ResponseFormatter::success($kaskecil-
>orderBy('tanggal','desc')->paginate($limit),'Kaskecil Found');
}

public function create(CreateKaskecilRequest $request)
{
    try {
        $kaskecil = Kaskecil::create([
            'name' => $request->name,
            'penerima' => $request->penerima,
            'nobuktikas' => $request->nobuktikas,
            'tanggal' => $request->tanggal,
            'ref' => $request->ref,
            'jumlah' => $request->jumlah,
            'CoAdebit_id' => $request->CoAdebit_id,
            'CoAkredit_id' => $request->CoAkredit_id,
            'cabang_id' => $request->cabang_id,
            'jenis_data' => $request->jenis_data
        ]);

        if(!$kaskecil)
        {
            throw new Exception('Kaskecil not created');
        }

        return ResponseFormatter::success($kaskecil, 'Kaskecil created');
    } catch (Exception $e) {
        return ResponseFormatter::error($e->getMessage(), 500);
    }
}

public function update(UpdateKaskecilRequest $request, $id)
{
    try {
        $kaskecil = Kaskecil::find($id);
    }
}

```

```

if(!$kaskecil)
{
    throw new Exception('Kaskecil not created');
}

//update kaskecil
$kaskecil->update([
    'name' => $request->name,
    'penerima' => $request->penerima,
    'nobuktikas' => $request->nobuktikas,
    'tanggal' => $request->tanggal,
    'ref' => $request->ref,
    'jumlah' => $request->jumlah,
    'CoAdebit_id' => $request->CoAdebit_id,
    'CoAkredit_id' => $request->CoAkredit_id,
]);

return ResponseFormatter::success($kaskecil, 'Kaskecil updated');

} catch (Exception $e) {
    return ResponseFormatter::error($e->getMessage(), 500);
}
}

public function destroy($id)
{
    try {
        //Get Kaskecil
        $kaskecil = Kaskecil::find($id);

        //check if kaskecil exists
        if (!$kaskecil) {
            throw new Exception('Kaskecil not Found');
        }

        //Delete Kaskecil
        $kaskecil->delete();

        return ResponseFormatter::success('Kaskecil deleted');
    } catch (Exception $e) {
        return ResponseFormatter::error($e->getMessage(),500);
    }
}

```



```

}
public function export(Request $request)
{
    $cabang_id = $request->input('cabang_id');
    $from = $request->input('from');
    $to = $request->input('to');
    return Excel::download(new KaskecilExport($cabang_id, $from, $to),
'kaskecil.xlsx');
}
}

```

UserController.php

Sourcecode 32 *UserController.php*

```

<?php

namespace App\Http\Controllers\API;

use Exception;
use App\Models\User;
use Illuminate\Http\Request;
use Illuminate\Validation\Rule;
use App\Helpers\ResponseFormatter;
use Laravel\Fortify\Rules\Password;
use App\Http\Controllers\Controller;
use App\Http\Requests\UpdateUserRequest;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;

class UserController extends Controller
{
    public function Login(Request $request)
    {
        try {
            $request->validate([
                'email' => 'required|email',
                'password' => 'required',

            ]);

            $credentials = request(['email', 'password']);
            if(!Auth::attempt($credentials)){
                return ResponseFormatter::error('Unauthorized', 401);
            }
        }
    }
}

```

```

        $user = User::where('email', $request->email)->first();
        if(!Hash::check($request->password, $user->password)) {
            throw new Exception('Invalid Password');
        }

        //Generate Token
        $tokenResult = $user->createToken('authToken')-
>plainTextToken;

        //return Response
        return ResponseFormatter::success([
            'access_token' => $tokenResult,
            'token_type' => 'Bearer',
            'user' => $user
        ], 'Login success');

    } catch (Exception $e) {
        return ResponseFormatter::error('Authentication Failed');
    }
    //Validasi request
}
public function register(Request $request)
{
    try {
        $request->validate([
            'name' => ['required', 'string', 'max:255'],
            'email' =>
['required', 'string', 'email', 'max:255', 'unique:users'],
            'password' => ['required', 'string', new Password],
            'role' => ['required'],
            'cabang_id' => ['required'],
        ]);

        //create user
        $user = User::create([
            'name' => $request->name,
            'email' => $request->email,
            'role' => $request->role,
            'cabang_id' => $request->cabang_id,
            'password' => Hash::make($request->password),
        ]);
    }
}

```

```

        //Generate Token
        $tokenResult = $user->createToken('authToken')-
>plainTextToken;

        //Return Response
        return ResponseFormatter::success([
            'access_token' => $tokenResult,
            'token_type' => 'Bearer',
            'user' => $user
        ], 'Register success');
    } catch (Exception $error) {
        //error response
        return ResponseFormatter::error($error->getMessage());
    }
}
public function update(UpdateUserRequest $request)
{
    try {
        $user = User::findOrFail($request->id);
        if(!$user)
        {
            throw new Exception('CoA not created');
        }
        //Update user
        $user->update([
            'name' => $request->name,
            'email' => $request->email,
            'role' => $request->role,
            'cabang_id' => $request->cabang_id,
        ]);
        if($request->has('password')) {
            $user->password = Hash::make($request->password);
        }

        //Generate Token
        $tokenResult = $user->createToken('authToken')-
>plainTextToken;

        //Return Response
        return ResponseFormatter::success([
            'access_token' => $tokenResult,
            'token_type' => 'Bearer',

```

```

        'user' => $user
    ], 'Register success');
} catch (Exception $e) {
    return ResponseFormatter::error($e->getMessage(), 500);
}
}

public function Logout(Request $request)
{
    //Revoke Token
    $token = $request->user()->currentAccessToken()->delete();

    //Return response
    return ResponseFormatter::success($token, 'Logout Success');
}

public function fetch(Request $request)
{
    $user = $request->user();
    return ResponseFormatter::success($user, 'Fetch Success');
}

public function fetchAll(Request $request)
{
    $id = $request->input('id');
    $name = $request->input('name');
    $email = $request->input('email');
    $role = $request->input('role');
    $cabang_id = $request->input('cabang_id');
    $limit = $request->input('limit');
    $search = $request->input('search');

    $userQuery = User::with(['cabang']);

    //get single data
    if($id)
    {
        $user = $userQuery->find($id);

        if($user)
        {
            return ResponseFormatter::success($user, "User found");
        }
    }
}

```

```

        return ResponseFormatter::error('User not found');
    }

    //get multiple data
    $users = $userQuery;

    if($search){
        $users->where('name','like','%'.$search.'%')-
>orWhere('email','like','%'.$search.'%');
    }

    if($name){
        $users->where('name',$name);
    }

    if($email){
        $users->where('email',$email);
    }

    if($role){
        $users->where('role', $role);
    }
    if($cabang_id){
        $users->where('cabang_id', $cabang_id);
    }

    if($limit)
    {
        return ResponseFormatter::success($users-
>orderBy('cabang_id','asc')->paginate($limit),'Users Found');
    }

    return ResponseFormatter::success($users->get(),'Users
Found');
}
}

```

Controller.php

Sourcecode 33 Controller.php

```
<?php
```

```

namespace App\Http\Controllers;

use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
use Illuminate\Foundation\Bus\DispatchesJobs;
use Illuminate\Foundation\Validation\ValidatesRequests;
use Illuminate\Routing\Controller as BaseController;

class Controller extends BaseController
{
    use AuthorizesRequests, DispatchesJobs, ValidatesRequests;
}

```

Muzakkicontroller.php

Sourcecode 34 Muzakkicontroller.php

```

<?php

namespace App\Http\Controllers\API;

use Exception;
use App\Models\Muzakki;
use Illuminate\Http\Request;
use App\Helpers\ResponseFormatter;
use App\Http\Controllers\Controller;
use App\Http\Requests\CreateMuzakkiRequest;
use App\Http\Requests\UpdateMuzakkiRequest;
use App\Imports\MuzakkiImport;
use Maatwebsite\Excel\Facades\Excel;

class Muzakkicontroller extends Controller
{
    public function fetch(Request $request)
    {
        $id = $request->input('id');
        $name = $request->input('name');
        $lembaga = $request->input('lembaga');
        $no_telp = $request->input('no_telp');
        $nik = $request->input('nik');
        $NPWP = $request->input('NPWP');

        //get single data
        if($id)
        {

```

```

        $Muzakki = Muzakki::where('cabang_id',$request-
>cabang_id)->find($id);

        if($Muzakki)
        {
            return ResponseFormatter::success($Muzakki);
        }
        return ResponseFormatter::error('Muzakki not found');
    }
    //get multiple data
    $Muzakkis = Muzakki::query()->where('cabang_id',$request-
>cabang_id);

    if($name){
        $Muzakkis->where('name','like','%'.$name . '%')-
>where('cabang_id',$request->cabang_id);
    }
    return ResponseFormatter::success($Muzakkis-
>paginate($limit),'Muzakkis Found');
}

public function create(CreateMuzakkiRequest $request)
{
    try {
        $Muzakki = Muzakki::create([
            'name' => $request->name,
            'lembaga' => $request->lembaga,
            'no_telp' => $request->no_telp,
            'nik' => $request->nik,
            'NPWP' => $request->NPWP,
        ]);

        if(!$Muzakki)
        {
            throw new Exception('Muzakki not created');
        }

        return ResponseFormatter::success($Muzakki, 'Muzakki
created');
    } catch (Exception $e) {
        return ResponseFormatter::error($e->getMessage(), 500);
    }
}

```

```

    }

    public function update(UpdateMuzakkiRequest $request, $id)
    {
        try {
            $Muzakki = Muzakki::find($id);

            if(!$Muzakki)
            {
                throw new Exception('Muzakki not created');
            }

            //update Muzakki
            $Muzakki->update([
                'name' => $request->name,
                'lembaga' => $request->lembaga,
                'no_telp' => $request->no_telp,
                'nik' => $request->nik,
                'NPWP' => $request->NPWP,
            ]);

            return ResponseFormatter::success($Muzakki, 'Muzakki
updated');

        } catch (Exception $e) {
            return ResponseFormatter::error($e->getMessage(), 500);
        }
    }

    public function destroy($id)
    {
        try {
            //Get Muzakki
            $Muzakki = Muzakki::find($id);

            //check if Muzakki exists
            if (!$Muzakki) {
                throw new Exception('Muzakki not Found');
            }

            //Delete Muzakki
            $Muzakki->delete();
        }
    }

```



```

        return ResponseFormatter::success('Muzakki deleted');
    } catch (Exception $e) {
        return ResponseFormatter::error($e->getMessage(),500);
    }
}
public function import_excel(Request $request)
{
    // import data
    Excel::import(new MuzakkiImport, $request-
>file('file')->store('temp'));

    return ResponseFormatter::success('Import Success');
}
}

```

Cabang.php

Sourcecode 35 Cabang.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Cabang extends Model
{
    use HasFactory;
    protected $fillable = [
        'name'
    ];

    public function users()
    {
        return $this->hasMany(User::class);
    }

    public function CoAdebits()
    {
        return $this->hasMany(CoA::class);
    }
}

```

```

    public function CoAkredits()
    {
        return $this->hasMany(CoA::class);
    }

    public function frontoffices()
    {
        return $this->hasMany(Frontoffice::class);
    }

    public function jurnalumums()
    {
        return $this->hasMany(Jurnalumum::class);
    }

    public function kasbanks()
    {
        return $this->hasMany(Kasbank::class);
    }

    public function kasbesars()
    {
        return $this->hasMany(Kasbesar::class);
    }

    public function kaskecils()
    {
        return $this->hasMany(Kaskecil::class);
    }
}

```

CoA.php

Sourcecode 36 CoA.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Cabang extends Model

```

```

{
    use HasFactory;
    protected $fillable = [
        'name'
    ];

    public function users()
    {
        return $this->hasMany(User::class);
    }

    public function CoAdebits()
    {
        return $this->hasMany(CoA::class);
    }

    public function CoAkredits()
    {
        return $this->hasMany(CoA::class);
    }

    public function frontoffices()
    {
        return $this->hasMany(Frontoffice::class);
    }

    public function jurnalumums()
    {
        return $this->hasMany(Jurnalumum::class);
    }

    public function kasbanks()
    {
        return $this->hasMany(Kasbank::class);
    }

    public function kasbesars()
    {
        return $this->hasMany(Kasbesar::class);
    }

    public function kaskecils()

```

```

    {
        return $this->hasMany(Kaskecil::class);
    }
}

```

CoAdebit.php

Sourcecode 37 CoAdebit.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class CoAdebit extends Model
{
    use HasFactory;

    /**
     * The attributes that are mass assignable.
     *
     * @var string[]
     */
    protected $fillable = [
        'name',
        'kode',
        'laporan',
        'cabang_id'
    ];

    public function frontoffices()
    {
        return $this->hasMany(Frontoffice::class);
    }
    public function kasbesars()
    {
        return $this->hasMany(Frontoffice::class);
    }
    public function cabang()
    {
        return $this->belongsTo(Cabang::class);
    }
}

```

```
}
}
```

CoAkredit.php

Sourcecode 38 CoAkredit.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class CoAkredit extends Model
{
    use HasFactory;

    /**
     * The attributes that are mass assignable.
     *
     * @var string[]
     */
    protected $fillable = [
        'name',
        'kode',
        'laporan',
        'cabang_id'
    ];

    public function frontoffices()
    {
        return $this->hasMany(Frontoffice::class);
    }
}
```

Frontoffice.php

Sourcecode 39 Frontoffice.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```

use Illuminate\Database\Eloquent\Model;

class Frontoffice extends Model
{
    use HasFactory;

    /**
     * The attributes that are mass assignable.
     *
     * @var string[]
     */
    protected $fillable = [
        'name',
        'penyetor',
        'penerima',
        'nobuktipenerima',
        'tanggal',
        'ref',
        'jumlah',
        'tempatbayar',
        'CoAdebit_id',
        'CoAkredit_id',
        'cabang_id',
    ];

    public function CoAdebit()
    {
        return $this->belongsTo(CoA::class,'CoAdebit_id');
    }
    public function CoAkredit()
    {
        return $this->belongsTo(CoA::class,'CoAkredit_id');
    }
    public function cabang()
    {
        return $this->belongsTo(Cabang::class);
    }
}

```

Kasbank.php

Sourcecode 40 Kasbank.php

```
<?php
```

```

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Kasbank extends Model
{
    use HasFactory;

    protected $fillable = [
        'name',
        'penerima',
        'nobuktikas',
        'tanggal',
        'ref',
        'jumlah',
        'CoAdebit_id',
        'CoAkredit_id',
        'cabang_id',
        'jenis_data'
    ];

    public function CoAdebit()
    {
        return $this->belongsTo(CoA::class,'CoAdebit_id');
    }
    public function CoAkredit()
    {
        return $this->belongsTo(CoA::class,'CoAkredit_id');
    }
}

```

CoAkredit.php

Sourcecode 41 CoAkredit.php

```

<?php

namespace App\Models;

use Laravel\Sanctum\HasApiTokens;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Factories\HasFactory;

```

```

class Kasbesar extends Model
{
    use HasFactory;

    protected $fillable = [
        'name',
        'penerima',
        'nobuktikas',
        'tanggal',
        'ref',
        'jumlah',
        'CoAdebit_id',
        'CoAkredit_id',
        'cabang_id',
        'jenis_data'
    ];

    public function CoAdebit()
    {
        return $this->belongsTo(CoA::class,'CoAdebit_id');
    }
    public function CoAkredit()
    {
        return $this->belongsTo(CoA::class,'CoAkredit_id');
    }
}

```

Kaskecil.php

Sourcecode 42 Kaskecil.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Kaskecil extends Model
{
    use HasFactory;

    protected $fillable = [

```



```

        'name',
        'penerima',
        'nobuktikas',
        'tanggal',
        'ref',
        'jumlah',
        'CoAdebit_id',
        'CoAkredit_id',
        'cabang_id',
        'jenis_data'
    ];

    public function CoAdebit()
    {
        return $this->belongsTo(CoA::class,'CoAdebit_id');
    }
    public function CoAkredit()
    {
        return $this->belongsTo(CoA::class,'CoAkredit_id');
    }
}

```

Muzakki.php

Sourcecode 43 Muzakki.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Muzakki extends Model
{
    use HasFactory;

    protected $fillable = [
        'name',
        'lembaga',
        'no_telp',
        'nik',
        'NPWP',
    ];
}

```

```

    public function CoAdebit()
    {
        return $this->belongsTo(CoA::class,'CoAdebit_id');
    }
    public function CoAkredit()
    {
        return $this->belongsTo(CoA::class,'CoAkredit_id');
    }
}

```

User.php

Sourcecode 44 User.php

```

<?php

namespace App\Models;

use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Fortify\TwoFactorAuthenticatable;
use Laravel\Jetstream\HasProfilePhoto;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens;
    use HasFactory;
    use HasProfilePhoto;
    use Notifiable;
    use TwoFactorAuthenticatable;

    /**
     * The attributes that are mass assignable.
     *
     * @var string[]
     */
    protected $fillable = [
        'name',
        'email',
        'password',
    ];
}

```

```

        'cabang_id',
        'role'
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array
     */
    protected $hidden = [
        'password',
        'remember_token',
        'two_factor_recovery_codes',
        'two_factor_secret',
    ];

    /**
     * The attributes that should be cast.
     *
     * @var array
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];

    /**
     * The accessors to append to the model's array form.
     *
     * @var array
     */
    protected $appends = [
        'profile_photo_url',
    ];

    // Relasi one to many cabang
    public function cabang()
    {
        return $this->belongsTo(Cabang::class);
    }
}

```