BAB I

PENDAHULUAN

A. Latar Belakang

Usaha Mikro, Kecil, dan Menengah (UMKM) memiliki kontribusi yang signifikan terhadap pertumbuhan ekonomi nasional. Menurut data Kementerian Koperasi dan UKM, UMKM menyumbang lebih dari 60% terhadap Produk Domestik Bruto (PDB) Indonesia dan menyerap lebih dari 97% tenaga kerja nasional. Namun, salah satu tantangan utama yang dihadapi UMKM adalah pengelolaan keuangan yang kurang optimal akibat pencatatan manual atau penggunaan alat yang tidak terintegrasi.

Sistem pencatatan manual sering kali menyebabkan masalah seperti kehilangan data, kesalahan dalam perhitungan, dan ketidakmampuan untuk menghasilkan laporan keuangan yang akurat. Laporan keuangan yang tidak terstruktur ini dapat berdampak buruk pada pengambilan keputusan bisnis, terutama saat UMKM perlu mengajukan pinjaman modal atau menarik investor.

Di era transformasi digital, penggunaan sistem laporan keuangan berbasis web menjadi solusi yang relevan. Sistem ini menawarkan efisiensi dalam pengelolaan data keuangan, aksesibilitas melalui perangkat digital, dan kemudahan dalam menghasilkan laporan sesuai dengan standar akuntansi (Rizki et al., 2022).

Penelitian terdahulu terkait system pelaporan keuangan yaitu Penelitian yang dilakukan oleh: Hasan (2023), berjudul Sistem Informasi Manajemen PT Calidad

Coco Indonesia. Hasil dari penelitian ini dibangunnya suatu sistem berbasis Android, memberikan kemudahan kepada pengguna dalam menjalankan tugasnya dalam perusahaan PT. Calidad Coco Indonessia. Penelitian yang dilakukan oleh: Muh. Ade Candra (2017) berjudul Aplikasi Sistem Evaluasi Dan Pelaporan Alokasi Dana Zis Pada Baznas Berbasis Web. hasil dari penelitian ini membangun suatu inovasi layanan digital dalam mempermudah Baznas dalam melaksanakan pengelolaan zakat, infaq dan sedekah (ZIS) dengan pemanfaatan suatu Aplikasi.

Berdasarkan bahasan diatas peneliti mengangkat judul penelitian sistem laporan keuangan UMKM berbasis web, perbedaan antara penelitian terdahulu dan penelitian yang akan dibuat berbeda dari lokasi penelitian dan teknologi yang digunakan, teknologi yang digunakan pada penelitian ini adalah *progressive web applications* dengan kelebihan instalasi mudah pengguna dapat menambahkan aplikasi ke layar utama perangkat mereka langsung dari browser tanpa perlu mengunduh melalui toko aplikasi, Akses *offline* yaitu menggunakan *service worker*, *progressive web applications* dapat menyimpan aset dan data secara local. *Notifikasi push* yaitu memungkin kan pengembang untuk berkomunikasi secara *real-time* dengan pengguna, meningkatkan interaksi dan retensi pengguna. *Responsive* yaitu menyesuaikan dengan berbagai ukuran layar dan perangkat, memastikan pengalaman pengguna yang konsisten.

B. Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah dalam penelitian ini adalah bagaimana cara membuat sistem laporan keuangan UMKM ?

C. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah untuk mengetahui cara merancang system laporan keuangan berbasis web.

D. Batasan Masalah

Agar penelitian ini lebih terarah, beberapa batasan masalah yang diterapkan adalah:

1. Sistem tidak mencakup penjualan dalam aplikasi, penggajian karyawan, dan absensi karyawan.
2. Sistem tidak dirancang dalam bentuk aplikasi android.

E. Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah:

1. Bagi UMKM: Meningkatkan kemampuan dalam mengelola dan melaporkan keuangan secara profesional.
2. Bagi Akademisi: Menambah wawasan tentang penerapan teknologi informasi dalam bidang akuntansi UMKM.
3. Bagi Pengembang Teknologi: Memberikan referensi untuk pengembangan sistem serupa dengan fitur yang lebih inovatif.

F. Sistematika Penulisan

Proposal ini disusun dengan sistematika sebagai berikut:

1. BAB I Pendahuluan: Berisi latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metode penelitian, dan

sistematika penulisan terkait dengan penelitian sistem laporan keuangan UMKM .

2. BAB II Tinjauan Pustaka: Berisi tentang teori-teori yang mendukung penelitian seperti teori tentang laporan keuangan, teori UMKM, teori website serta tinjauan hasil penelitian terdahulu.
3. BAB III Metode Penelitian: Membahas metode pengumpulan data, perancangan sistem, serta tahapan pengembangan.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Usaha Mikro Kecil dan Menengah UMKM

a. Pengertian UMKM

UMKM di Indonesia telah menjadi bagian penting dari sistem perekonomian di Indonesia. Hal ini dikarenakan UMKM merupakan unit-unit usaha yang lebih banyak jumlahnya dibandingkan usaha industri berskala besar dan memiliki keunggulan dalam menyerap tenaga kerja lebih banyak dan juga mampu mempercepat proses pemerataan sebagai bagian dari pembangunan (Rahmah Z.Z, 2022).

Usaha Mikro, Kecil dan Menengah (UMKM) adalah usaha yang memproduksi barang dan jasa yang menggunakan bahan baku utama berbasis pada pendayagunaan sumber daya alam, bakat dan karya seni tradisional dari daerah setempat. Adapun ciri-ciri UMKM adalah bahan baku mudah diperolehnya, menggunakan teknologi sederhana sehingga mudah dilakukan alih teknologi, keterampilan dasar umumnya sudah dimiliki secara turun temurun, bersifat padat karya atau menyerap tenaga kerja yang cukup banyak, peluang pasar cukup luas, sebagian besar produknya terserap di pasar lokal atau domestik dan tidak tertutup sebagian lainnya berpotensi untuk diekspor, beberapa komoditi tertentu memiliki ciri khas terkait dengan karya seni budaya daerah

setempat serta melibatkan masyarakat ekonomi lemah setempat secara ekonomis dan menguntungkan.

UMKM mempunyai peran dan strategi dalam membangun ekonomi bangsa. Selain berperan dalam pertumbuhan ekonomi dan penyerapan tenaga kerja, UMKM juga berperan dalam mendistribusikan hasil-hasil pembangunan (Rahmah Z.Z, 2022).

b. Undang-Undang Yang Mengatur UMKM

Undang-Undang Nomor 20 Tahun 2008 tentang Usaha Mikro, Kecil dan Menengah (UMKM) adalah usaha produktif milik orang perorangan dan/atau badan usaha perorangan yang memenuhi kriteria Usaha Mikro sebagaimana diatur dalam Undang-Undang ini. Usaha Kecil adalah usaha ekonomi produktif yang berdiri sendiri, yang dilakukan oleh orang perorangan atau badan usaha yang bukan merupakan anak perusahaan atau bukan cabang perusahaan yang dimiliki, dikuasai, atau menjadi bagian baik langsung maupun tidak langsung dari Usaha Menengah atau Usaha Besar yang memenuhi kriteria Usaha Kecil sebagaimana dimaksud dalam Undang-Undang ini.

Usaha kecil saat ini sudah banyak di Indonesia dan memang Indonesia kini membutuhkan pembisnis. Adanya pembisnis di Indonesia akan mendongkrak ekonomi Indonesia menjadi lebih baik lagi. Sealin itu, dengan adanya usaha kecil tersebut juga bisa menciptkan lapangan pekerjaan bagi yang membutuhkan. Saat kalian tidak mendapatkan pekerjaan apa salahnya jika mencoba untuk membuka lapangan pekerjaan.

Usaha Menengah adalah usaha ekonomi produktif yang berdiri sendiri, yang dilakukan oleh orang perorangan atau badan usaha yang bukan merupakan anak perusahaan atau cabang perusahaan yang dimiliki, dikuasai, atau menjadi bagian baik langsung maupun tidak langsung dengan Usaha Kecil atau Usaha Besar dengan jumlah kekayaan bersih atau hasil penjualan tahunan sebagaimana diatur dalam UndangUndang ini.

Usaha Besar adalah usaha ekonomi produktif yang dilakukan oleh badan usaha dengan jumlah kekayaan bersih atau hasil penjualan tahunan lebih besar dari Usaha Menengah, yang meliputi usaha nasional milik negara atau swasta, usaha patungan, dan usaha asing yang melakukan kegiatan ekonomi di Indonesia. (https://peraturan.go.id/terjemahresmi/uu-no-20-tahun-2008).

c. Kriteria UMKM

Ada beberapa kriteria-kriteria tertentu supaya sebuah usaha dapat dikatakan sebagai UMKM menurut Peraturan Pemerintah Nomor 7 Tahun 2021 perihal Kemudahan, Pelindungan, Dan Pemberdayaan Koperasi Dan Usaha Mikro, Kecil, Dan Menengah (PP 7 tahun 2021), berikut ini adalah penjelasannya

1) Usaha Mikro

   Mempunyai modal usaha hingga dengan paling banyak Rp1 miliar, tidak termasuk tanah dan bangunan tempat usaha.

2) Usaha Kecil

   Mempunyai modal usaha lebih Rp1 – 5 miliar, tidak termasuk tanah dan bangunan tempat usaha.

3) Usaha Menengah

Mempunyai modal usaha Rp5 – 10 miliar, tidak termasuk tanah dan bangunan tempat usaha.

2. Laporan Keuangan

a. Pengertian Laporan Keuangan

Dalam menjalankan kegiatan operasionalnya, perusahaan secara periodik menyusun laporan keuangan untuk memberikan informasi kepada stakeholder atau pemegang kepentingan. Untuk lebih memahami apa itu laporan keuangan, berikut beberapa pendapat tentang pengertian laporan keuangan menurut para ahli:

1) Menurut Kasmir (2019: 7) Laporan keuangan adalah laporan yang menunjukkan kondisi keuangan perusahaan pada saat ini atau dalam suatu periode tertentu.
2) Menurut Werner R. Murhadi (2019: 1) laporan keuangan merupakan bahasa bisnis. Di dalam laporan keuangan berisi informasi mengenai kondisi keuangan perusahaan kepada pihak pengguna. Dengan memahami laporan keuangan suatu perusahaan, maka berbagai pihak yang berkepentingan dapat melihat kondisi kesehatan keuangan suatu perusahaan.
3) Menurut Raymond Budiman (2020: 3) laporan keuangan merupakan suatu dokumen yang menggambarkan kondisi keuangan perusahaan dan kinerja perusahaan dalam periode tertentu.

4) Menurut Sutrisno (2012: 9) laporan keuangan adalah hasil akhir dari proses akuntansi yang meliputi dua laporan utama yakni neraca dan laporan laba rugi.(Syaharman, 2021).

b. Jenis-jenis Laporan Keuangan

Menurut Kasmir (2019: 28-30) secara umum ada lima macam jenis laporan keuangan yang biasa di susun yaitu:

1) Neraca Neraca (balance sheet) merupakan laporan yang menunjukkan posisi keuangan perusahaan pada tanggal tertentu.
2) Laporan Laba Rugi Laporan laba rugi (income statement) merupakan laporan keuangan yang menggambarkan hasil usaha perusahaan dalam suatu periode tertentu. Dalam laporan laba rugi ini tergambar jumlah pendapatan dan sumber-sumber pendapatan yang diperoleh. Kemudian, juga tergambar jumlah biaya dan jenis-jenis biaya yang dikeluarkan selama periode tertentu.
3) Laporan Perubahan Modal Laporan perubahan modal merupakan laporan yang berisi jumlah dan jenis modal yang dimiliki pada saat ini. Kemudian, laporan ini juga menjelaskan perubahan modal dan sebab-sebab terjadinya perubahan modal di perusahaan.
4) Laporan Arus Kas Laporan arus kas merupakan laporan yang menunjukkan semua aspek yang berkaitan dengan kegiatan perusahaan, baik yang berpengaruh langsung atau tidak langsung terhadap kas. 5. Laporan Catatan Atas Laporan Keuangan Laporan Catatan Atas Laporan Keuangan merupakan laporan yang memberikan informasi apabila ada laporan keuangan yang memerlukan penjelasan tertentu (Syaharman, 2021).

c. Data Laporan Keuangan

Berikut adalah table naraca saldo sebagai berikut:

Tabel 2.1 Naraca Saldo

Kode Akun	Nama Akun	Debit	Kredit
	Kas	Rp. 410.600.000	
	Perlengkapan	Rp. 10.000.000	
	Peralatan	Rp. 40.000.000	
	Persediaan barang dagang	Rp. 20.000.000	
	Persediaan bahan baku	Rp. 100.000.000	
	Sewa dibayar dimuka	Rp. 10.000.000	
	Modal		Rp. 500.000.000
	Prive	Rp. 300.000	
	Penjualan		Rp. 110.300.000
	Beban gaji	Rp. 15.000.000	
	Beban Listrik, air dan pulsa	Rp. 2.000.000	
	Beban pajak	Rp. 400.000	
	Beban lain-lain	Rp. 2.000.000	
		Rp. 610.300.000	Rp. 610.300.000

Berikut adalah table jurnal umum sebagai berikut:

Tabel 2.2 Jurnal Umum

Hari/Tanggal	Keterangan	Ref	Debit	Kredit
01/01/2024	Kas		Rp. 10.000.000	
	Modal			Rp. 10.000.000
02/01/2024	Persediaan bahan baku		Rp.20.000.000	
	Kas			Rp. 20.000.000
03/01/2024	Persediaan bahan baku		Rp. 10.000.000	
	Kas			Rp. 10.000.000
04/01/2024	Sewa dibayar di muka		Rp. 1.500.00	
	Kas			Rp. 1.500.000
07/01/2024	Peralatan		Rp. 10.000.000	
	Kas			Rp. 10.000.000
07/01/2024	Perlengkapan		Rp. 500.000	
	Kas			Rp. 500.000
08/01/2024	Persediaan bahan baku		Rp. 15. 000.000	
	Kas			Rp. 15. 000.000
10/01/2024	Persediaan bahan baku		Rp. 10.000.000	
	Kas			Rp. 10.000.000
11/01/2024	Peralatan		Rp.20.000.000	
	Kas			Rp. 20.000.000

Lanjutan Tabel 2.2 Jurnal Umum

Hari/Tanggal	Keterangan	Ref	Debit	Kredit
12/01/2024	Perelengkapan		Rp. 10.000.000	
	Kas			Rp. 10.000.000
13/01/2024	Kas		Rp. 1.500.00	
	Penjualan			Rp. 1.500.000
14/01/2024	Persediaan barang dagang		Rp. 10.000.000	
	Kas			Rp. 10.000.000
14/01/2024	Kas		Rp.20.000.000	
	Penjualan			Rp. 20.000.000
15/01/2024	Kas		Rp. 10.000.000	
	Penjualan			Rp. 10.000.000
16/01/2024	Kas		Rp. 1.500.00	
	Penjualan			Rp. 1.500.000
17/01/2024	Kas		Rp. 10.000.000	
	Penjualan			Rp. 10.000.000
18/01/2024	Kas		Rp. 500.000	
	Penjualan			Rp. 500.000
19/01/2024	Kas		Rp. 15. 000.000	
	Penjualan			Rp. 15. 000.000
20/01/2024	Kas		Rp. 10.000.000	
	Penjualan			Rp. 10.000.000
21/01/2024	Kas		Rp. 10.000.000	
	Penjualan			Rp. 10.000.000
22/01/2024	Kas		Rp.20.000.000	
	Penjualan			Rp. 20.000.000
23/01/2024	Kas		Rp. 10.000.000	
	Penjualan			Rp. 10.000.000
24/01/2024	Kas		Rp. 1.500.00	
	Penjualan			Rp. 1.500.000
25/01/2024	Kas		Rp. 10.000.000	
	Penjualan			Rp. 10.000.000
26/01/2024	Kas		Rp. 500.000	
	Penjualan			Rp. 500.000
27/01/2024	Beban gaji		Rp. 15. 000.000	
	Kas			Rp. 15. 000.000
28/01/2024	Beban listrik, air dan pulsa		Rp. 2.000. 000	
	Kas			Rp. 2.000.000
29/01/2024	Beban pajan		Rp. 400.000	
	Kas			Rp. 400.000
30/01/2024	Beban lain lain		Rp. 200.000	
	Kas			Rp. 200.000
31/01/2024	Prive		Rp. 300.000	
	Kas			Rp. 300.000
	Total		Rp. 810.000.000	Rp. 810.000.000

Berikut adalah table buku besar sebagai berikut:

Tabel 2.3 Buku Besar Kas

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
1	Menyetor modal		Rp. 500.000.000		Rp. 500.000.000	
2	Membeli bahan			Rp. 30.000.000	Rp. 470.000.000	
3	Membeli bahan			Rp. 20.000.000	Rp. 450.000.000	
4	Membayar sewa			Rp. 10.000.000	Rp. 440.000.000	
7	Membeli peralatan			Rp. 20.000.000	Rp. 420.000.000	
7	Membeli perlengkapan			Rp. 5.00.000	Rp. 415.000.000	
8	Membeli bahan			Rp. 30.000.000	Rp. 385.000.000	
10	Membeli bahan			Rp. 20.000.000	Rp. 365.000.000	
11	Membeli peralatan			Rp. 20.000.000	Rp. 345.000.000	
12	Membeli perlengkapan			Rp. 5.000.000	Rp. 340.000.000	

Tabel 2.4 Buku Besar Perlengkapan

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
7	Membeli perlengkapan		Rp. 5.000.00		Rp. 5.000.000	
12	Membeli perlengkapan		Rp.5.000.000		Rp. 5.000.000	

Tabel 2.5 Buku Besar Peralatan

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
7	Membeli peralatan		Rp. 20.000.000		Rp. 20.000.000	
11	Membeli peralatan		Rp.20. 000.000		Rp. 40.000.000	

Tabel 2.6 Buku Besar Persediaan Barang Dagang

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
14	Membeli barang		Rp. 20.000.000		Rp. 20.000.000	

Tabel 2.7 Buku Besar Persediaan Bahan

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
2	Membeli bahan		Rp.30.000.000		Rp. 30.000.000	
3	Membeli bahan		Rp. 20.000.000		Rp. 50.000.000	
8	Membeli bahan		Rp. 30.000.000		Rp. 80.000.000	
10	Membeli bahan		Rp. 20.000.000		Rp. 100.000.000	

Tabel 2.5 Buku Besar Sewa dibaya dimuka

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
4	Membayar sewa		Rp. 10.000.000		Rp. 10.000.000	

Tabel 2.5 Buku Besar Modal

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
1	Menyetor modal			Rp. 500.000.000		Rp. 500.000.000

Tabel 2.6 Buku Besar Prive

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
31	Pengembalian pribadi			Rp. 300.000.000		Rp. 300.000.000

Tabel 2.7 Buku Besar Penjualan

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
13	Penjualan			Rp. 2.800.000		Rp. 2.800.000
14	Penjualan			Rp. 20.000.000		Rp. 22.800.00
15	Penjualan			Rp. 5.000.000		Rp. 27. 800.000
16	Penjualan			Rp. 500.000		Rp. 28.300.000

Tabel 2.4 Buku Besar Beban Gaji

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
2	Membeli bahan		Rp.30.000.000		Rp. 30.000.000	
3	Membeli bahan		Rp. 20.000.000		Rp. 50.000.000	

Tabel 2.5 Buku Besar Beban Listrik,Air, Dan Pulsa

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
7	Membeli peralatan		Rp. 20.000.000		Rp. 20.000.000	
11	Membeli peralatan		Rp.20. 000.000		Rp. 40.000.000	

Tabel 2.6 Buku Besar Beban Pajak

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
14	Membeli barang		Rp. 20.000.000		Rp. 20.000.000	

Tabel 2.7 Buku Besar Beban Lain-Lain

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
2	Membeli bahan		Rp.30.000.000		Rp. 30.000.000	
3	Membeli bahan		Rp. 20.000.000		Rp. 50.000.000	
8	Membeli bahan		Rp. 30.000.000		Rp. 80.000.000	
10	Membeli bahan		Rp. 20.000.000		Rp. 100.000.000	

Berikut adalah table neraca jalur sebagai berikut:

Tabel 2.7 Buku Besar Persediaan Bahan

Kode Akun	Nama Akun	Neraca Saldo		Laba Rugi		Naraca	
		Debit	Kredit	Debit	Kredit	Debit	Kredit
	Kas	Rp. 410.600.000				Rp. 410.600.000	
	Perlengkapan	Rp. 10.000.000				Rp. 10.000.000	
	Peralatan	Rp. 40.000.000				Rp. 40.000.000	

Lanjutan Tabel 2.7 Buku Besar Persediaan Bahan

Kode Akun	Nama Akun	Neraca Saldo		Laba Rugi		Naraca	
		Debit	**Kredit**	**Debit**	**Kredit**	**Kredit**	**Debit**
	Persediaan barang dagang	Rp. 20.000.000				Rp. 20.000.000	
	Persediaan bahan baku	Rp. 100.000.000				Rp. 100.000.000	
	Sewa dibayar dimuka	Rp. 10.000.000				Rp. 10.000.000	
	Modal		Rp. 500.000.000				Rp. 500.000.000
	Prive	Rp. 300.000			Rp. 110.300.000	Rp. 300.000	
	Penjualan		Rp. 110.300.000	Rp. 15.000.000			Rp. 110.300.000
	Beban gaji	Rp. 15.000.000		Rp. 2.000.000		Rp. 15.000.000	
	Beban Listrik, air dan pulsa	Rp. 2.000.000		Rp. 400.000		Rp. 2.000.000	
	Beban pajak	Rp. 400.000		Rp. 2.000.000		Rp. 400.000	
	Beban lain-lain	Rp. 2.000.000				Rp. 2.000.000	

3. *Progressive Web Application* (PWA)

a. Pengertian *Progressive Web Application* (PWA)

Progressive Web Application (PWA) adalah aplikasi web yang dibangun menggunakan teknologi web standar seperti HTML, CSS, dan JavaScript, namun menawarkan pengalaman pengguna yang mirip dengan aplikasi native. PWA dapat diinstal pada perangkat pengguna, berfungsi secara offline, dan terintegrasi dengan fitur perangkat seperti notifikasi push.

b. Karakteristik Utama PWA

PWA memiliki beberapa karakteristik yang membedakannya dari aplikasi web tradisional:

1) *Progressive*: Berfungsi untuk setiap pengguna, terlepas dari pilihan browser, karena dibangun dengan prinsip progressive enhancement.
2) *Responsif*: Tampilan dan fungsionalitas menyesuaikan dengan berbagai ukuran layar dan perangkat.
3) *Connectivity Independent*: Menggunakan service worker untuk bekerja secara offline atau pada jaringan yang tidak stabil.
4) *App-like Interactions*: Memberikan pengalaman pengguna yang mirip dengan aplikasi native melalui shell aplikasi yang terpisah dari konten.
5) *Fresh*: Selalu terbaru dengan pembaruan otomatis tanpa memerlukan interaksi pengguna.
6) *Safe*: Dikirim melalui HTTPS untuk memastikan keamanan dan integritas konten.
7) *Discoverable*: Dapat ditemukan oleh mesin pencari karena diidentifikasi sebagai "aplikasi" berkat manifest W3C dan pendaftaran service worker.
8) *Re-engageable*: Mendukung fitur seperti notifikasi push untuk meningkatkan keterlibatan pengguna.
9) *Installable*: Memungkinkan pengguna untuk menambahkan aplikasi ke layar utama mereka tanpa melalui toko aplikasi.
10) *Linkable*: Dapat dibagikan dengan mudah melalui URL tanpa proses instalasi yang rumit.

c. Komponen Utama PWA

Untuk mencapai karakteristik di atas, PWA memanfaatkan beberapa komponen kunci:

1) *Web App Manifest:* File JSON yang menyediakan informasi tentang aplikasi, seperti nama, ikon, dan warna tema, yang memungkinkan aplikasi ditampilkan dengan benar saat diinstal pada perangkat pengguna.
2) *Service Worker*: Skrip yang berjalan di latar belakang browser, memungkinkan fitur seperti caching, sinkronisasi latar belakang, dan notifikasi push, yang meningkatkan kinerja dan pengalaman pengguna.
3) *HTTPS*: Protokol yang memastikan komunikasi aman antara aplikasi dan server, mencegah serangan man-in-the-middle dan memastikan integritas data.

d. Keuntungan Menggunakan PWA

Penerapan PWA menawarkan berbagai keuntungan bagi pengembang dan pengguna:

1) Pengalaman Pengguna yang Lebih Baik: Dengan responsivitas tinggi dan kemampuan offline, pengguna mendapatkan pengalaman yang lebih konsisten dan andal.
2) Pengurangan Biaya Pengembangan: Satu kode basis dapat digunakan di berbagai platform, mengurangi waktu dan biaya pengembangan dibandingkan dengan aplikasi native.
3) Peningkatan Keterlibatan Pengguna: Fitur seperti notifikasi push dan kemampuan untuk diinstal meningkatkan interaksi dan retensi pengguna.

4) Kinerja yang Lebih Baik**:** Dengan caching yang efisien dan pengoptimalan lainnya, PWA menawarkan waktu muat yang lebih cepat dan penggunaan data yang lebih efisien.

e. Tantangan dalam Implementasi PWA

Meskipun memiliki banyak keuntungan, implementasi PWA juga menghadapi beberapa tantangan:

1) Dukungan Browser yang Beragam: Meskipun sebagian besar browser modern mendukung PWA, tingkat dukungan dan implementasi fitur dapat berbeda-beda.
2) Akses Terbatas ke Fitur Perangkat: Tidak semua fitur perangkat tersedia untuk PWA, terutama pada platform tertentu seperti iOS.
3) Kesadaran Pengguna: Beberapa pengguna mungkin tidak familiar dengan konsep PWA dan manfaatnya, sehingga edukasi dan promosi diperlukan.

4. Sistem Informasi

Sistem informasi didefinisikan sebagai kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi untuk mendukung operasi dan manajemen, atau sistem informasi didefinisikan sebagai kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi untuk mendukung operasi dan manajemen.

Menurut definisi sistem informasi sebelumnya, sistem informasi memiliki hubungan dengan sistem data saat ini di satu sisi dan sistem aktivitas di sisi lain.

Definisi Sistem Informasi Menurut para ahli :

a. Menurut John F.Nash (1995:8) Sistem informasi adalah kombinasi dari orang-orang, fasilitas atau teknologi, media, prosedur-prosedur, dan pengendalian yang mempunyai maksud untuk menata jaringan komunikasi yang penting, proses-proses dan transaksi-transaksi tertentu secara teratur, membantu manajemen dan para pemakain internal dan eksternal, dan menyediakan suatu dasar pengambilan keputusan yang tepat.
b. Menurut Robert A. Leitch (1999:11), sistem informasi adalah suatu sistem yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi yang bersifat manajerial dan kegiatan strategi dari suatu organisasi, dan menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu(Hasan, 2023).

5. Website

Website atau situs web adalah sekumpulan halaman web yang saling berhubungan dan dapat diakses secara online oleh perorangan, perusahaan, institusi pendidikan, pemerintahan, dan organisasi.

Dengan adanya website, informasi yang diinginkan akan mudah dan cepat ditemukan bagi yang membutuhkan, kemudian mudah didesiminasikan oleh pemilik informasi. Oleh karena itu, website juga dapat digunakan sebagai sarana bisnis yang baik, di samping sebagai sarana hiburan.

Beberapa contoh website yang banyak dikenal orang antara lain Google (www.google.co.id), Facebook (www.facebook.com), Twitter (www.twitter.com), Shopee (www.shopee.co.id), Tokopedia (www.tokopedia.com), Traveloka (traveloka.co.id), Netflix (www.netflix.com), dan sebagainya.

Beberapa manfaat website sebagai berikut.

a. Sumber informasi

   Website yang selalu terhubung online 24 jam sehari nonstop memungkinkan informasi dapat diakses kapan pun dan di mana pun, memudahkan user mendapatkan informasi. Terlebih untuk para pegiat bisnis dan akademisi, informasi yang cepat dan dapat dipercaya ini sangatlah dibutuhkan.

b. Sarana promosi

   Dibandingkan cara konvensional yang menggunakan kertas cetak untuk promosi, seperti membuat pamflet atau poster, kehadiran website dapat menjadi sarana promosi yang praktis dan murah.

c. Tampil profesional

   Dengan memiliki website, sebuah perusahaan atau instansi akan terlihat lebih serius dan siap terhadap produk atau jasa yang ditawarkan.

d. Meningkatkan bisnis

   Website dapat meningkatkan peluang bisnis dengan hadirnya SEO (search engine optimization), layanan pelanggan, layanan ketersedian produk dan jasa, serta sebagainya yang tentunya akan memudahkan calon pelanggan untuk membeli produk atau jasa.

e. Sarana menuangkan ide, gagasan, portofolio, atau bahkan untuk sekadar cerita(Achmad, L.G.A, 2019).

6. Pemrograman Web

Pemrograman web adalah proses pembuatan dan pengembangan aplikasi atau situs web yang dapat diakses melalui jaringan internet menggunakan

browser. Pemrograman web mencakup berbagai aktivitas, mulai dari merancang antarmuka pengguna (user interface), menulis kode untuk fungsi-fungsi tertentu, hingga mengelola basis data di server untuk menyimpan dan memproses data.

Menurut Welling & Thomson (2021), pemrograman web dapat dibagi menjadi dua kategori utama:

a. *Client-side Programming* (Pemrograman Sisi Klien): Fokus pada elemen yang dilihat dan digunakan oleh pengguna langsung, seperti HTML, CSS, dan JavaScript.
b. *Server-side Programming* (Pemrograman Sisi Server): Menangani logika aplikasi dan interaksi dengan basis data, menggunakan bahasa seperti PHP, Python, atau Java.

Pemrograman web bertujuan untuk menciptakan aplikasi web yang interaktif, responsif, dan fungsional, yang mendukung berbagai kebutuhan seperti e-commerce, sistem informasi, hingga media sosia (Adi Nugroho, 2020).

7. PHP (*Hypertext Preprocessor*)

a. Pengertian PHP

PHP (Hypertext Preprocessor) adalah bahasa pemrograman server-side yang dirancang untuk pengembangan aplikasi web. PHP digunakan untuk membuat halaman web dinamis yang dapat berinteraksi dengan database, menerima input dari pengguna, dan menampilkan informasi yang dihasilkan oleh server.

PHP awalnya dikembangkan oleh Rasmus Lerdorf pada tahun 1994 dan sejak itu menjadi salah satu bahasa pemrograman populer untuk pengembangan web. (Sumber: PHP Official Website)

b. Kelebihan PHP

1) Gratis dan Open Source: PHP tersedia secara bebas dan dapat digunakan tanpa biaya.
2) Kompatibilitas Platform: Dapat dijalankan di berbagai platform seperti Windows, macOS, dan Linux.
3) Integrasi Database: Mendukung berbagai sistem manajemen basis data seperti MySQL, PostgreSQL, dan SQLite.
4) Komunitas Besar: PHP memiliki komunitas pengembang yang luas, sehingga dokumentasi dan dukungan sangat mudah ditemukan.
5) Performa Tinggi: PHP dirancang untuk menangani permintaan server dengan cepat dan efisien

8. HTML

a. Pengertian HTML

HyperText Markup Language (HTML) adalah bahasa markah standar yang digunakan untuk membuat dan menyusun halaman web. HTML memungkinkan pengembang untuk menentukan struktur konten melalui elemen-elemen seperti paragraf, heading, tautan, gambar, dan lainnya. Dengan menggunakan tag-tag tertentu, HTML membantu browser menampilkan konten sesuai struktur yang diinginkan.(Ardi Rahmawan, 2023).

b. Sejarah Perkembangan HTML

HTML pertama kali diperkenalkan oleh Tim Berners-Lee pada tahun 1991 sebagai bahasa sederhana untuk membuat dokumen yang dapat diakses melalui internet. Seiring waktu, HTML mengalami beberapa pembaruan signifikan:

1) HTML 2.0 (1995): Merupakan standar resmi pertama yang mencakup fitur-fitur dasar untuk pembuatan halaman web.
2) HTML 3.2 (1997): Menambahkan dukungan untuk elemen presentasi seperti tabel dan applet.
3) HTML 4.01 (1999): Memperkenalkan pemisahan antara konten dan presentasi melalui penggunaan Cascading Style Sheets (CSS).
4) HTML5 (2014): Versi terbaru yang membawa banyak fitur baru dan peningkatan untuk pengembangan web.(BALAIWEB.COM)

c. Fitur Utama HTML5

1) HTML5 memperkenalkan berbagai fitur dan elemen baru yang meningkatkan kemampuan dan fleksibilitas dalam pengembangan web.
2) Elemen Semantik: Penambahan elemen seperti <header>, <footer>, <article>, dan <section> membantu dalam memberikan makna yang lebih jelas pada struktur dokumen.
3) Multimedia: Dukungan bawaan untuk elemen <audio> dan <video> memungkinkan penyematan media tanpa memerlukan plugin eksternal.
4) Canvas: Elemen <canvas> memungkinkan penggambaran grafis 2D dan 3D secara dinamis langsung di browser.

5) Formulir yang Ditingkatkan: Penambahan tipe input baru seperti email, date, dan range, serta atribut seperti placeholder dan required, meningkatkan fungsionalitas formulir.
6) Dukungan Offline: Fitur seperti Application Cache dan Local Storage memungkinkan aplikasi web berfungsi tanpa koneksi internet dan menyimpan data secara lokal.

d. Implementasi HTML dalam Pengembangan Web

HTML berfungsi sebagai kerangka dasar dalam pengembangan halaman web. Dengan menggabungkan HTML dengan CSS untuk styling dan JavaScript untuk interaktivitas, pengembang dapat menciptakan situs web yang responsif, interaktif, dan user-friendly. Pemahaman mendalam tentang HTML5 memungkinkan pengembang memanfaatkan fitur-fitur terbaru untuk meningkatkan pengalaman pengguna dan fungsionalitas situs web. .(Ardi Rahmawan, 2023).

9. MySQL



Gambar 2. 7 *Logo MySQL*

a. Pengertian MySQL

MySQL adalah salah satu sistem manajemen basis data relasional (Relational Database Management System/RDBMS) yang bersifat open-source. MySQL dirancang untuk mengelola data secara efisien melalui struktur tabel yang terorganisir, memungkinkan pengguna untuk menyimpan, memproses, dan mengakses data dengan cepat dan aman. MySQL menggunakan Structured Query Language (SQL) sebagai bahasa standar untuk operasi basis data seperti penyisipan, pembaruan, dan penghapusan data (Nugroho, 2020).

b. Sejarah Perkembangan MySQL

MySQL pertama kali dikembangkan oleh MySQL AB pada tahun 1995 dan kemudian diakuisisi oleh Sun Microsystems pada tahun 2008, yang kemudian menjadi bagian dari Oracle Corporation pada tahun 2010. Hingga saat ini, MySQL terus dikembangkan dan diperbarui untuk memenuhi kebutuhan dunia teknologi yang semakin kompleks (Suryana, 2020).

c. Fitur Utama MySQL

Beberapa fitur utama yang menjadikan MySQL sebagai salah satu RDBMS terpopuler adalah:

1) **Open Source**: MySQL didistribusikan secara gratis di bawah lisensi GNU General Public License (GPL).
2) **Multithreading**: Mendukung pemrosesan paralel untuk menangani banyak permintaan sekaligus.

3) **Cross-Platform**: MySQL dapat dijalankan pada berbagai sistem operasi seperti Windows, Linux, dan macOS.
4) **Keamanan Tinggi**: Mendukung autentikasi pengguna dan enkripsi data untuk melindungi informasi sensitif.
5) **Skalabilitas**: Cocok untuk proyek kecil hingga aplikasi dengan kebutuhan basis data yang besar.
6) **Replikasi**: MySQL mendukung replikasi basis data, yang memungkinkan sinkronisasi data antara server utama dan server cadangan.
7) **Integrasi Mudah**: Mendukung banyak bahasa pemrograman seperti PHP, Python, Java, dan lainnya.

d. Penggunaan MySQL dalam Pengembangan Aplikasi Web

MySQL sering digunakan sebagai basis data utama dalam aplikasi web. Integrasinya dengan bahasa pemrograman seperti PHP dan framework web seperti Laravel membuatnya menjadi pilihan utama untuk proyek pengembangan web. MySQL memungkinkan pengelolaan data secara efisien, termasuk pembuatan sistem login, manajemen produk, dan pengelolaan konten dinamis. (Suryana, 2020).

10. Flowchart

a. Pengertian *Flowchart*

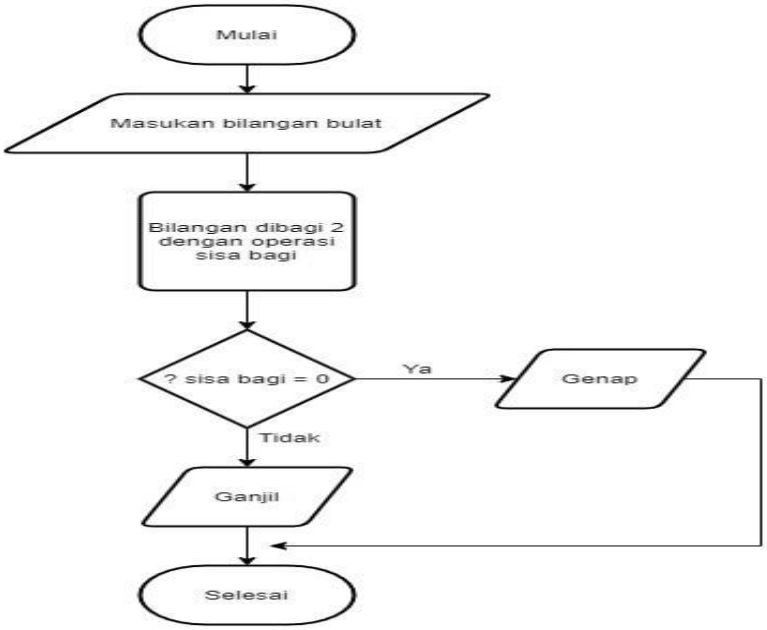
Flowchart adalah sebuah diagram yang menggambarkan alur proses atau alur logika suatu sistem. *Flowchart* menggunakan simbol-simbol standar untuk menunjukkan aktivitas, kondisi, dan alur logika dari proses yang digambarkan. *Flowchart* bisa digunakan dalam berbagai bidang, seperti pengembangan

perangkat lunak, perencanaan bisnis, manajemen proyek, desain sistem, dan lain-lain.

Flowchart dapat membantu dalam menentukan alur proses, menemukan kesalahan atau kekurangan dalam sistem, dan meningkatkan efisiensi suatu proses. *Flowchart* juga dapat digunakan untuk menjelaskan bagaimana suatu sistem bekerja pada orang yang tidak familiar dengan sistem tersebut. *Flowchart* dapat dibuat menggunakan *software* khusus atau dibuat secara manual menggunakan simbol-simbol standar(Awaluddin, 2022).

b. Contoh *Flowchart*

Untuk membuat sebuah flowchart sederhana kamu diharuskan untuk mengetahui setiap simbol dan juga fungsinya. Nah, di bawah ini saya akan memberikan sebuah contoh flowchart sederhana untuk menentukan apakah bilangan yang dimasukan ganjil atau genap. Berikut adalah contohnya:



Gambar 2.8 Contoh *Flowchart*
Sumber: https://www.dicoding.com/blog/flowchart

11. Metode Pengembangan Perangkat Lunak

a. Metode Pengumpulan Data

Pengumpulan data bertujuan untuk melengkapi dan mendukung materi dalam uraian pembahasan. Pengumpulan data dilakukan guna mendapatkan data yang real sesuai refrensi yang diperlukan. Metode ini meliputi studi pustaka, observasi dan wawancara.

b. Prototype

Metode yang digunakan dalam pengembangan sistem ini adalah prototype, yaitu metode pengembangan perangkat lunak yang memungkinkan adanya interaksi antara pengembang sistem dengan pengguna sistem, sehingga dapat mengatasi ketidakserasian antara pengembang dan pengguna. Model prototype merupakan salah satu model SDLC yang mempunyai ciri khas sebagai model proses evolusioner. Prototype sendiri bertujuan agar pengguna dapat memahami alur proses sistem dengan tampilan dan simulasi yang terlihat siap digunakan. Model prototype dapat dilihat pada gambar 2.1 dibawah ini:



Gambar 2.9 *Prototype* Paradigma

Tahap-tahap pada model prototype diatas adalah sebagai berikut:

1) Mendengarkan Pelanggan Tahap ini dilakukan pengumpulan data untuk memenuhi kebutuhan sistem dengan cara mendengar keluhan dari pelanggan. Analisis kebutuhan pemakai digunakan untuk membuat suatu sistem yang sesuai kebutuhan, maka harus diketahui terlebih dahulu bagaimana sistem yang sedang berjalan untuk kemudian mengetahui masalah yang terjadi melalui studi pustaka, observasi, wawancara.
2) Merancang dan Membuat Prototype Tahap ini melakukan pembuatan dan perancangan sistem prototype serta perancangan antarmuka (User Interface). Prototype yang akan dibuat disesuaikan dengan kebutuhan sistem yang telah didefinisikan sebelumnya dari keluhan pelanggan atau pengguna.
3) Uji Coba Prototype Tahap ini prototype yang telah dirancang akan diuji coba. Kemudian akan dievaluasi kekurangan-kekurangan apa saja yang terdapat pada sistem. Pengembang kemudian memperbaiki sistem sesuai kekurangan sampai dihasilkan sistem yang benar-benar dikehendaki oleh pengguna (Melisa Adhari, 2021).

c. UML (*Unified Modelling Language*)

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk *visualisasi*, merancang dan mendokumentasikan sistem piranti lunak. *UML* menawarkan sebuah standar untuk merancang model sebuah sistem. *UML* adalah sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap seb uah sistem atau perangkat

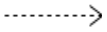
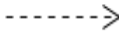
lunak berbasis objek. *UML* merupakan singkatan dari *Unified Modeling Language*. *UML* juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan(Adi Nugroho, 2020).

d. *Use Case Diagram*

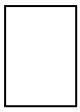
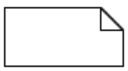
Diagram *Use Case* menjelaskan kapabilitas sistem yang diharapkan. Penekanannya adalah pada "apa" yang dilakukan sistem daripada "bagaimana". Sebuah Use Case menggambarkan interaksi aktor dan sistem. Kasus penggunaan adalah pekerjaan tertentu, seperti masuk ke sistem atau membuat daftar belanja. Aktor adalah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan tugas tertentu(Gilang Ramadhan, 2023).

Di antara simbol-simbol *Use Case Diagram* adalah:

Tabel 2. 1 *Simbol* Usecase *Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna *Main*kan ketika berinteraksi dengan *Use Case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *Use Case* sumber secara *eksplisit*.

Lanjutan Tabel

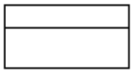
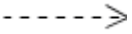
NO	GAMBAR	NAMA	KETERANGAN
5		*Extend*	Menspesifikasikan bahwa *Use Case* target memperluas perilaku dari *Use Case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

e. Class Diagram

Class diagram adalah jenis diagram struktur statis dalam UML yang menggambarkan struktur sistem dengan menunjukkan sistem *class*, atributnya, metode, dan hubungan antar objek. *Class diagram* disebut jenis diagram struktur karena menggambarkan apa yang harus ada dalam sistem yang dimodelkan dengan berbagai komponen.

Adapun simbol-simbol *Class Diagram* antara lain :

Tabel 2. 2 Simbol *Class* Diagram

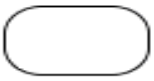
NO	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya

f. *Activity Diagram*

Menyatakan bahwa diagram aktivitas atau *activity diagram* menggambarkan *workflow* (alir kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

Adapun simbol-simbol *Activity Diagram* antara lain :

Tabel 2.3 Simbol *Activity* Diagram

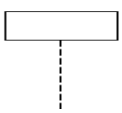
No.	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Activity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

g. Sequence Diagram

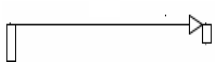
Menyatakan bahwa sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasi komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu obyek-obyek tersebut kemudian diurutkan dari kiri kekanan, aktor yang menginisialisasi interaksi biasanya ditaruh di paling kiri dari diagram.

Adapun simbol-simbol *Sequence* Diagram antara lain :

Tabel 2.4 Simbol *Sequence* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.

Lanjutan Tabel

2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

h. *StateChart Diagram*

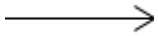
*Statechar*t diagram merupakan model perilaku yang dinamis dari class secara individual maupun beberapa bentuk dari obyek. Semua itu menunjukkan deretan dari state yang dilakukan obyek melalui event yang menyebabkan sebuah transisi dari satu ke aktifitas yang lain, dan beberapa aksi yang menghasilkan dari satu state atau aktifitas yang berubah. Statechart diagram fokus pada state dan secara bentuk digunakan untuk memodelkan tahapan yang nampak terpotong-potong dari sebuah aktivitas yang kontinyu dari obyek. Diagram ini memiliki hubungaan yang tetrtutup dengan activity diagram. Activity diagram fokus pada aktifitas dan pada pemodelan deretan aktivitas dari proses.

Adapun simbol-simbol *StateChart Diagram* antara lain :

Tabel 2.5 Simbol *StateChart* Diagram

No.	GAMBAR	NAMA	KETERANGAN
1		*State*	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		*Initial Pseudo State*	Bagaimana objek dibentuk atau diawali

Lanjutan Tabel

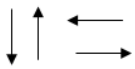
3		*Final State*	Bagaimana objek dibentuk dan dihancurkan
4		*Transition*	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		*Node*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

i. *Flowchart*

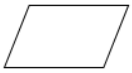
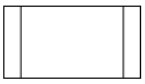
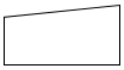
Flowchart adalah diagram dengan simbol-simbol yang merinci urutan proses dan hubungan antara proses (instruksi) dan proses lain dalam suatu program(Awaluddin, 2022).

Beberapa simbol yang digunakan dalam membuat flowchart adalah sebagai berikut:

Tabel 2.6. Simbol *Flowchart*

Simbol	Nama	Fungsi
	*Terminal Point Symbol*	*Terminal Menu*njukkan permulaan (start) atau akhir (stop) dari suatu proses.
	*Flow Direction Symbol*	Adalah simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain (*connecting line*). Simbol ini juga berfungsi untuk *Menu*njukkan garis alir dari proses.

Lanjutan Tabel

Simbol	Nama	Fungsi
	*Processing Symbol*	Digunakan untuk *Menu*njukkan kegiatan yang dilakukan oleh komputer. Pada bidang industri (proses produksi barang), simbol ini menggambarkan kegiatan inspeksi atau yang biasa dikenal dengan simbol inspeksi
	*Decision Symbol*	Merupakan simbol yang digunakan untuk memilih proses atau keputusan berdasarkan kondisi yang ada. Simbol ini biasanya ditemui pada *Flowchart* program.
	I*nput-Output*	Masuk menunjukkan proses *input-output* yang terjadi tanpa bergantung dari jenis peralatannya.
	*Predefined Process*	*Terdefinisi*merupakan simbol yang digunakan untuk *Menu*njukkan pelaksanaan suatu bagian prosedur (*sub*-proses).
	*Connector (On-page)*	Simbol ini fungsinya adalah untuk menyederhanakan hubungan antar simbol yang letaknya berjauhan atau rumit bila dihubungkan dengan garis dalam satu halaman
	*Connector (Off-page)*	Sama seperti *on-page connector*, hanya saya simbol ini digunakan untuk menghubungkan simbol dalam halaman berbeda. label dari simbol ini dapat menggunakan huruf atau angka
	*Preparation Symbol*	Merupakan simbol yang digunakan untuk mempersiapkan penyimpanan di dalam storage.
	*Manual Input Symbol*	Digunakan untuk menunjukkan input data secara manual menggunakan online keyboard.

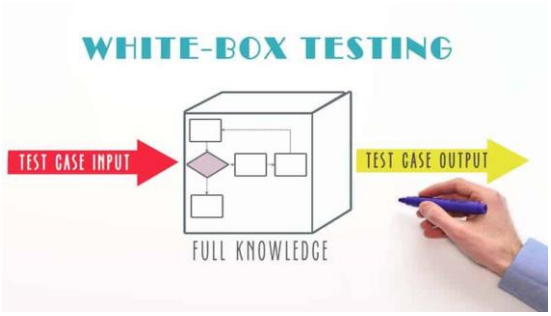
Lanjutan Tabel

	*Manual Operation Symbol*	Yang digunakanuntuk menunjukkankegiatan/proses yang tidak dilakukan oleh komputer.
	*Document Symbol*	Jika Anda menemukan simbol ini artinya input berasal dari dokumen dalam bentuk kertas, atau output yang perlu dicetak di atas kertas.
	*Multiple Documents*	Sama seperti *document symbol* hanya saja dokumen yang digunakan lebih dari satu dalam simbol ini
	*Display Symbol*	Adalah simbol yang menyatakan penggunaan peralatan output, seperti layar monitor, printer, plotter dan lain sebagainya
	*Delay Symbol*	Sesuai dengan namanya digunakan untuk *Menu*njukkan proses delay (*Menu*nggu) yang perlu dilakukan. Seperti *Menu*nggu surat untuk diarsipkan

12. Metode Pengujian

a. White Box

White box testing adalah suatu pengujian oleh software atau aplikasi dengan melihat modul untuk menganalisis kode program ada yang salah atau tidak. Bila suatu modul yang diproduksi tidak memenuhi syarat, kode akan dikompilasi ulang dan periksa lagi hingga mencapai apa yang diharapkan dalam arti bahwa white box testing menguji dengan cara melihat pure code dari suatu perangkat lunak atau aplikasi yang diuji tanpa mempedulikan tampilan atau UI dari aplikasi tersebut.



Gambar 2.10 *White Box Testing*

Sumber: https://folarium.co.id/id/blogs/perbedaan-white-box-dan-black-box-pada-software-testing

1) Teknik

Pada white box testing ada beberapa teknik yang digunakan. Berikut akan kami sajikan :

a) Basis path testing adalah metode yang memungkinkan perancang tast case untuk membuat pengukuran kompleksitas logika dari rancangan prosedural dan menggunakan pengukuran untuk mendefinisikan himpunan basis dari jalur eksekusi.

b) Flow graph merupakan notasi sederhana guna mempresentasi control flow.

c) Cyclomatic complexity yang digunakan guna mengetahui jumlah path yang perlu dicari. Di mana cyclomatic complexity merupakan metric perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logika program.

d) Branch coverage merupakan pengujian yang didesain agar masing-masing branch code yang paling tidak diuji satu kali.

e) Loop testing merupakan testing pada berbagai looping/perulangan yang terdapat pada program, seperti for, while, dan do-while. Dengan pengujian

ini Anda bisa mengecek kondisi perulangan, sudahkah berjalan dengan benar ataupun tidak.

f) Condition coverage merupakan teknik pengujian yang bertujuan guna melakukan uji semua kode untuk menghasilkan value FALSE atau TRUE. Tester bisa memastikan software dalam beroperasi dan menghasilkan output sesuai input user.

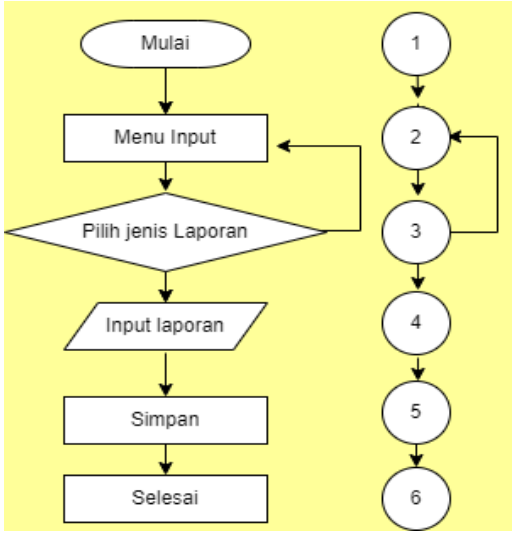
g) Multiple condition coverage merupakan teknik yang digunakan guna menguji semua kombinasi beragam kode yang digunakan dalam berbagi kondisi.

h) Statement coverage merupakan teknik yang minimal 1 kali dilakukan guna menguji masing-masing statement pada software. Anda bisa mengetahui beberapa code yang error dengan pengujian ini sehingga bisa segera memperbaikinya.

2) Trik Pengujian

White box testing merupakan pengujian dengan dasar pada pengecekan di dalam detail desain dari suatu rancangan pemrograman untuk membagi pengujian. Dalam membuat pengujian white box maka tester perlu memiliki pemahaman terkait pemrograman. Sebab untuk memahami kode sumber yang dipelajari dan harus memiliki pandangan global tentang fungsi aplikasi serta elemen yang dirancangnya. Metode white test ini dapat diterapkan untuk tes integrasi, tes unit dan tes sistem.

3) Contoh White Box



Gambar 4.11 Contoh *White box*

Hasan (2023) Pengujian white box adalah pengujian yang didasarkan pada pengecekan detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Pemetaan flowchart ke dalam flowgraph dan proses perhitungan V(G) terhadap perangkat lunak dapat dilihat pada penjelasan berikut :

Diketahui :

N = 6 E = 6 R = 2

Penyelasaian :

V(G) = (E- N) + 2

= (6 – 6) + 2

= 2

Independent Path :

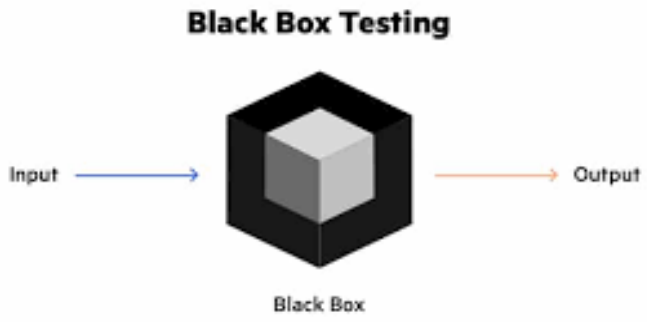
Path 1 = 1 – 2 – 3 – 4 – 5 – 6

Path 2 = 1 – 2 – 3 – 2 – 3 – 4 –5 – 6

Tabel 2.7. *White Box* Perhituangan

	1	2	3	4	5	6	E-1
1		1					1-1=0
2			1				1-1=0
3		1		1			2-1=1
4					1		1-1=0
5						1	1-1=0
6							0
SUM(E+1)							1+1=2

b. *Black Box*



Gambar 2.12 Black Box Testing

Sumber: https://folarium.co.id/id/blogs/perbedaan-white-box-dan-black-box-pada-software-testing

Pengujian black box adalah pengujian yang dilakukan semata-mata dengan memantau hasil eksekusi melalui data pengujian dan memvalidasi fungsionalitas perangkat lunak. Jadi, mirip dengan melihat kotak hitam, kita hanya bisa melihat tampilan luarnya saja dan tidak tahu apa yang ada di balik bungkus hitamnya. Mirip dengan pengujian black box, hanya mengevaluasi tampilan luar (antarmuka) dan fungsionalitasnya. Tanpa mengetahui apa yang terjadi dalam proses detail (hanya mengetahui input dan output).

1) Jenis Pengujian

Pada suatu black box testing terdapat tiga jenis pengujian, berikut akan kami jelaskan :

a) Functional testing

Merupakan suatu proses pengujian pada fitur spesifik atau fungsi pada suatu software. Pengujian ini bisa dilakukan guna memastikan bahwa user tidak dapat masuk pada aplikasi yang diuji tanpa informasi sehingga program tersebut terjaga keamanannya.

b) Non functional testing

Memiliki fungsi yang tidak kalah dari functional testing, di mana Non functional testing merupakan suatu pengujian pada beberapa aspek tambahan non fungsional. Bila functional testing menguji software menjalankan suatu tugas atau peringkat.

c) Regression testing

Jenis ini merupakan pengujian yang dilakukan pada software untuk mengamati apakah terjadinya kemunduran atau regresi ketika software diperbaharui. Dalam regression test dilakukan pengecekan aspek non fungsional maupun fungsional.

2) Teknik Menguji Black Box Dengan Software

Tulisan di atas sudah menjelaskan apa itu black box testing dan contohnya. Namun black box testing memiliki beberapa teknik dalam menguji suatu software. Berikut akan kami jelaskan :

a) Equivalence partitioning merupakan cara kerja dengan melakukan pembagian atau partition menjadi beberapa partisi dari input data.

b) Fuzzing merupakan teknik untuk mencari gangguan atau bug dari software dengan cara injeksi data yang terbilang sesi semi otomatis atau cacat.

c) Boundary value analysis merupakan teknik yang lebih fokus pada boundary di mana jika ada error dari sisi atau luar baik minimum maupun maksimum pada pada software yang didapat.

d) Orthogonal array testing merupakan teknik yang digunakan jika input domain yang relatif ukurannya kecil tetapi cukup berat jika digunakan dalam skala besar.

e) Cause effect graph merupakan teknik yang digunakan di mana graphic sebagai pacuannya. Dalam teknik ini di mana graphic menggambarkan relasi antara penyebab dan efek dari error tersebut.

f) State transition merupakan testing yang digunakan untuk mengecek apakah kondisi dari navigasi dan mesin dalam bentuk grafik.

g) All pair testing, pada teknik ini semua pasangan dari test case didesain agar bisa dieksekusi semua kemungkinan kombinasi diskrit berdasarkan input parameternya. Dengan teknik ini memiliki tujuan agar semua memiliki pasangan test case (Sumber : https://itbox.id/blog/black-box-testing-adalah/).

3. Contoh Pengujian Black Box

Pengujian Black Box didasarkan pada detail aplikasi, fungsi – fungsi yang ada pada aplikasi dan kesesuaian alur fungsi dengan proses yang diinginkan oleh pengguna atau user, pengujian ini tidak melihat dan menguji Logika program (Hasan, 2023).`

Tabel 2.7 Contoh Tabel Pengujian *Black Box*

Test Faktor	Hasil	Kesimpulan
Login	✔	Sukses, karena ketika *users* memasukkan *user* ID dan *Password*, akan masuk di halaman Utama
Screen Shot		
		

B. Kajian Hasil Penelitian Terdahulu

Dalam penelitian ini, mengacu pada penelitian terdahulu yang relevan dengan penelitian yang akan dilakukan. Berikut beberapa hasil penelitian yang relevan atau berhubungan dengan penelitian yang dijadikan bahan telaah bagi penelitian yang akan dilakukan, antara lain :

1. Penelitian yang dilakukan oleh: Hasan (2023), Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Parepare yang berjudul “Sistem Informasi Manajemen PT Calidad Coco Indonesia”. PT Candida Coco Indonesia perusahaan yang bergerak di bidang pembuatan turunan kelapa. Berdasarkan latar belakang tersebut maka tujuan dari

penelitian ini Untuk mengefesiensikan waktu dalam pengelolaan data perusahaan dan menghindari pengelolaan data yang kurang akurat.

2. Penelitian yang dilakukan oleh: Muh. Ade Candra (2017) Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Parepare yang berjudul, "Aplikasi Sistem Evaluasi Dan Pelaporan Alokasi Dana Zis Pada Baznas Berbasis Web" Badan Amil Zakat Nasional (BAZNAS) Kabupaten Pinrang adalah organisasi/ lembaga pengelolaan zakat di Kabupaten Pinrang yang memiliki tugas menghimpun dan menyalurkan zakat, infaq/ sedekah (ZIS). Sejalan dengan itu saat ini masih menggunakan sistem terkomputerisasi aplikasi pengolah kata (word) dan aplikasi pengolah angka (excel) untuk pengelolaan data penerima dana zakat, infaq dan sedekah (ZIS), mulai dari penginputan pengajuan layanan kemudian perhitungan dana yang sudah diterima oleh penerima manfaat, hingga proses pengelolaan pelaporan pendayagunaan dananya. Pengolahan data yang masih menggunakan metode komputerisasi ini masih memiliki beberapa kekurangan dan kelemahan seperti tidak efektif dalam mengolah data-data dalam jumlah yang banyak, resiko data rusak dan hilang. Penelitian bertujuan untuk merancang dan membangun suatu inovasi layanan digital dalam mempermudah Baznas dalam melaksanakan pengelolaan zakat, infaq dan sedekah (ZIS) dengan pemanfaatan suatu Aplikasi.

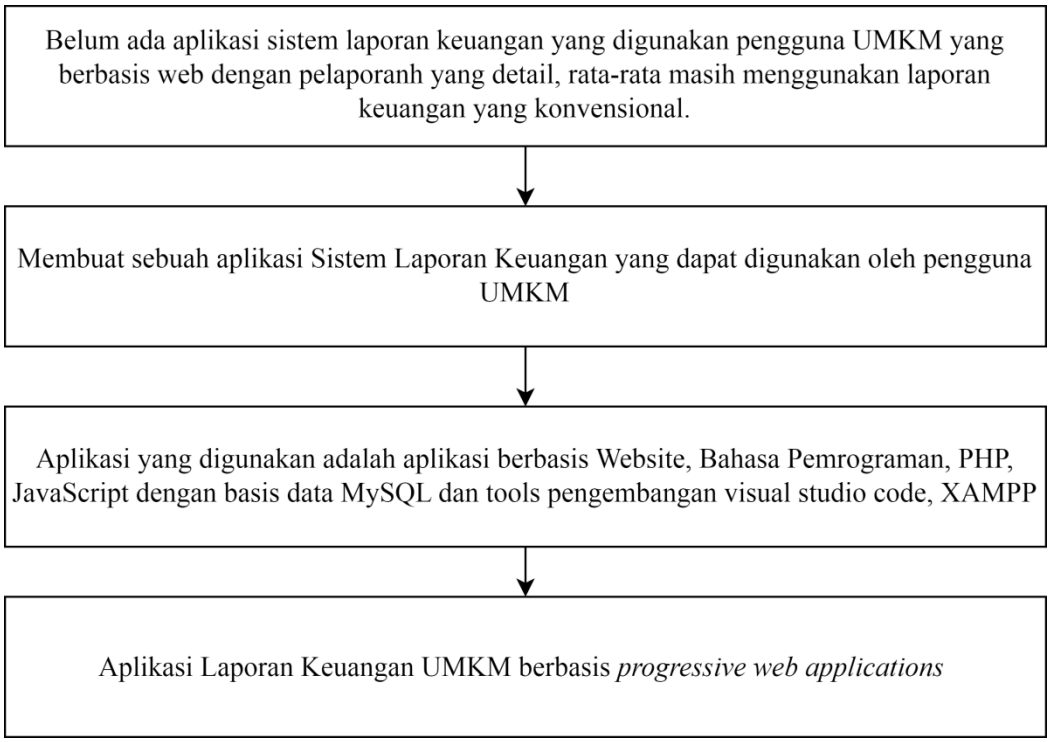
Perbandingan dari penelitian sebelumnya adalah Dari penelitian Hasan (2023). Sistem Informasi Manajemen PT Calidad Coco Indonesia, Merupakan aplikasi system informasi manajemen dengan laporan yang detail, sama halnya

dengan penelitian Muh. Ade Candra (2017) Aplikasi Sistem Evaluasi Dan Pelaporan Alokasi Dana Zis Pada Baznas Berbasis Web, juga merupakan aplikasi pelaporan dan aplikasi yang ingin dirancang adalah aplikasi Sistem Laporan Keuangan UMKM.

Kesimpulan dari penelitian terdahulu dan penelitian yang akan dibuat berbeda dari lokasi penelitian dan teknologi yang digunakan, teknologi yang digunakan pada penelitian ini adalah progressive web dengan kelebihan Instalasi Mudah Pengguna dapat menambahkan aplikasi ke layar utama perangkat mereka langsung dari browser tanpa perlu mengunduh melalui toko aplikasi. Akses Offline: Dengan menggunakan service worker, PWA dapat menyimpan aset dan data secara lokal. Notifikasi Push: PWA mendukung notifikasi push, memungkinkan pengembang untuk berkomunikasi secara real-time dengan pengguna, meningkatkan interaksi dan retensi pengguna. Responsif: Desain PWA menyesuaikan dengan berbagai ukuran layar dan perangkat, memastikan pengalaman pengguna yang konsisten.

C. Kerangka Pikir

Untuk memahami kerangka pikir penelitian ini dapat dilihat dibawah ini.



Gambar 2.13 Kerangka Pikir

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang digunakan dalam studi ini adalah penelitian waterfall. Metode waterfall merupakan metode terurut dan sistematis dalam melakukan suatu permodelan data. Dalam penelitian ini, metode waterfall terdiri dari studi literatur, analisis kebutuhan, perancangan sistem, implementasi dan pengujian sistem. Dengan melakukan permodelan yang terstruktur tersebut dapat menjadikan sistem akan tetap terjaga dan mampu bekerja dengan maksimal.

B. Waktu dan Tempat Penelitian

Penelitian ini dilakukan pada UMKM yang beroperasi di desa Anabannae, Kecamatan Pitu Riawa, Kabupaten Sidenreng Rappang, Provinsi Sulawesi Selatan. Objek penelitian mencakup pelaku usaha mikro, kecil, dan menengah yang telah menjalankan usaha mereka setidaknya selama dua tahun terakhir dan memiliki data keuangan yang dapat diakses. Pemilihan lokasi ini didasarkan pada pertimbangan ketersediaan data dan relevansi lokasi terhadap tujuan penelitian.

Tabel 3. 1 Jadwal Penelitian

No	UraianKegiatan	Bulan 2025		
		Januari	Februari	Maret
1	*Studi Literatur*			
2	Perancangan aplikasi			
3	Pembuatan aplikasi			
4	Pengujian			

C. Alat dan Bahan Penelitian

Dalam melakukan penelitian, maka diperlukan alat dan bahan penelitian yang mendukung kegiatan penelitian tersebut. Alat dan bahan yang diperlukan antara lain:

1. Alat

a. Perangkat Keras (Hardware)

Perangkat keras yang digunakan selama proses penelitian yaitu Laptop HP dengan spesifikasi sebagai berikut :

1) Processor : AMD Athlon Gold 3150U with Radeon Graphics 2.40 GHz
2) RAM : 4,00 GB
3) SSD : 500 GB
4) Monitor : 14 Inch

b. Perangkat Lunak (Software)

Perangkat lunak yang digunakan dalam pembuatan aplikasi adalah sebagai berikut :

a. Sistem Operasi : Windows 10 Ultimate 64 bit
b. Bahasa Pemrograman : PHP
c. Database : Xampp

2. Bahan

Adapun bahan yang dibutuhkan pada penelitian ini adalah sebagai berikut.

a. Data UMKM – Data laporan keuangan UMKM yang digunakan sebagai sampel untuk pengujian sistem.

b. Teknologi Sistem Informasi: Referensi mengenai sistem informasi laporan keuangan, perangkat lunak untuk UMKM. Literatur Pendukung: Jurnal, buku, dan penelitian sebelumnya terkait sistem laporan keuangan UMKM.

D. Tahap Penelitian

Tahap penelitian dilakukan dengan cara :

1. Persiapan Penelitian

Persiapan penelitian yang dimaksud adalah menyiapkan buku-buku, artikel tentang topik penelitian serta software yang digunakan selama penelitian.

2. Teknik Pengumpulan Data

Data dikumpulkan menggunakan metode berikut:

a. Wawancara: Dilakukan dengan pemilik atau pengelola UMKM untuk memahami praktik penyusunan laporan keuangan.
b. Kuesioner: Disusun untuk mendapatkan informasi mengenai kemampuan pengelola UMKM dalam memahami dan menerapkan laporan keuangan.
c. Dokumentasi: Mengumpulkan dokumen laporan keuangan UMKM untuk dianalisis lebih lanjut..

3. Teknik Analisis Data

Pada Data yang terkumpul dianalisis menggunakan metode berikut:

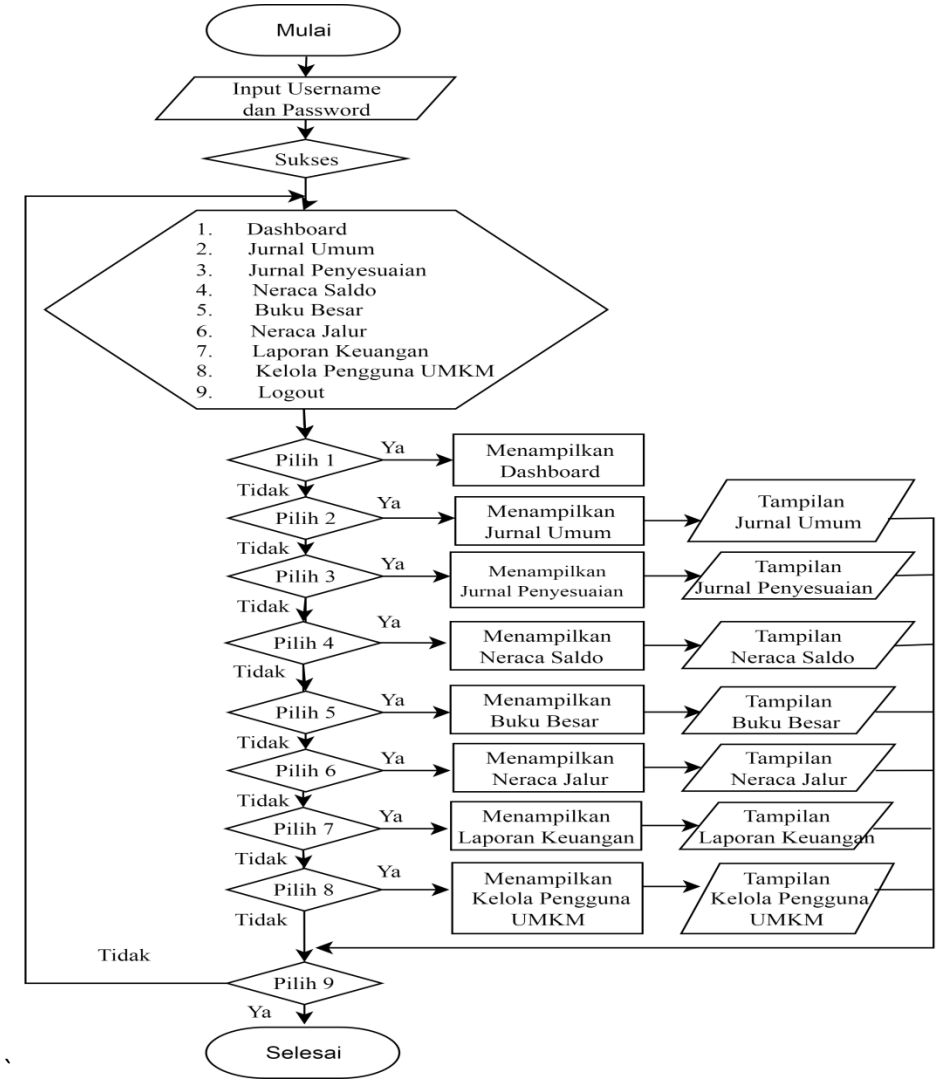
a. Analisis Deskriptif Statistik: Digunakan untuk memberikan gambaran umum mengenai data keuangan UMKM, seperti distribusi frekuensi, rata-rata, dan persentase.

b. Analisis Kualitatif: Digunakan untuk mengevaluasi kesesuaian laporan keuangan UMKM dengan prinsip akuntansi yang berlaku umum (SAK EMKM).

c. Analisis Komparatif: Membandingkan kualitas laporan keuangan antar UMKM berdasarkan ukuran usaha dan sektor industri.

E. Analisis Sistem

1. *Flowchart* Aplikasi

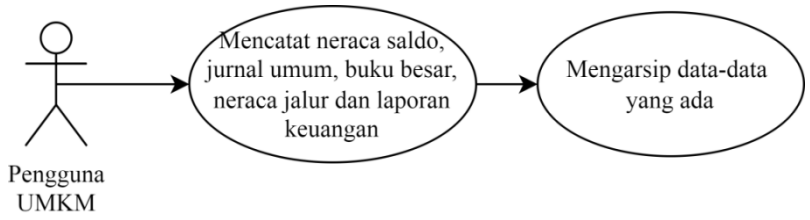


Gambar 3. 1 *Flowchart* Aplikasi

Pada gambar 3.1 yaitu *flowchart* dimana pada suatu *flowchart* pastinya diawali dengan mulai atau *start*. Kemudian menampilkan tampilan awal aplikasi laporan keuangan UMKM , apabila sudah punya akun maka lansung masuk ke halaman utama aplikasi, maka muncul semua menu aplikasi seperti neraca saldo, jurnal umum, buku besar, neraca jalur, dan laporan keuangan, kemudian selesai.

2. Sistem Yang Berjalan

Sistem saat ini digambarkan sebagai dekonstruksi sistem lengkap menjadi bagian-bagian komponennya dengan tujuan mendeteksi dan menganalisis kesulitan dan hambatan yang muncul dalam persyaratan yang diharapkan akan diajukan.



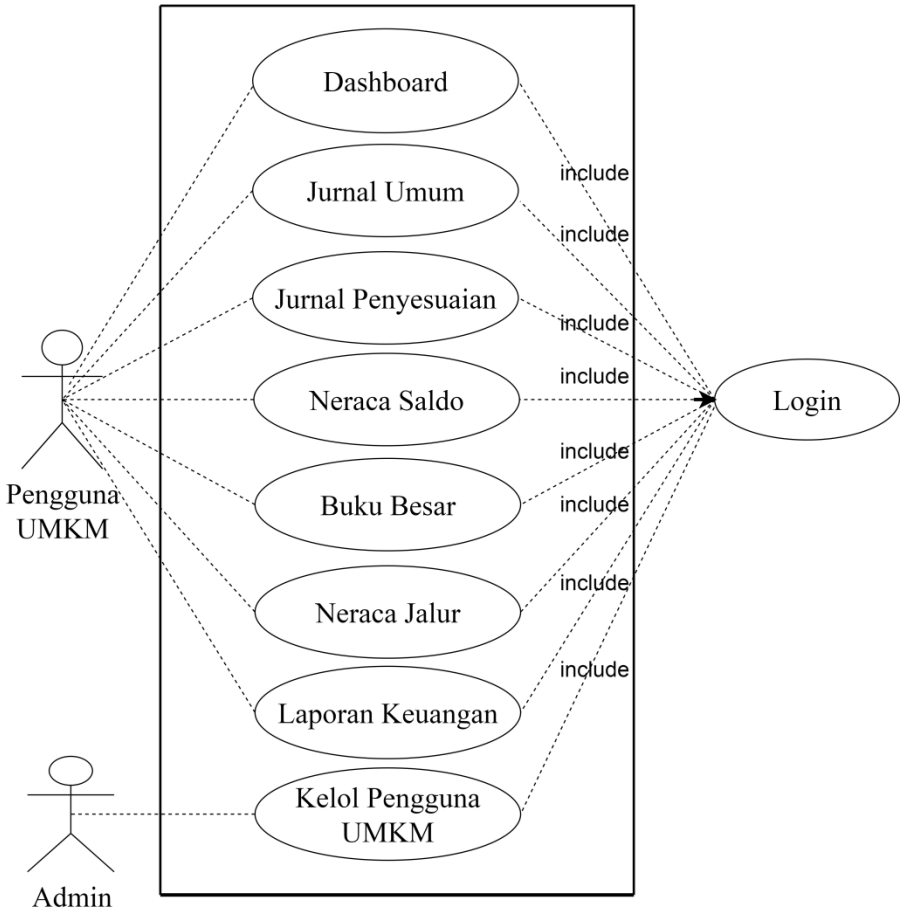
Gambar 3. 2 Sistem Yang Berjalan

Sedangkan untuk penjelasan definisi setiap use case pada sistem yang berjalan sebagai berikut :

Tabel 3.2. Definisi Sistem yang Berjalan

No	*Use Case*	Keterangan
1	Neraca Saldo	Mencatat neraca saldo kemudian mengarsip data neraca saldo
2	Jurnal Umum	Mencatat jurnal umum kemudian mengarsip data jurnal umum
3	Buku Besar	Mencatat buku besar kemudian mengarsip data buku besar
4	Neraca Jalur	Mencatat neraca jalur kemudian mengarsip data neraca jalur
5	Laporan Keuangan	Mencatat laporan keuangan kemudian mengarsip data laporan keuangan

3. Sistem Yang Diusulkan



Gambar 3.3 Sistem Yang Dusulkan

Penjabaran use case diagram di atas didefinisikan atas definisi aktor dan definisi use case pada sistem yang diusulkan. Definisi aktor sebagai berikut :

Tabel 3.3 Definisi Aktor dalam Sistem.

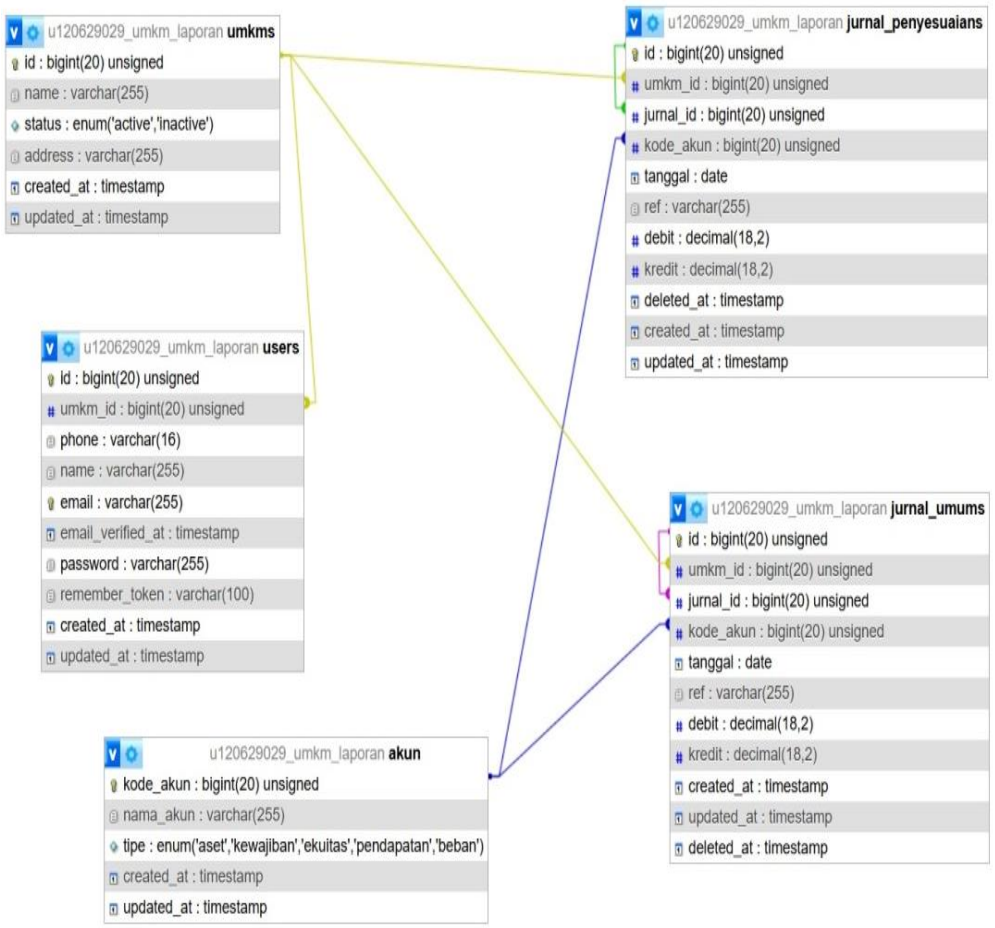
No	Aktor	Keterangan
1	Pengguna UMKM	Pengguna UMKM atau *User* merupakan aktor yang berperan sebagai pengguna aplikasi laporan keuangan UMKM
2	Admin	Admin merupakan aktor yang dapat ,mengelola pengguna UMKM

Sedangkan untuk penjelasan definisi setiap use case pada sistem yang diusulkan sebagai berikut :

Tabel 3.4 Definisi Use Case Sistem

No	*Use Case*	Keterangan
1	Login	Menampilkan halaman *login* kemudian masukkan *username* dan *password*
2	*Dashboard*	Menampilkan halaman utama aplikasi
3	Jurnal Umum	Menampilkan halaman Jurnal Umum
4	Jurnal Penyesuaian	Menampilkan halaman Jurnal Penyesuaian
5	Neraca Saldo	Menampilkan halaman Neraca Saldo
6	Buku Besar	Manampilkan halaman Buku Besar
7	Neraca Jalur	Manampilkan halaman Neraca Jalur
8	Laporan Keuangan	Manampilkan halaman Laporan Keuangan
9	Kelola pengguna UMKM	Menampilkan halaman kelola pengguna UMKM

4. Relasi Antar Tabel



Gambar 3.4 Relasi Antar Tabel

a. Tabel: umkms

- Kolom kunci utama: id
- Relasi:
- users: satu UMKM bisa memiliki banyak user (relasi satu ke banyak).
- jurnal_penyesuaians: satu UMKM bisa memiliki banyak jurnal penyesuaian.
- jurnal_umums: satu UMKM bisa memiliki banyak jurnal umum.

b. Tabel: users

- Kolom kunci utama: id
- Kolom foreign key: umkm_id -> mengacu ke umkms.id
- Relasi: satu user terkait dengan satu UMKM.

c. Tabel: akun

- Kolom kunci utama: kode_akun
- Digunakan di tabel:
- jurnal_penyesuaians.kode_akun
- jurnal_umums.kode_akun
-Relasi: satu akun bisa digunakan di banyak entri jurnal (penyesuaian maupun umum).

d. Tabel: jurnal_umums

- Kolom kunci utama: id
- Foreign keys:
- umkm_id -> ke umkms.id
- kode_akun -> ke akun.kode_akun
- Merupakan jurnal umum untuk masing-masing UMKM dan akun.

e. Tabel: jurnal_penyesuaians

- Kolom kunci utama: id
- Foreign keys:
- umkm_id -> ke umkms.id
- kode_akun -> ke akun.kode_akun

-Berisi jurnal penyesuaian yang berhubungan dengan UMKM dan akun tertentu..

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Hasil Penelitian

Selama melakukan penelitian di UMKM PT Ali Baba yang beroperasi di desa Anabannae, Kecamatan Pitu Riawa, Kabupaten Sidenreng Rappang, Provinsi Sulawesi Selatan, dalam membangun Aplikasi Sistem Laporan Keuangan Umkm, didapatkanlah hasil penelitian antara lain. Neraca saldo awal, Jurnal umum, Buku Besar, Neraca saldo akhir, Jurnal penyelesaian, Neraca jalur dan Laporan keuangan

Tabel 4.1. Neraca Saldo Awal .

NAMA AKUN	**NERACA SALDO**	
	DEBET	KREDIT
Kas	410,600,000	
Piutang	100,000,000	
Perlengkapan	10,000,000	
Sewa Dibayar Dimuka	50,000,000	
Peralatan	40,000,000	
Persediaan Barang Dagang	20,000,000	
Persediaan Bahan Baku	100,000,000	
Modal		650,000,000
Prive	300,000	
Penjualan		100,000,000
Beban Gaji	15,000,000	
Beban Listrik,Air, Dan Pulsa	2,000,000	
Beban Pajak	400,000	
Beban Lain Lain	2,000,000	
Jumlah	**750,300,000**	**750,000,000**

Tabel 4.2. Jurnal Umum

HARI/ TANGGAL	KETERANGAN	REF	DEBET	KREDIT
1/1/2024	Kas		650,000,000	
	Modal			650,000,000
1/2/2024	Persediaan Bahan		30,000,000	
	Kas			30,000,000
1/3/2024	Persediaan Bahan		20,000,000	
	Kas			20,000,000
1/4/2024	Sewa dibayar dimuka		50,000,000	
	Kas			50,000,000
1/7/2024	Peralatan		20,000,000	
	Kas			20,000,000
1/7/2024	Perlengkapan		5,000,000	
	Kas			5,000,000
1/8/2024	Persediaan Bahan		30,000,000	
	Kas			30,000,000
1/10/2024	Persediaan Bahan		20,000,000	
	Kas			20,000,000
1/11/2024	Peralatan		20,000,000	
	Utang Usaha			20,000,000
1/12/2024	Perlengkapan		5,000,000	
	Kas			5,000,000
1/13/2024	kas		2,800,000	
	Penjualan			2,800,000
1/14/2024	kas		20,000,000	
	Penjualan			20,000,000
1/14/2024	Persediaan Barang Dagang		20,000,000	
	Kas			20,000,000
1/15/2024	Kas		5,000,000	
	Penjualan			5,000,000
1/16/2024	Kas		500,000	
	Penjualan			500,000
1/17/2024	Kas		10,000,000	
	Penjualan			10,000,000
1/17/2024	Piutang dagang		100,000,000	
	Penjualan			100,000,000
1/18/2024	Kas		8,000,000	
	Penjualan			8,000,000
1/19/2024	Kas		2,000,000	
	Penjualan			2,000,000

Lanjutan Tabel

1/20/2024	Kas		10,000,000	
	Penjualan			10,000,000
1/21/2024	Kas		20,000,000	
	Penjualan			20,000,000
1/22/2024	Kas		10,000,000	
	Penjualan			10,000,000
1/23/2024	Kas		10,000,000	
	Penjualan			10,000,000
1/24/2024	Kas		1,500,000	
	Penjualan			1,500,000
1/25/2024	Kas		10,000,000	
	Penjualan			10,000,000
1/26/2024	Kas		500,000	
	Penjualan			500,000
1/27/2024	Beban Gaji		15,000,000	
	Kas			15,000,000
1/28/2024	Beban Listrik, Air, dan Pulsa		2,000,000	
	Kas			2,000,000
1/29/2024	Beban Pajak		400,000	
	Kas			400,000
1/30/2024	Beban Lain lain		2,000,000	
	Kas			2,000,000
1/31/2024	Prive		300,000	
	Kas			300,000
1/31/2024	Peralatan		100,000,000	
	Utang Usaha			100,000,000
Jumlah			1,200,000,000	1,200,000,000

Tabel 4.2 Buku Besar Kas

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
1	Menyetor modal		Rp. 500.000.000		Rp. 500.000.000	
2	Membeli bahan			Rp. 30.000.000	Rp. 470.000.000	
3	Membeli bahan			Rp. 20.000.000	Rp. 450.000.000	
4	Membayar sewa			Rp. 10.000.000	Rp. 440.000.000	
7	Membeli peralatan			Rp. 20.000.000	Rp. 420.000.000	

Lanjutan Tabel 4.2 Buku Besar Kas

7	Membeli perlengkapan			Rp. 5.00.000	Rp. 415.000.000	
8	Membeli bahan			Rp. 30.000.000	Rp. 385.000.000	
10	Membeli bahan			Rp. 20.000.000	Rp. 365.000.000	
11	Membeli peralatan			Rp. 20.000.000	Rp. 345.000.000	
12	Membeli perlengkapan			Rp. 5.000.000	Rp. 340.000.000	

Tabel 4.3 Buku Besar Perlengkapan

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
7	Membeli perlengkapan		Rp. 5.000.00		Rp. 5.000.000	
12	Membeli perlengkapan		Rp.5.000.000		Rp. 5.000.000	

Tabel 4.4 Buku Besar Peralatan

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
7	Membeli peralatan		Rp. 20.000.000		Rp. 20.000.000	
11	Membeli peralatan		Rp.20. 000.000		Rp. 40.000.000	

Tabel 4.5 Buku Besar Persediaan Barang Dagang

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
14	Membeli barang		Rp. 20.000.000		Rp. 20.000.000	

Tabel 4.6 Buku Besar Persediaan Bahan

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
2	Membeli bahan		Rp.30.000.00 0		Rp. 30.000.000	
3	Membeli bahan		Rp. 20.000.000		Rp. 50.000.000	
8	Membeli bahan		Rp. 30.000.000		Rp. 80.000.000	
10	Membeli bahan		Rp. 20.000.000		Rp. 100.000.000	

Tabel 4.7 Buku Besar Sewa dibaya dimuka

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
4	Membayar sewa		Rp. 10.000.000		Rp. 10.000.000	

Tabel 4.8 Buku Besar Modal

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
1	Menyetor modal			Rp. 500.000.000		Rp. 500.000.000

Tabel 4.9 Buku Besar Prive

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
31	Pengembalian pribadi			Rp. 300.000.000		Rp. 300.000.000

Tabel 4.10 Buku Besar Penjualan

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
13	Penjualan			Rp. 2.800.000		Rp. 2.800.000
14	Penjualan			Rp. 20.000.000		Rp. 22.800.00
15	Penjualan			Rp. 5.000.000		Rp. 27. 800.000
16	Penjualan			Rp. 500.000		Rp. 28.300.000

Tabel 4.11 Buku Besar Beban Gaji

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
2	Membeli bahan		Rp.30.000.000		Rp. 30.000.000	
3	Membeli bahan		Rp. 20.000.000		Rp. 50.000.000	

Tabel 4.12 Buku Besar Beban Listrik,Air, Dan Pulsa

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
7	Membeli peralatan		Rp. 20.000.000		Rp. 20.000.000	
11	Membeli peralatan		Rp.20. 000.000		Rp. 40.000.000	

Tabel 4.13 Buku Besar Beban Pajak

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
14	Membeli barang		Rp. 20.000.000		Rp. 20.000.000	

Tabel 4.14 Buku Besar Beban Lain-Lain

Tgl	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
2	Membeli bahan		Rp.30.000.000		Rp. 30.000.000	
3	Membeli bahan		Rp. 20.000.000		Rp. 50.000.000	
8	Membeli bahan		Rp. 30.000.000		Rp. 80.000.000	
10	Membeli bahan		Rp. 20.000.000		Rp. 100.000.000	

Tabel 4.15 Neraca Saldo Akhir

NAMA AKUN	NERACA SALDO	
	DEBET	KREDIT
Kas	520,600,000	
Piutang	100,000,000	
Sewa Dibayar Dimuka	50,000,000	
Persediaan Bahan Baku	100,000,000	
Persediaan Barang Dagang	20,000,000	
Perlengkapan	10,000,000	
Peralatan	140,000,000	
Utang Usaha		100,000,000
Modal		650,000,000
Prive	300,000	
Penjualan		210,300,000
Beban Gaji	15,000,000	
Beban Listrik,Air, Dan Pulsa	2,000,000	
Beban Pajak	400,000	
Beban Lain Lain	2,000,000	
Total	**960,300,000**	**960,300,000**

Tabel 4.16 Jurnal Penyesuaian

JURNAL PENYESUAIN			
	NAMA AKUN	**DEBET**	**KREDIT**
1	Biaya Perlengkapan	2,000,000	
	Perlengkapan		2,000,000
2	Biaya Penyusutan Peralatan	1,000,000	
	Peralatan		1,000,000
3	Biaya Sewa	10,000,000	
	Sewa dibayar dimuka		10,000,000
	JUMLAH	**13,000,000**	13,000,000

Tabel 4.17 Neraca Jalur

Kode Akun	**Nama Akun**	**Neraca Saldo**		**Laba Rugi**		**Naraca**	
		Debit	**Kredit**	**Debit**	**Kredit**	**Kredit**	**Debit**
	Kas	410.600.000				410.600.000	
	Perlengkapan	10.000.000				10.000.000	
	Peralatan	40.000.000				40.000.000	
	Persediaan barang dagang	20.000.000				20.000.000	
	Persediaan bahan baku	100.000.000				100.000.000	
	Sewa dibayar dimuka	10.000.000				10.000.000	
	Modal		500.000.000				500.000.000
	Prive	300.000			110.300.000	300.000	
	Penjualan		110.300.000	15.000.000			110.300.000
	Beban gaji	15.000.000		2.000.000		15.000.000	
	Beban Listrik, air dan pulsa	2.000.000		400.000		2.000.000	
	Beban pajak	400.000		2.000.000		400.000	
	Beban lain-lain	2.000.000				2.000.000	

Tabel 4.18 Laporan Keuangan Laba/Rugi

Penjualan					210,300,000
Beban gaji			15,000,000		
Beban listrik,air, dan pulsa			2,000,000		
Beban pajak			400,000		
Beban lain lain			2,000,000		
biaya perlengkapan			2,000,000		
biaya penyusutan peralatan			1,000,000		
biaya sewa			10,000,000		
JUMLAH BIAYA					32,400,000
LABA TAHUN BERJALAN					177,900,000

Tabel 4.19 Laporan Keuangan Perubahan Modal

Modal Awal					650,000,000
laba tahun berjalan					177,900,000
					827,900,000
prive					300,000
Modal Akhir					827,600,000

Tabel 4.20 Laporan Keuangan Neraca

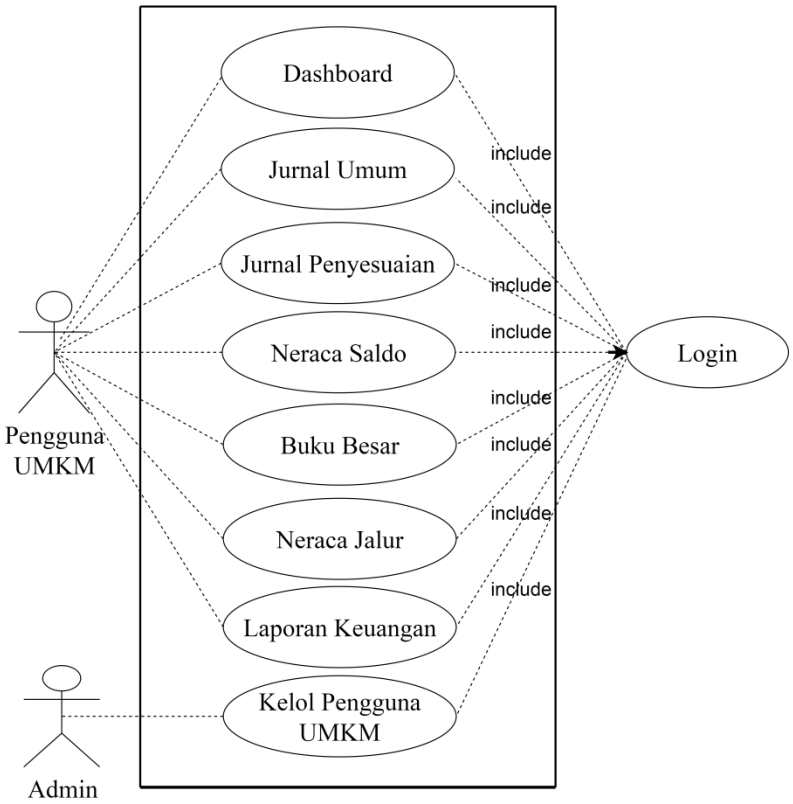
AKTIVA		**PASSIVA**	
kas	520,600,000	Utang usaha	100,000,000
Piutang	100,000,000		
Sewa dibayar dimuka	40,000,000		
persediaan bahan baku	100,000,000	Modal Akhir	827,600,000
persediaan barang dagang	20,000,000		
Perlengkapan	8,000,000		
Peralatan	139,000,000		
JUMLAH	927,600,000		927,600,000

B. Pembahasan

1. Analisis Aliran Data UML

a. *Use Case Diagram*

Use Case Diagram atau diagram *use case* merupakan pemodelan untuk memodelkan kelakuan (*behavior*) sistem informasi yang dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi. *Use case* juga dapat digunakan untuk merepresentasikan interaksi yang terjadi antara aktor dengan proses sistem yang dibuat. Pada perancangan *use case* juga terdapat skenario, yaitu langkah yang menerangkan urutan kejadian antara pengguna dengan sistem. Diagram *use case* pada Aplikasi ini digambarkan seperti berikut ini:



Gambar 4 1. *Use Case Diagram.*

b. Tabel *Use Case*

Penjabaran use case diagram di atas didefinisikan atas definisi aktor dan definisi use case pada sistem yang diusulkan. Definisi aktor sebagai berikut :

Tabel 4.21 Definisi Aktor dalam Sistem.

No	Aktor	Keterangan
1	Pengguna UMKM	Pengguna UMKM atau *User* merupakan aktor yang berperan sebagai pengguna aplikasi laporan keuangan UMKM
2	Admin	Admin merupakan aktor yang dapat ,mengelola pengguna UMKM

Sedangkan untuk penjelasan definisi setiap use case pada sistem yang diusulkan sebagai berikut :

Tabel 4.22 Definisi Use Case Sistem

No	*Use Case*	Keterangan
1	Login	Menampilkan halaman *login* kemudian masukkan *username* dan *password*
2	*Dashboard*	Menampilkan halaman utama aplikasi
3	Jurnal Umum	Menampilkan halaman Jurnal Umum
4	Jurnal Penyesuaian	Menampilkan halaman Jurnal Penyesuaian
5	Neraca Saldo	Menampilkan halaman Neraca Saldo
6	Buku Besar	Manampilkan halaman Buku Besar
7	Neraca Jalur	Manampilkan halaman Neraca Jalur
8	Laporan Keuangan	Manampilkan halaman Laporan Keuangan
9	Kelola pengguna UMKM	Menampilkan halaman kelola pengguna UMKM

Dalam use case yang telah didefinisikan di atas, terdapat beberapa use case yang memiliki alur proses yang hampir sama, sehingga dalam penjabaran selanjutnya akan didefinisikan secara ringkas untuk memudahkan pemahaman proses tiap use case. Berikut adalah skenario use case yang telah didefinisikan sebagai berikut :

1) Use Case Login

Aktor : Admin, Pengguna UMKM

Deskripsi : Sebuah kegiatan untuk masuk kedalam aplikasi.

Pre-condition : Admin, Pengguna UMKM berada di halaman login.

Post-condition : -

Tabel 4 23. Skenario *Use Case Login*

Aksi Aktor	Reaksi Sistem
1. Masuk kehalaman login dan Memasukkan Username dan Password.	
	2. Masuk ke halaman utama aplikasi

2) Use Case Home.

Aktor : Admin, Pengguna UMKM

Deskripsi : Sebuah kegiatan untuk menampilkan menu-menu aplikasi.

Pre-condition : Admin, Pengguna UMKM berada di halaman utama.

Post-condition : -

Tabel 4 24. Skenario *Use Case Home*.

Aksi Aktor	Reaksi Sistem
1. Masuk kehalaman utama aplikasi	
	2. Menampilkan menu-menu aplikasi.

3) Use Case Jurnal Umum

Aktor : Pengguna UMKM

Deskripsi : Sebuah kegiatan untuk menampilkan Jurnal Umum.

Pre-condition : Pengguna UMKM berada di halaman Jurnal Umum.

Post-condition : Data dalam sistem akan diperbarui.

Tabel 4 25. Skenario *Use Case* Jurnal Umum

Aksi Aktor	Reaksi Sistem
1. Memilih menu Weddi Jurnal Umum	
	2. Menampilkan form Jurnal Umum

4) Skenario Use Case Jurnal Penyesuaian

Aktor : Pengguna UMKM

Deskripsi : Sebuah kegiatan untuk melihat Jurnal Penyesuaian.

Pre-condition : Pengguna UMKM berada di halaman Jurnal Penyesuaian

Post-condition : Data dalam sistem akan diperbarui.

Tabel 4 26. Skenario Use Case Jurnal Penyesuaian

Aksi Aktor	Reaksi Sistem
1. Menekan menu V Jurnal Penyesuaian	
	2. Menampilkan data Jurnal Penyesuaian

5) Use Case Neraca Saldo

Aktor : Pengguna UMKM

Deskripsi : Sebuah kegiatan untuk melihat Neraca Saldo.

Pre-condition : Pengguna UMKM berada di halaman Neraca Saldo .

Post-condition : Data dalam sistem akan diperbarui.

Tabel 4 27. Skenario Use Case Neraca Saldo

Aksi Aktor	Reaksi Sistem
1. Menekan menu Neraca Saldo	
	2. Menampilkan menu Neraca Saldo

6) Use Case Buku Besar

Aktor : Pengguna UMKM

Deskripsi : Sebuah kegiatan untuk melihat Buku Besar.

Pre-condition : Pengguna UMKM berada di halaman Buku Besar

Post-condition : Data dalam sistem akan diperbarui.

Tabel 4 28. Skenario Use Case Buku Besar

Aksi Aktor	Reaksi Sistem
1.Menekan menu Buku Besar	
	1. Menampilkan menu Buku Besar

7) Use Case Neraca Jalur

Aktor : Pengguna UMKM

Deskripsi : Sebuah kegiatan untuk melihat Neraca Jalur

Pre-condition : Pengguna UMKM berada di halamanNeraca Jalur.

Post-condition : Data dalam sistem akan diperbarui.

Tabel 4 29. Skenario Use Case Neraca Jalur

Aksi Aktor	Reaksi Sistem
1.Menekan menu Neraca Jalur	
	2. Menampilkan menu Neraca Jalur

8) Use Case Laporan Keuangan

Aktor : Pengguna UMKM

Deskripsi : Sebuah kegiatan untuk melihat Laporan Keuangan.

Pre-condition : Pengguna UMKM berada di halaman Laporan Keuangan.

Post-condition : Data dalam sistem akan diperbarui.

Tabel 4 30. Skenario Use Case Laporan Keuangan

Aksi Aktor	Reaksi Sistem
1.Menekan menu Laporan Keuangan	
	2. Menampilkan menu Laporan Keuangan

9) Use Case Kelola Pengguna UMKM

Aktor : Pengguna UMKM

Deskripsi : Sebuah kegiatan untuk melihat Kelola Pengguna UMKM.

Pre-condition : Pengguna UMKM berada di halaman Kelola Pengguna UMKM.

Post-condition : Data dalam sistem akan diperbarui.

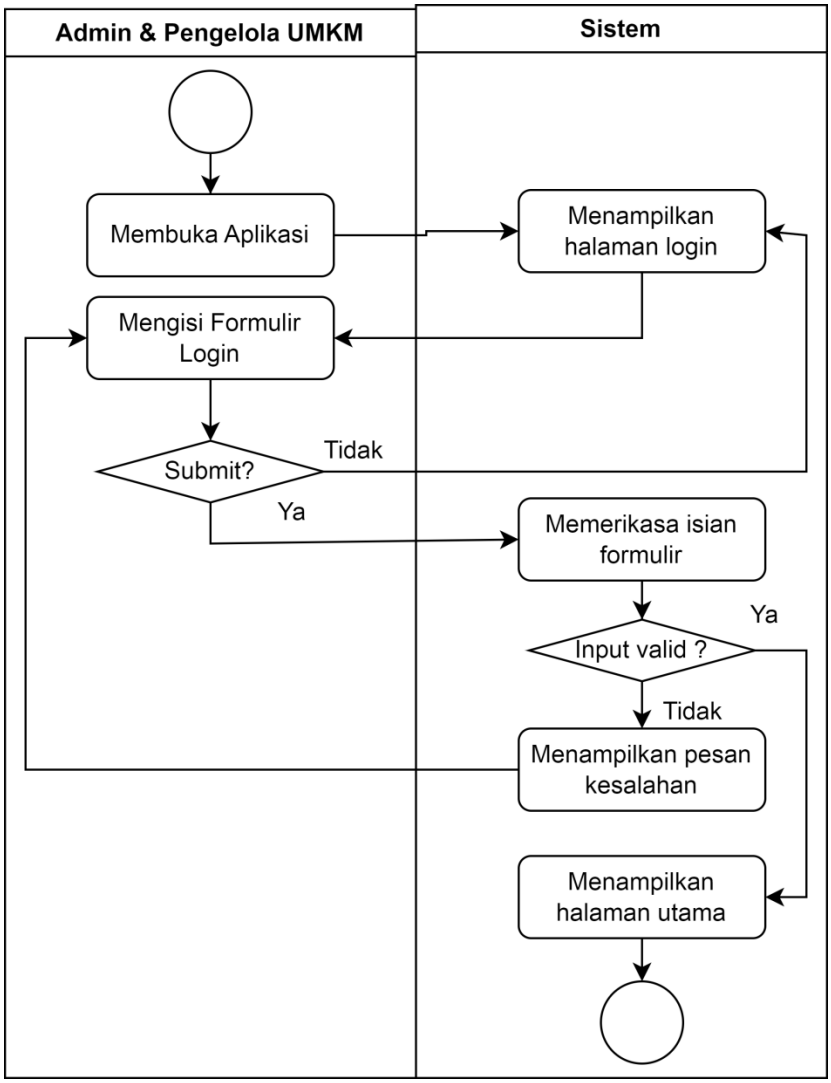
Tabel 4 31. Skenario Use Case Kelola Pengguna UMKM

Aksi Aktor	Reaksi Sistem
1.Menekan menu Kelola Pengguna UMKM	
	2. Menampilkan menu Kelola Pengguna UMKM

2. Activity Diagram

Activity diagram adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membanngun perangkat lunak *Activity diagram* ini menjelaskan tentang aktifitas-aktifitas yang terjadi dalam sebuah aliran proses pada sebuah system.

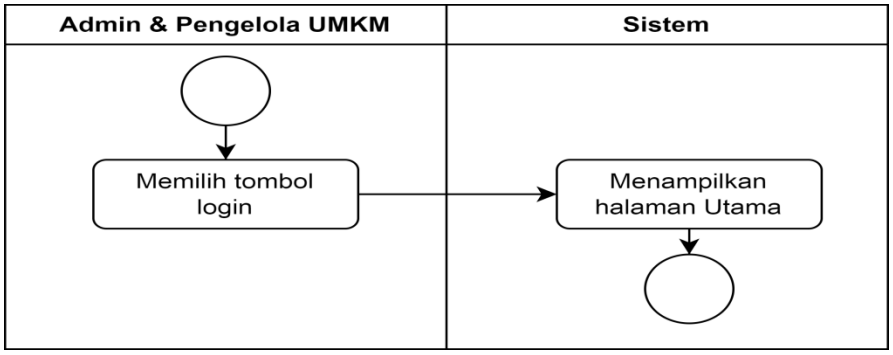
a. ***Activity Diagram Login***



Gambar 4 2. *Activity Diagram Login*

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas login. Proses dimulai dengan admin dan pengelola UMKM memilih tombol login, kemudian sistem menampilkan form isian yang kemudian admin atau user akan mengisi form isian. Ketika sudah disimpan, sistem akan memperbarui data di database yang akan mengecek data yang dikirim, jika terdapat kesalahan, akan menampilkan pesan kesalahan, jika tidak, sistem menampilkan halaman utama.

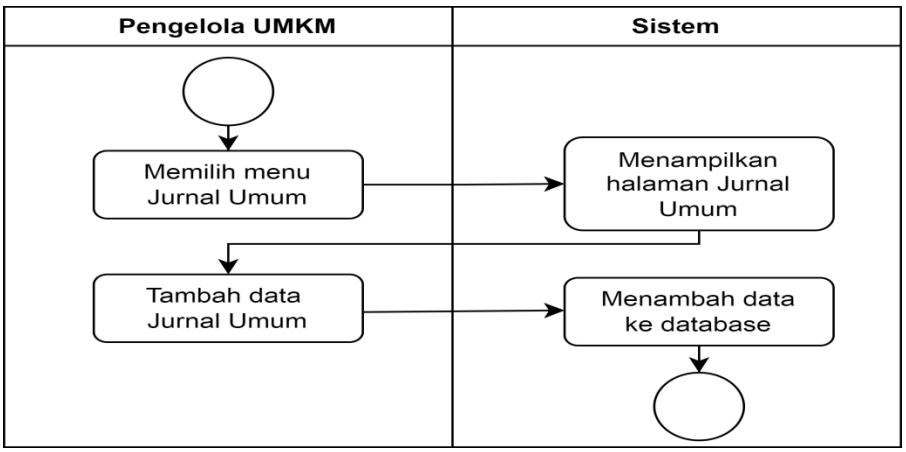
b. ***Activity Diagram Home***



Gambar 4 3. Activity Diagram Home

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas Home. Proses dimulai dengan admin dan pengelola UMKM memilih tombol login, kemudian system berhasil menampilkan halaman utama aplikasi.

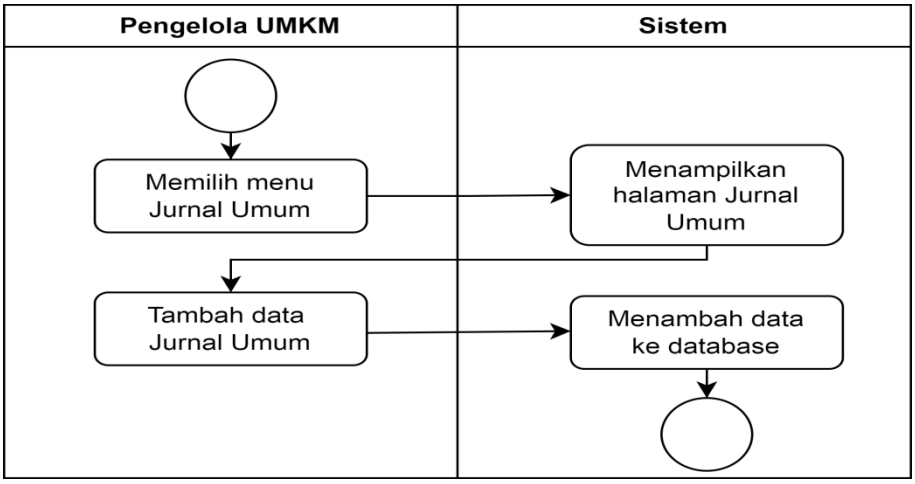
c. Activity Diagram Jurnal Umum



Gambar 4 4. *Activity Diagram* Jurnal Umum

Diagram aktivitas berikut menerangkan alur proses untuk aktivitas Jurnal Umum Proses dimulai dengan Pengguna UMKM memilih tombol Jurnal Umum, kemudian sistem menampilkan halaman Jurnal Umum. Jika ingin tambah data, maka dapat memilih tambah data sistem akan menambahkan data tersebut ke database.

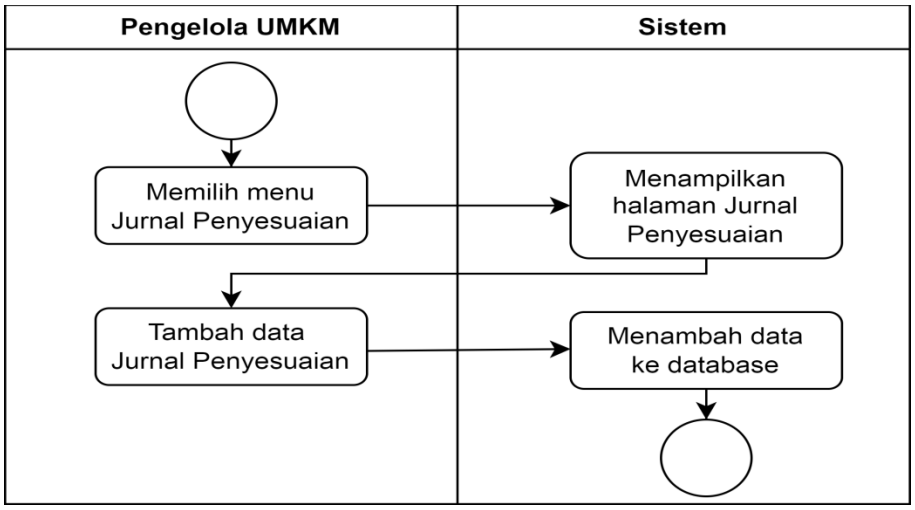
d. Activity Diagram Jurnal Penyesuaian



Gambar 4 5. *Activity Diagram* Jurnal Penyesuaian

Diagram aktivitas berikut menerangkan alur proses untuk aktivitas Jurnal Penyesuaian Proses dimulai dengan Pengguna UMKM memilih tombol Jurnal Penyesuaian, kemudian sistem menampilkan halaman Jurnal Penyesuaian. Jika ingin tambah data, maka dapat memilih tambah data sistem akan menambahkan data tersebut ke database.

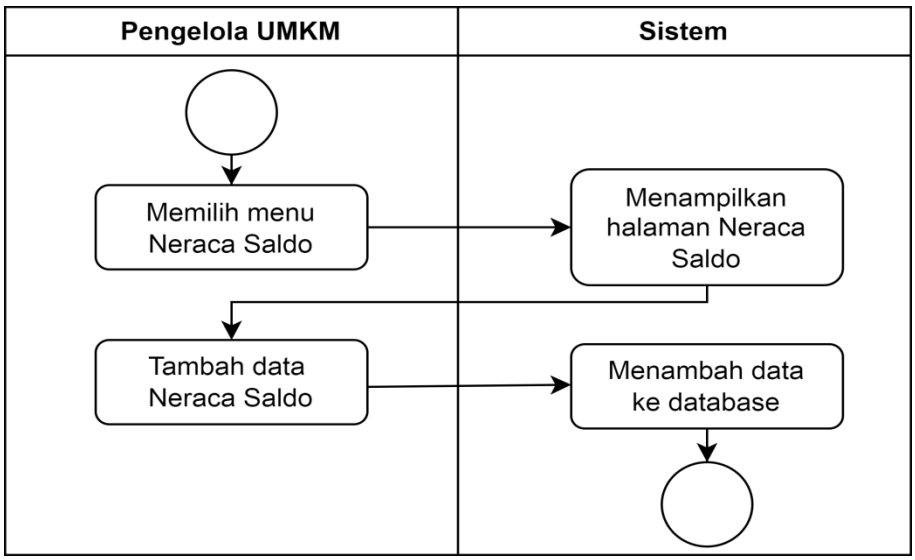
d. Activity Diagram Jurnal Penyesuaian



Gambar 4 5. *Activity Diagram* Jurnal Penyesuaian

Diagram aktivitas berikut menerangkan alur proses untuk aktivitas Jurnal Penyesuaian Proses dimulai dengan Pengguna UMKM memilih tombol Jurnal Penyesuaian, kemudian sistem menampilkan halaman Jurnal Penyesuaian. Jika ingin tambah data, maka dapat memilih tambah data sistem akan menambahkan data tersebut ke database.

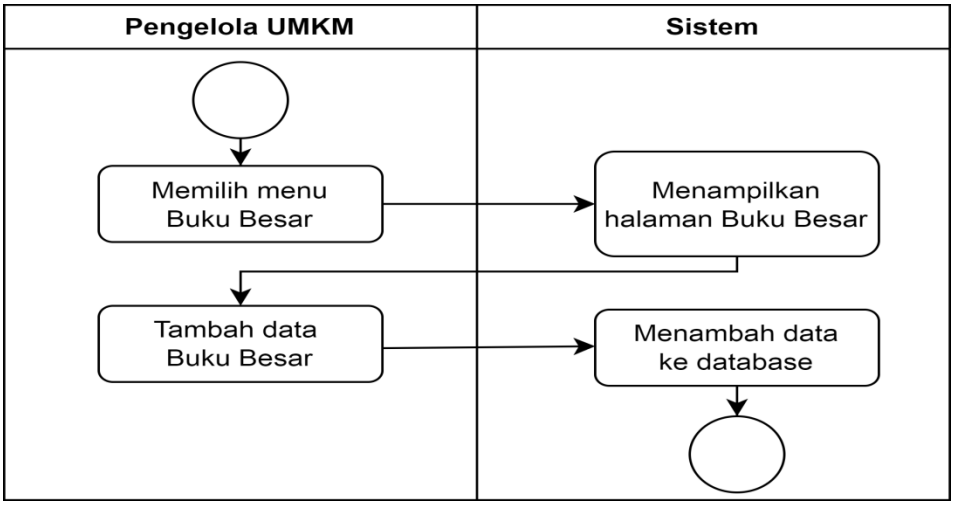
e. Activity Diagram Neraca Saldo



Gambar 4 6. *Activity Diagram* Neraca Saldo

Diagram aktivitas berikut menerangkan alur proses untuk aktivitas Neraca Saldo Proses dimulai dengan Pengguna UMKM memilih tombol Neraca Saldo, kemudian sistem menampilkan halaman Neraca Saldo. Jika ingin tambah data, maka dapat memilih tambah data sistem akan menambahkan data tersebut ke database.

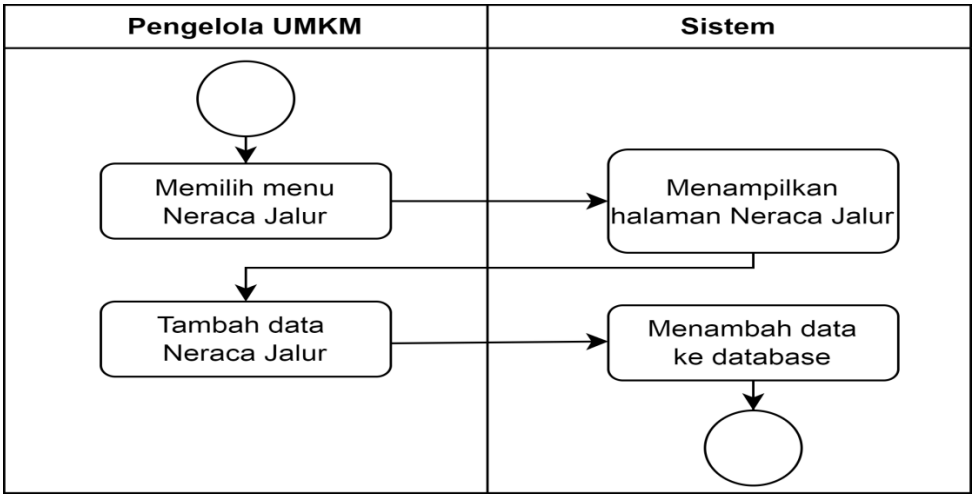
f. Activity Diagram Buku Besar



Gambar 4 7. *Activity Diagram* Buku Besar

Diagram aktivitas berikut menerangkan alur proses untuk aktivitas Buku Besar Proses dimulai dengan Pengguna UMKM memilih tombol Buku Besar, kemudian sistem menampilkan halaman Buku Besar. Jika ingin tambah data, maka dapat memilih tambah data sistem akan menambahkan data tersebut ke database.

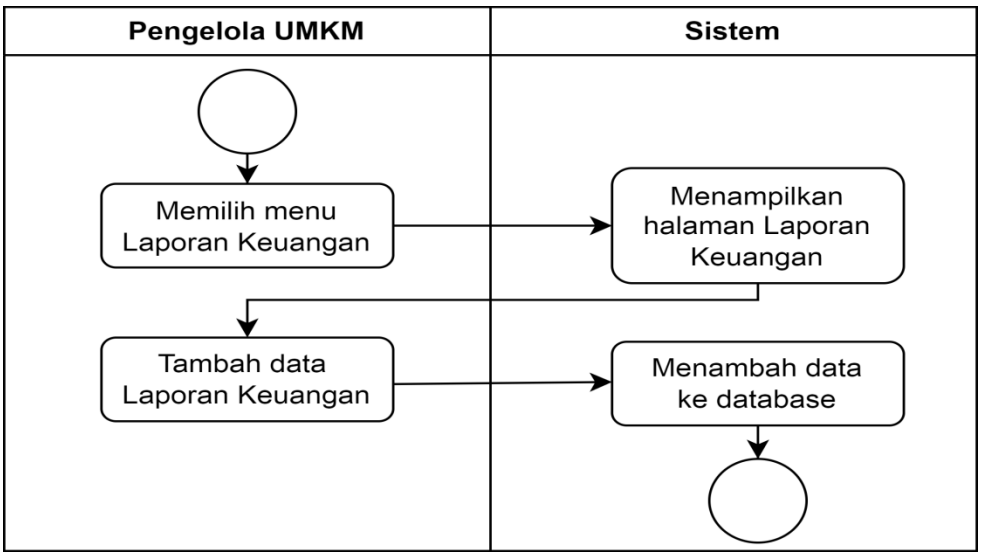
g. Activity Diagram Neraca Jalur



Gambar 4 8. *Activity Diagram* Neraca Jalur

Diagram aktivitas berikut menerangkan alur proses untuk aktivitas Neraca Jalur Proses dimulai dengan Pengguna UMKM memilih tombol Neraca Jalur, kemudian sistem menampilkan halaman Neraca Jalur. Jika ingin tambah data, maka dapat memilih tambah data sistem akan menambahkan data tersebut ke database.

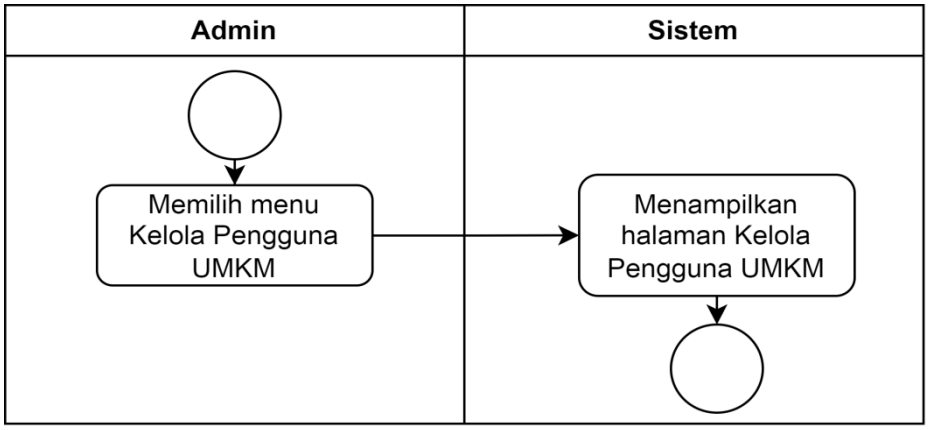
h. Activity Diagram Laporan Keuangan



Gambar 4 9. *Activity Diagram* Laporan Keuangan

Diagram aktivitas berikut menerangkan alur proses untuk aktivitas Laporan Keuangan Proses dimulai dengan Pengguna UMKM memilih tombol Laporan Keuangan, kemudian sistem menampilkan halaman Laporan Keuangan. Jika ingin tambah data, maka dapat memilih tambah data sistem akan menambahkan data tersebut ke database.

i. Activity Diagram Kelola Pengguna UMKM



Gambar 4 10. *Activity Diagram* Kelola Pengguna UMKM

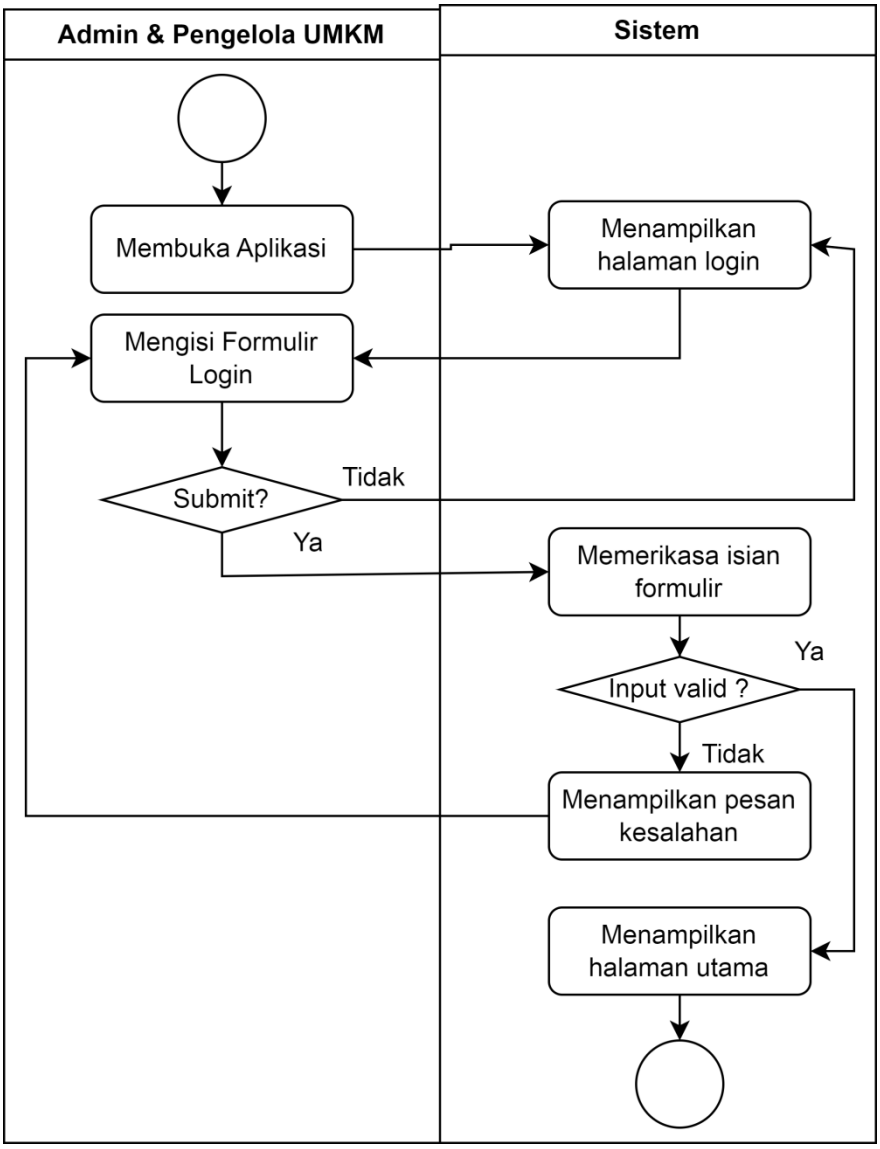
Diagram aktivitas berikut menerangkan alur proses untuk aktivitas Kelola Pengguna UMKMProses dimulai dengan admin memilih tombol Kelola Pengguna UMKM, kemudian sistem menampilkan halaman Kelola Pengguna UMKM.

3. Sequence Diagram

Sequence diagram atau diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripisikan waktu hidup objek dan message yang dikirim dan diterima antar objek. Proses menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta method yang dimiliki kelas yang diinisialisasi menjadi objek yang sudah tergambar dalam class diagram.

Berdasarkan desain use case, terdapat beberapa use case yang prosesnya hampir sama satu sama lain. Untuk mempermudah pembaha-san proses dalam pembuatan diagram sekuen, berikut ini ringkasan diagram sekuen pada aplikasi Sistem Laporan Keuangan Umkm.

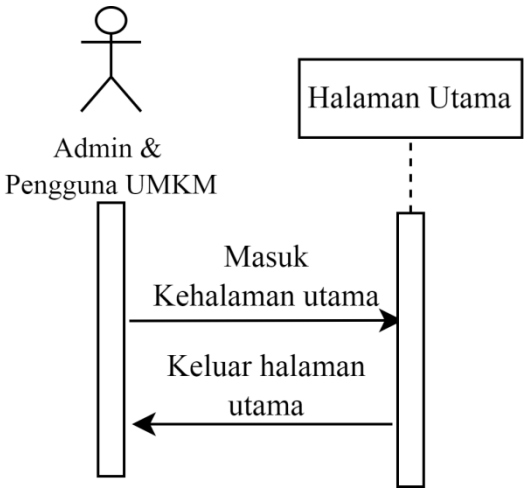
a. *Sequence Diagram*: *Login*



Gambar 4 5. *Sequence* Diagram login

Diagram Sequence diagram login admin dan pengelola UMKM merupakan penggambaran proses login yang dilakukan admin dan pengelola UMKM. Bila proses login benar admin dan pengelola UMKM bisa masuk ke dalam halaman utama sebaliknya bila proses login gagal akan kembali ke menu *login*.

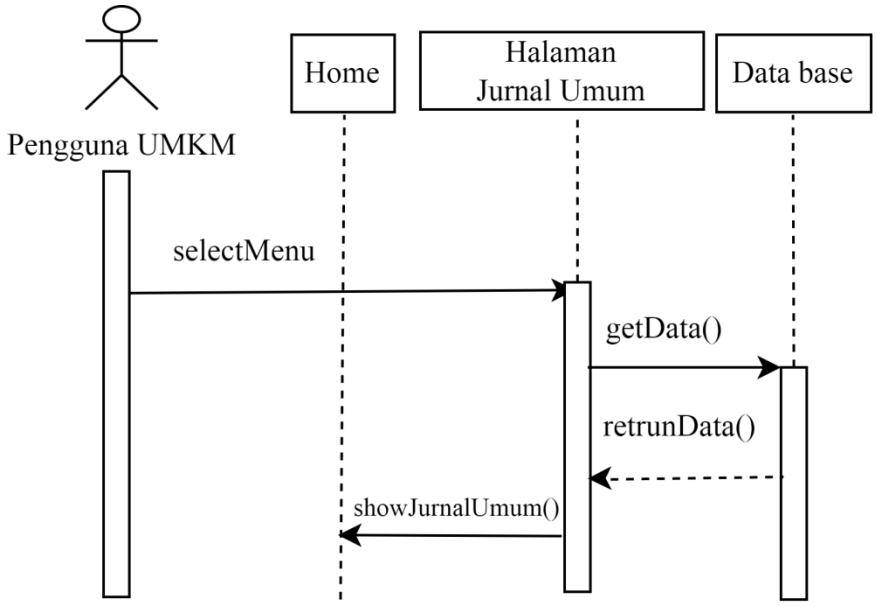
b. *Sequence Diagram*: Home



Gambar 4 6. *Sequence* Diagram Home

Diagram Sequence diagram home admin dan pengelola UMKM merupakan penggambaran proses masuk ke halaman utama. Bila proses login benar admin dan pengelola UMKM bisa masuk ke dalam halaman utama dan menampilkan menu-menu yang ada pada aplikasi.

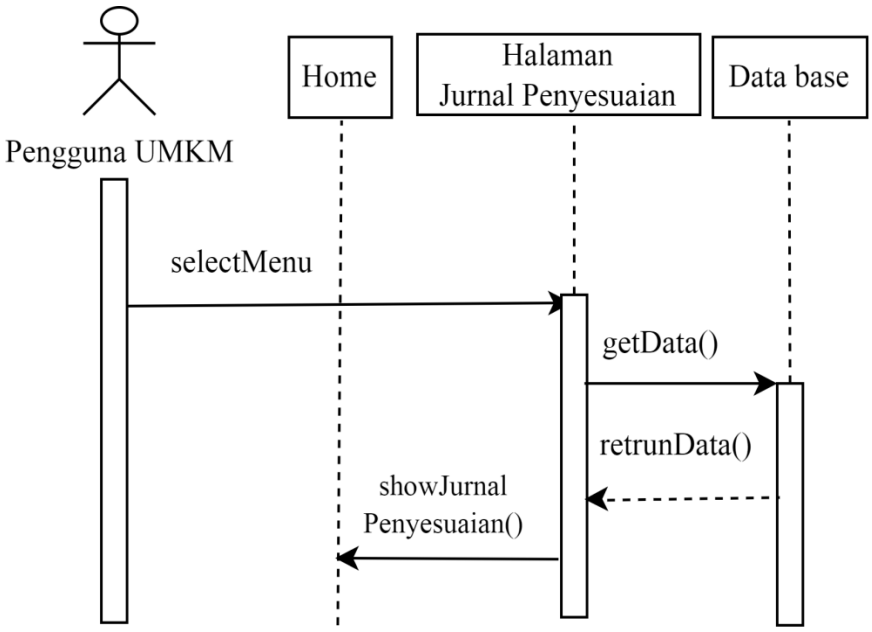
c. *Sequence Diagram* Jurnal Umum



Gambar 4 7. *Sequence* Diagram Jurnal Umum

Diagram sekuen di atas menerangkan alur proses untuk aktivitas Jurnal Umum. pengelola UMKM memilih data yang akan ditampilkan pada halaman Jurnal Umum. kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method showJurnalUmum() akan menampilkan Jurnal Umum.

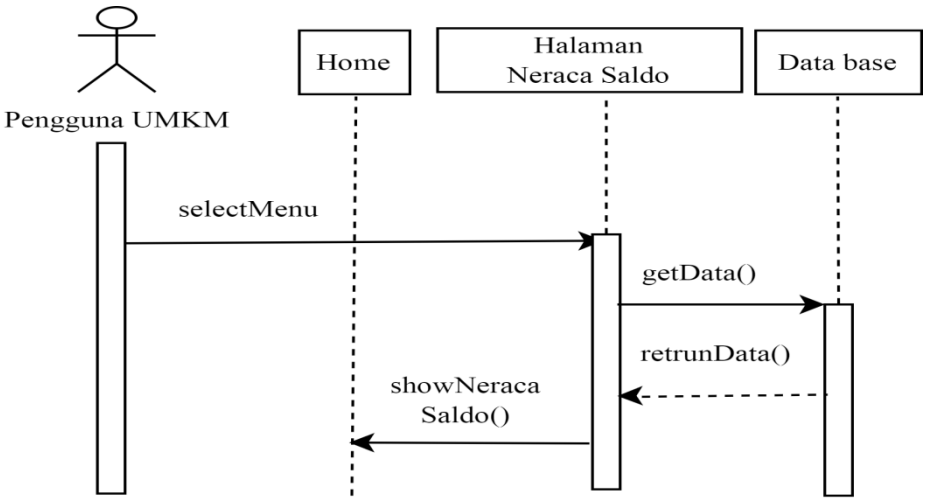
d. *Sequence Diagram* Jurnal Penyesuaian



Gambar 4 8. *Sequence* Diagram Jurnal Penyesuaian

Diagram sekuen di atas menerangkan alur proses untuk aktivitas Jurnal Penyesuaian. pengelola UMKM memilih data yang akan ditampilkan pada halaman Jurnal Penyesuaian. kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method showJurnalPenyesuaian() akan menampilkan Jurnal Penyesuaian.

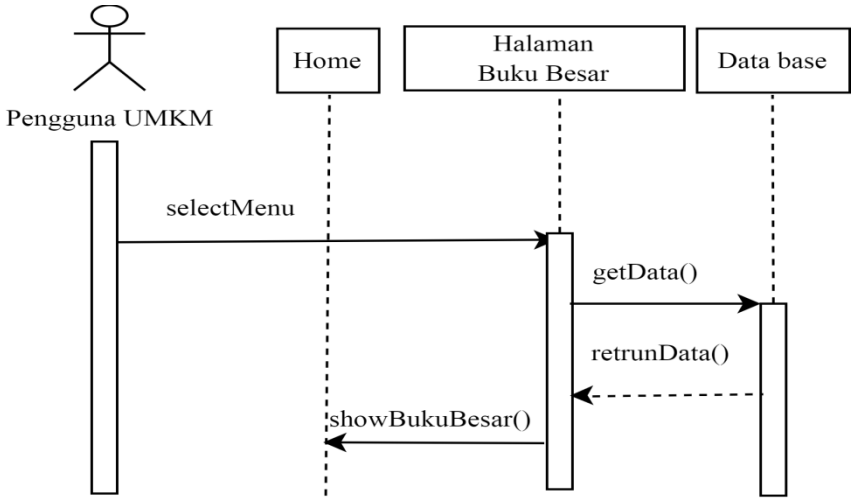
e. *Sequence Diagram* Neraca Saldo



Gambar 4 9. *Sequence* Diagram Neraca Saldo

Diagram sekuen di atas menerangkan alur proses untuk aktivitas Neraca Saldo. pengelola UMKM memilih data yang akan ditampilkan pada halaman Neraca Saldo. kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method showNeracaSaldo() akan menampilkan Neraca Saldo.

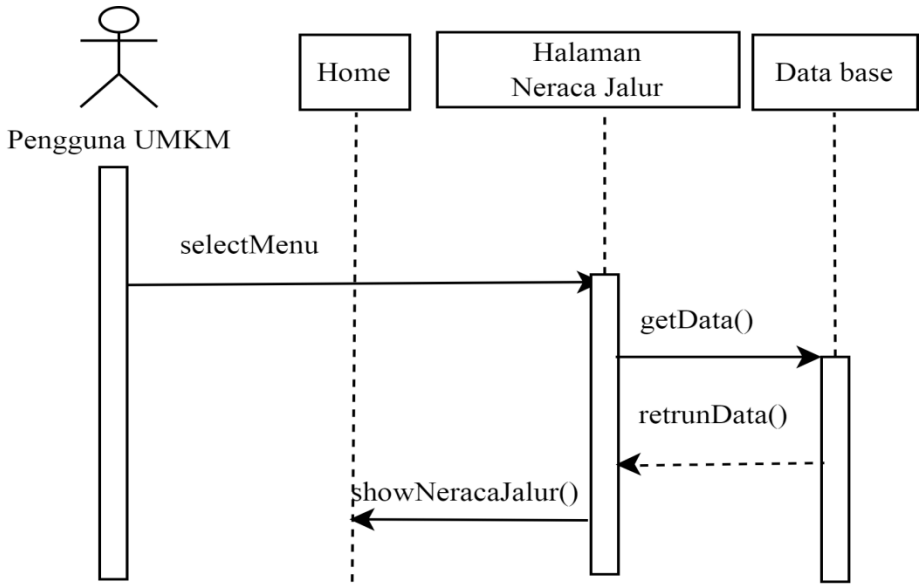
f. *Sequence Diagram* Buku Besar



Gambar 4 10. *Sequence* Diagram Buku Besar

Diagram sekuen di atas menerangkan alur proses untuk aktivitas Buku Besar. pengelola UMKM memilih data yang akan ditampilkan pada halaman Buku Besar. kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method showBukuBesar() akan menampilkan Buku Besar.

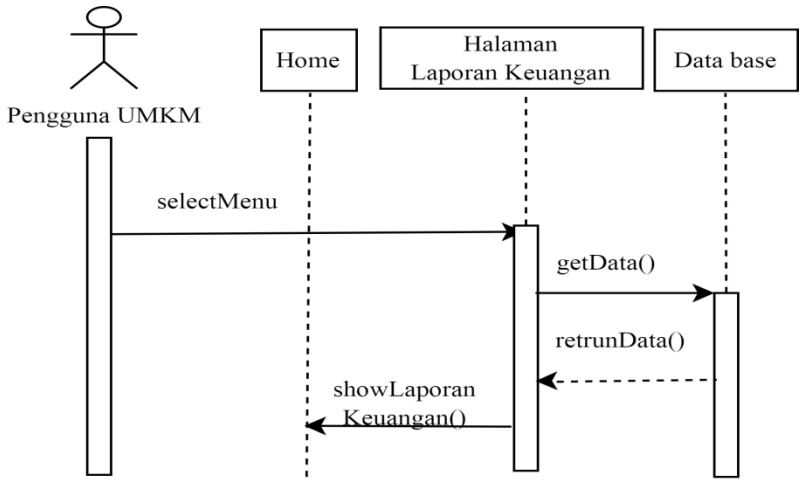
g. *Sequence Diagram* Neraca Jalur



Gambar 4 11. *Sequence* Diagram Neraca Jalur

Diagram sekuen di atas menerangkan alur proses untuk aktivitas Neraca Jalur. pengelola UMKM memilih data yang akan ditampilkan pada halaman Neraca Jalur. kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method showNeracaJalur() akan menampilkan Neraca Jalur.

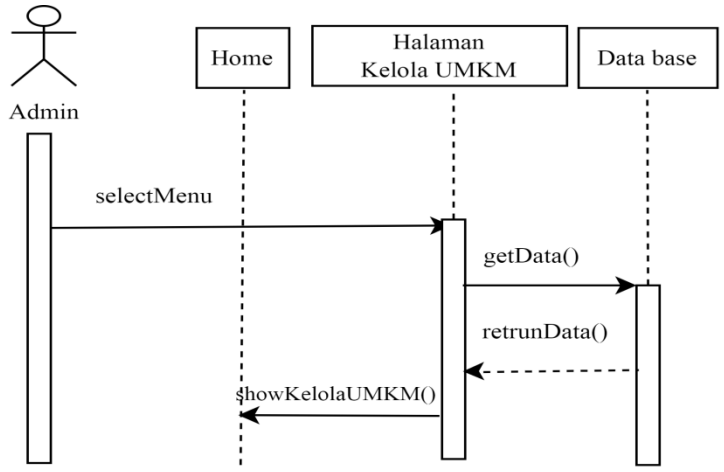
h. *Sequence Diagram* Laporan Keuangan



Gambar 4 12. *Sequence* Diagram Laporan Keuangan

Diagram sekuen di atas menerangkan alur proses untuk aktivitas Laporan Keuangan. Pengelola UMKM memilih data yang akan ditampilkan pada halaman Laporan Keuangan. kemudian method getData() digunakan untuk mengambil data yang akan ditampilkan grafiknya ke database. Database akan mengembalikan data, kemudian method showLaporanKeuangan() akan menampilkan Laporan Keuangan.

i. *Sequence Diagram* Kelola Pengguna UMKM



Gambar 4 13. *Sequence* Diagram Kelola Pengguna UMKM

Diagram sekuen di atas menerangkan alur proses untuk aktivitas Kelola UMKM, Pengelola UMKM memilih data yang akan ditampilkan pada halaman Kelola UMKM.

4. Kamus Data

Kamus data merupakan kumpulan data yang digunakan serta dihasilkan oleh perangkat lunak. Berikut ini daftar seluruh data yang akan digunakan dan dihasilkan oleh perangkat lunak sistem laporan keuangan UMKM.

Tabel 4.14. users
Primary key
Foreign key

No	Nama Field	Tipe Data	Ukuran
1	id	bigint	20
2	umkm_id	bigint	20
3	phone	Varchar	16
4	name	Varchar	255
5	nomor_hp	Varchar	255
6	email	Varchar	30
7	email_verified_ad	timestamp	-
8	password	Varchar	255
9	remember_token	Varchar	100
10	created_at	timestamp	-
11	updated_at	timestamp	-

Tabel 4. 15 umkms
Primary key
Foreign key

No	Nama Field	Tipe Data	Ukuran
1	id	bigint	20
2	name	Varchar	255
3	status	enum(‘active,’inactive’	-
4	address	Varchar	255
5	created_at	timestamp	-
6	updated_at	timestamp	-

Tabel 4.16. jurnal_penyesuaian
Primary key
Foreign key

No	Nama Field	Tipe Data	Ukuran
1	id	bigint	20
2	umkm_id	bigint	20
3	jurnal_id	bigint	20
4	tanggal	date	-
5	ref	Varchar	255
6	debit	decimal	18,2
7	kredit	decimal	18,2
8	deleted_at	timestamp	-
9	created_at	timestamp	-
10	updated_at	timestamp	-

Tabel 4.17. jurnal_umum
Primary key
Foreign key

No	Nama Field	Tipe Data	Ukuran
1	id	bigint	20
2	umkm_id	bigint	20
3	jurnal_id	bigint	20
4	kode_akun	bigint	20
4	tanggal	date	-
5	ref	Varchar	255
6	debit	decimal	18,2
7	kredit	decimal	18,2
8	deleted_at	timestamp	-
9	created_at	timestamp	-
10	updated_at	timestamp	-

Tabel 4.17. akun
Primary key
Foreign key

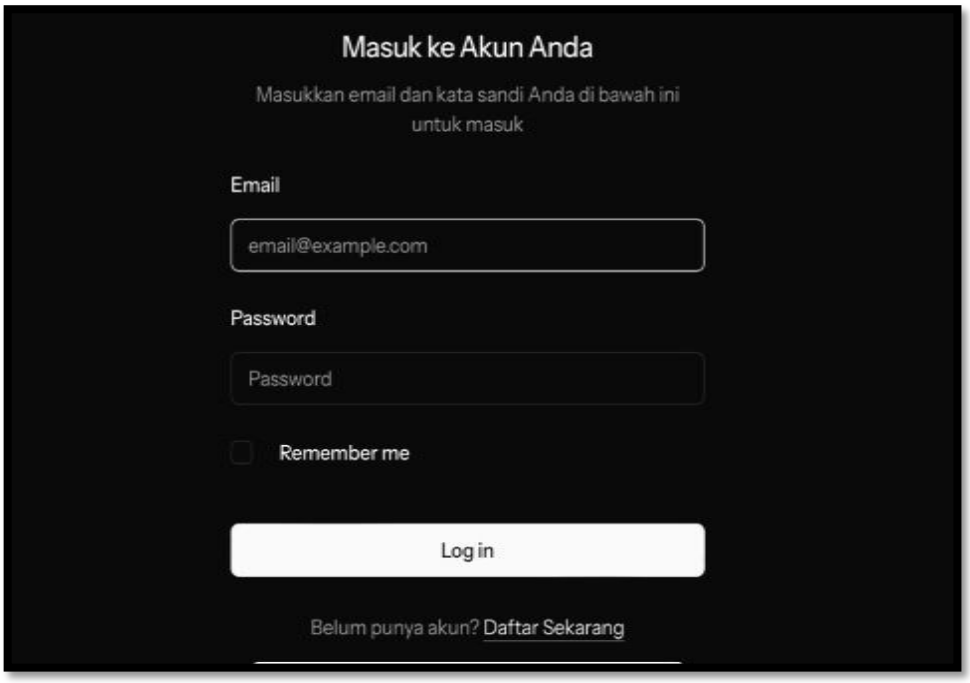
No	Nama Field	Tipe Data	Ukuran
1	kode_akun	bigint	20
2	nama_akun	Varchar	255
3	tipe	enum	-
4	created_at	timestamp	-
5	updated_at	timestamp	-

5. Perancangan Sistem

a. Halaman Admin

1) Halaman Login

Berikut adalah tampilan login pada Aplikasi sistem laporan keuangan UMKM. Halaman menu login merupakan tampilan yang ada pada sistem, yang digunakan untuk masuk ke halaman selanjutnya dengan memasukkan *username* dan *password*.



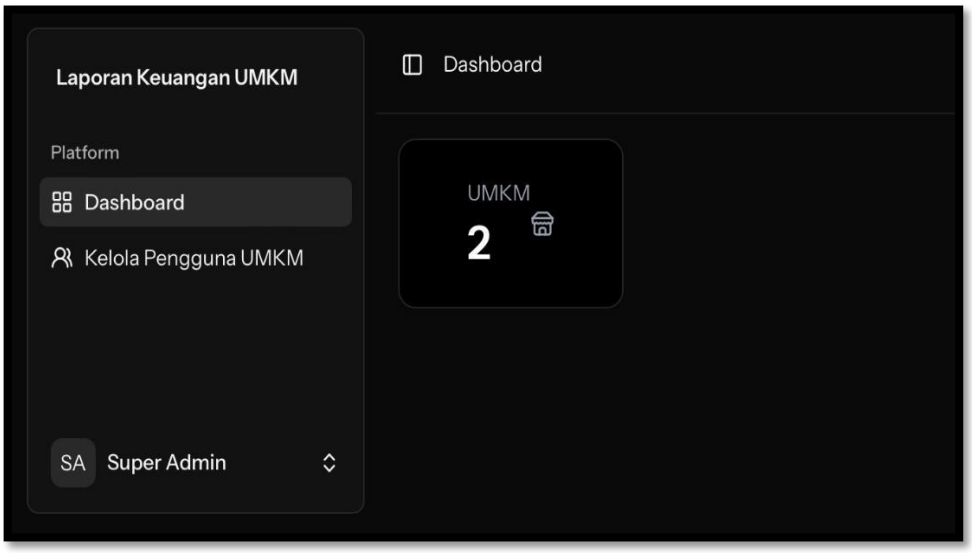
Gambar 4 19. Halaman Login

```
<script setup lang="ts">
import InputError from '@/components/InputError.vue';
import TextLink from '@/components/TextLink.vue';
import { Button } from '@/components/ui/button';
import { Checkbox } from '@/components/ui/checkbox';
import { Input } from '@/components/ui/input';
import { Label } from '@/components/ui/label';
import AuthBase from '@/layouts/AuthLayout.vue';
import { Head, useForm } from '@inertiajs/vue3';
import { LoaderCircle } from 'lucide-vue-next';
defineProps<{
    status?: string;
    canResetPassword: boolean;
}>();
const form = useForm({
    email: '',
    password: '',
    remember: false,
});
const submit = () => {
    form.post(route('login'), {
        onFinish: () => {
            form.password = '';
            console.log('oke');
        },
    });
};
</script>
<template>
    <AuthBase title="Masuk ke Akun Anda" description="Masukkan
email dan kata sandi Anda di bawah ini untuk masuk">
        <Head title="Log in" />
        <div v-if="status" class="mb-4 text-center text-sm font-
medium text-green-600">
            {{ status }}
        </div>
        <form @submit.prevent="submit" class="flex flex-col gap-
6">
            <div class="grid gap-6">
                <div class="grid gap-2">
                    <Label for="email">Email</Label>
                    <Input
                        id="email"
                        type="email"
                        required
                        autofocus
                        :tabindex="1"
                        autocomplete="off"
                        v-model="form.email"
                        placeholder="email@example.com"
                    <InputError :message="form.errors.email" />
                </div>
                <div class="grid gap-2">
                    <div class="flex items-center justify-
between">
                        <Label for="password">Password</Label>
```

```
                        <!-- <TextLink v-if="canResetPassword"
:href="route('password.request')" class="text-sm" :tabindex="5">
                            Forgot password?
                        </TextLink> -->
                    </div>
                    <Input
                        id="password"
                        type="password"
                        required
                        :tabindex="2"
                        autocomplete="current-password"
                        v-model="form.password"
                        placeholder="Password"
                    />
                    <InputError :message="form.errors.password"
/>
                </div>
                <div class="flex items-center justify-between"
:tabindex="3">
                    <Label for="remember" class="flex items-
center space-x-3">
                        <Checkbox id="remember" v-
model="form.remember" :tabindex="4" />
                        <span>Remember me</span>
                    </Label>
                </div>
                <Button type="submit" class="mt-4 w-full"
:tabindex="4" :disabled="form.processing">
                    <LoaderCircle v-if="form.processing"
class="h-4 w-4 animate-spin" />
                    Log in
                </Button>
                <div class="text-muted-foreground text-center
text-sm">
                    Belum punya akun?
                    <TextLink :href="route('register')"
class="underline underline-offset-4" :tabindex="6">Daftar
Sekarang</TextLink>
                </div>
            </div>
        </form>
    </AuthBase>
</template>
```

2) Halaman Dashboard

Berikut adalah tampilan dashboard pada Aplikasi sistem laporan keuangan UMKM. Halaman dashboard dalam aplikasi ini berfungsi sebagai pusat informasi utama dan titik navigasi bagi admin.

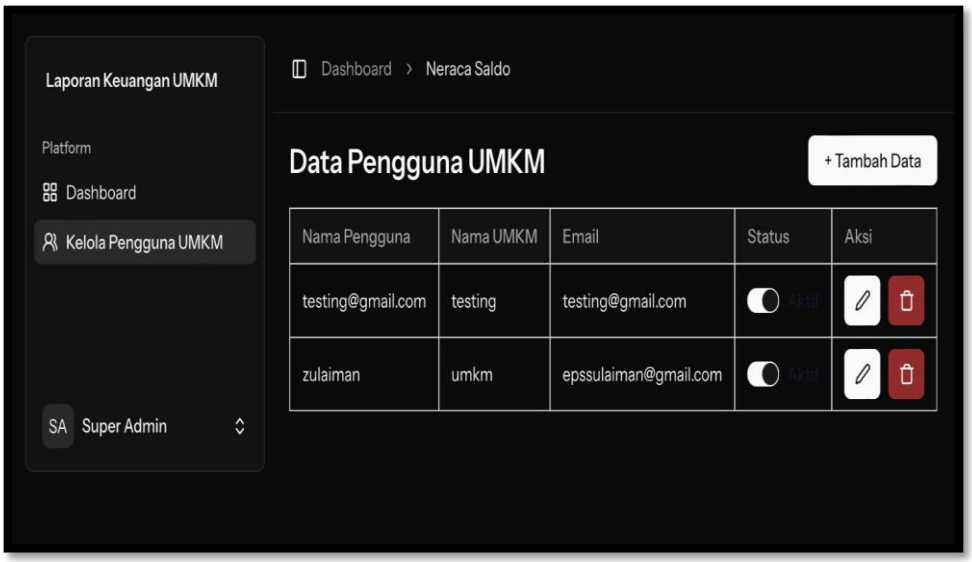


Gambar 4 20. Halaman Dashboard

```
<script setup lang="ts">
import { Card, CardContent } from '@/components/ui/card';
import AppLayout from '@/layouts/AppLayout.vue';
import { SharedData, type BreadcrumbItem } from '@/types';
import { Head, usePage } from '@inertiajs/vue3';
import { StoreIcon } from 'lucide-vue-next';
const { data } = defineProps<{ data: { umkmCount: number } }>();
const { auth } = usePage<SharedData>().props;
const breadcrumbs: BreadcrumbItem[] = [
    {
        title: 'Dashboard',
        href: '/dashboard',
    },
];
            <div v-if="auth.roles?.includes('super-admin')"
class="grid auto-rows-min gap-4 md:grid-cols-3">
                <Card class="rounded-xl bg-gray-100 p-6 shadow-
md dark:bg-black dark:text-white">
                    <CardContent class="flex items-center
justify-
                        </div>
                        <StoreIcon class="h-6 w-6 text-gray-400"
/>
                    </CardContent>
                </Card>
            </div>
```

3) Halaman Kelola Pengguna UMKM

Berikut adalah tampilan Kelola Pengguna UMKM pada Aplikasi sistem laporan keuangan UMKM. Halaman Kelola Pengguna UMKM dalam aplikasi ini berfungsi sebagai pusat informasi utama pada menu Kelola Pengguna UMKM.



Gambar 4 21. Halaman Kelola Pengguna UMKM

```
<template>
    <AppLayout :breadcrumbs="breadcrumbs">
        <Head title="Data Pengguna UMKM" />
        <div class="w-full p-4">
            <div class="mb-4 flex items-center justify-between">
                <h1 class="text-2xl font-bold">Data Pengguna
UMKM</h1>
                <Link :href="route('users.create')">
                    <Button>+ Tambah Data</Button>
                </Link>
            </div>
            <Table v-if="data.length > 0">
                <TableHeader>
                    <TableRow>
                        <TableHead>Nama Pengguna</TableHead>
                        <TableHead>Nama
UMKM</TableHead><TableHead>Aksi</TableHead>
                    </TableRow>
                </TableHeader>
                <TableBody>
```

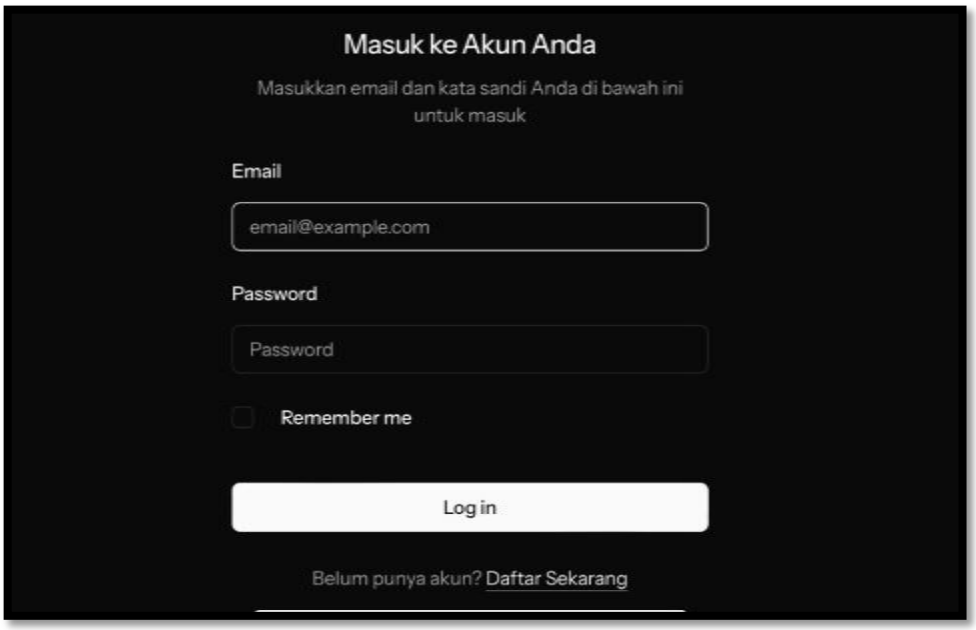
```
<TableHead>Aksi</TableHead>
                    </TableRow>
                </TableHeader>
                <TableBody>
                    <TableRow v-for="item in data"
:key="item.id">
                        <TableCell>{{ item.name || '-'
}}</TableCell>
                        <TableCell>{{ item.umkm.name
}}</TableCell>
                        <TableCell>{{ item.email }}</TableCell>
                        <TableCell>
                            <div class="flex items-center space-
x-2">
                                <Label>
                                    <Switch
                                        @update:model-
value="(status) => handleChange(status, item.id)"
                                        :default-
value="item.umkm.status == 'active'"
                                        id="airplane-mode"
                                        :disabled="isDisabled"
                                    />
                                    <span :id="String(item.id)"
class="text-sm font-medium text-gray-900">
                                        {{ item.umkm.status ==
'active' ? 'Aktif' : 'Non-Aktif' }}
                                    </span>
                                </Label>
                            </div>
                        </TableCell>
                        <TableCell>
                            <div class="flex items-center gap-
2">
                                <Link :href="route('users.edit',
item.id)">
                                    <Button size="icon"><Pen
/></Button>
                                </Link>
                                <Button @click="() =>
deleteItem(item.id)" variant="destructive" size="icon"><Trash
/></Button>
                            </div>
                        </TableCell>
                    </TableRow>
                </TableBody>
            </Table>
            <div v-else class="mt-20 text-center">
                <p class="text-gray-500">Tidak ada data
pengguna.</p>
            </div>
        </div>
    </AppLayout>
</template>
<script setup lang="ts">
import { Button } from '@/components/ui/button';
```

```
import { Label } from '@/components/ui/label';
import { Switch } from '@/components/ui/switch';
import { Table, TableBody, TableCell, TableHead, TableHeader,
TableRow } from '@/components/ui/table';
import AppLayout from '@/layouts/AppLayout.vue';
import { BreadcrumbItem } from '@/types';
import { Head, Link, router } from '@inertiajs/vue3';
import { Pen, Trash } from 'lucide-vue-next';
import { ref } from 'vue';
import { User } from './types';
const { data } = defineProps<{ data: User[] }>();
const breadcrumbs: BreadcrumbItem[] = [
    {
        title: 'Dashboard',
        href: '/dashboard',
    },
    {
        title: 'Kelola Pengguna UMKM',
        href: route('users.index'),
    },
];
const isDisabled = ref(false);
const handleChange = (status: boolean, id: number) => {
    const user = data.find((user) => user.id === id) as User;
    if (user) {
        isDisabled.value = true;
        const label = document.getElementById(String(id)) as
HTMLAnchorElement;
        label.innerHTML = '<span class="animate-
pulse">Loading...</span>';
        router.put(
            route('users.update.status', user.umkm.id),
            {
                status: !status ? 'inactive' : 'active',
            },
            {
                onFinish: () => {
                    isDisabled.value = false;
                    label.innerHTML = status ? 'Aktif' : 'Non-
Aktif';
                },
                preserveScroll: true,
                preserveState: true,
            },
        );
    }
};
const deleteItem = (id: number) => {
    if (confirm('Apakah Anda yakin ingin menghapus pengguna
ini?')) {
        router.delete(route('users.destroy', id), {
            onSuccess: () => {
```

b. Halaman Pengguna UMKMM

1) Halaman Login

Berikut adalah tampilan login pada Aplikasi sistem laporan keuangan UMKM. Halaman menu login merupakan tampilan yang ada pada sistem, yang digunakan untuk masuk ke halaman selanjutnya dengan memasukkan *username* dan *password*.



Gambar 4 22. Halaman Login

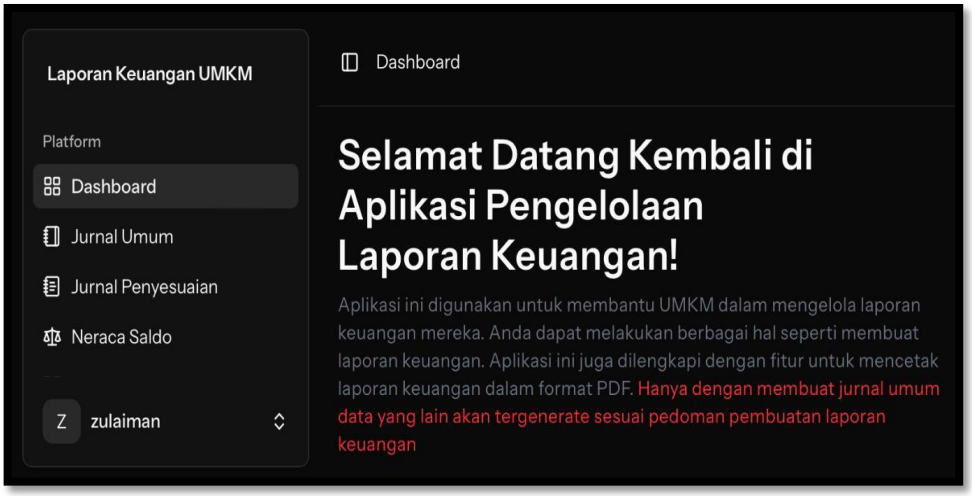
```
<script setup lang="ts">
import InputError from '@/components/InputError.vue';
import TextLink from '@/components/TextLink.vue';
import { Button } from '@/components/ui/button';
import { Checkbox } from '@/components/ui/checkbox';
import { Input } from '@/components/ui/input';
import { Label } from '@/components/ui/label';
import AuthBase from '@/layouts/AuthLayout.vue';
import { Head, useForm } from '@inertiajs/vue3';
import { LoaderCircle } from 'lucide-vue-next';
defineProps<{
    status?: string;
    canResetPassword: boolean;
```

```
const form = useForm({
    email: '',
    password: '',
    remember: false,
});
const submit = () => {
    form.post(route('login'), {
        onFinish: () => {
            form.password = '';
            console.log('oke');
        },
    });
};
</script>
<template>
    <AuthBase title="Masuk ke Akun Anda" description="Masukkan
email dan kata sandi Anda di bawah ini untuk masuk">
        <Head title="Log in" />
        <div v-if="status" class="mb-4 text-center text-sm font-
medium text-green-600">
            {{ status }}
        </div>
        <form @submit.prevent="submit" class="flex flex-col gap-
6">
            <div class="grid gap-6">
                <div class="grid gap-2">
                    <Label for="email">Email</Label>
                    <Input
                        id="email"
                        type="email"
                        required
                        autofocus
                        :tabindex="1"
                        autocomplete="off"
                        v-model="form.email"
                        placeholder="email@example.com"
                    <InputError :message="form.errors.email" />
                </div>
                <div class="grid gap-2">
                    <div class="flex items-center justify-
between">
                        <Label for="password">Password</Label>
                        <!-- <TextLink v-if="canResetPassword"
:href="route('password.request')" class="text-sm" :tabindex="5">
                            Forgot password?
                        </TextLink> -->
                    </div>
                    <Input
                        id="password"
                        type="password"
                        required
                        :tabindex="2"
                        autocomplete="current-password"
                        v-model="form.password"
                        placeholder="Password"
                    />
```

2) Halaman Dashboard

Berikut adalah tampilan dashboard pada Aplikasi sistem laporan keuangan UMKM. Halaman dashboard dalam aplikasi ini berfungsi sebagai pusat informasi utama dan titik navigasi bagi pengguna UMKM.

.



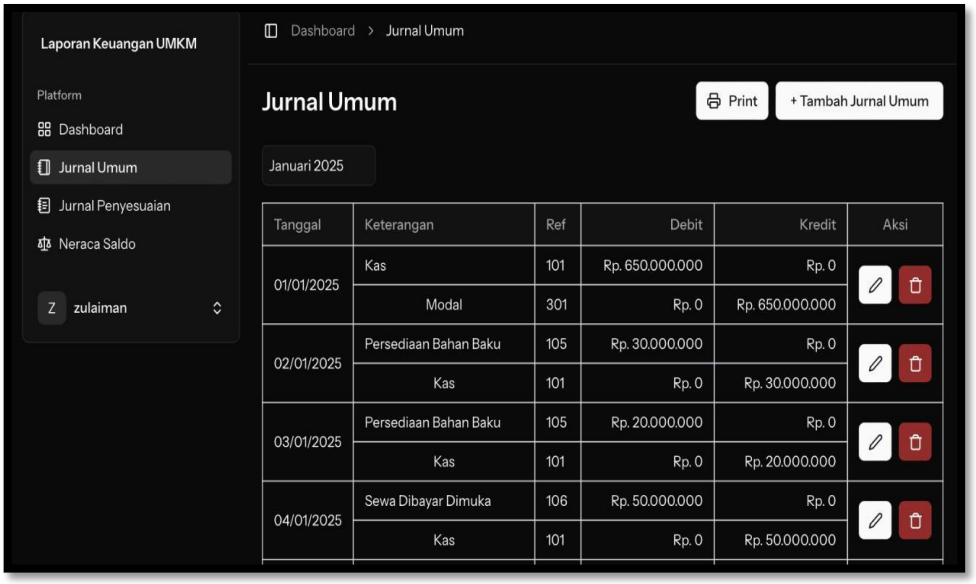
Gambar 4 23. Halaman Dashboard

```
<script setup lang="ts">
import { Card, CardContent } from '@/components/ui/card';
import AppLayout from '@/layouts/AppLayout.vue';
import { SharedData, type BreadcrumbItem } from '@/types';
import { Head, usePage } from '@inertiajs/vue3';
import { StoreIcon } from 'lucide-vue-next';
const { data } = defineProps<{ data: { umkmCount: number } }>();
const { auth } = usePage<SharedData>().props;
const breadcrumbs: BreadcrumbItem[] = [
    {
        title: 'Dashboard',
        href: '/dashboard',
    },
];
</script>
<template>
    <Head title="Dashboard" />
    <AppLayout :breadcrumbs="breadcrumbs">
        <div class="flex h-full flex-1 flex-col gap-4 rounded-xl
p-4">
```

```
                <Card class="rounded-xl bg-gray-100 p-6 shadow-
md dark:bg-black dark:text-white">
                    <CardContent class="flex items-center
justify-between">
                        <div>
                            <p class="mb-1 text-sm text-gray-
400">UMKM</p>
                            <h2 class="text-3xl font-bold">{{
data.umkmCount }}</h2>
                            <!-- <p class="mt-1 text-sm text-
green-400">+180.1% from last month</p> -->
                        </div>
                        <StoreIcon class="h-6 w-6 text-gray-400"
/>
                    </CardContent>
                </Card>
            </div>
            <div v-else>
                <h2 class="text-xl font-bold tracking-wide
md:text-3xl">Selamat Datang Kembali di Aplikasi Pengelolaan <br
/>Laporan Keuangan!</h2>
                <!-- cara penggunanaan -->
                <div class="mt-2 w-full max-w-3xl text-sm
tracking-wide text-gray-500">
                    Aplikasi ini digunakan untuk membantu UMKM
dalam mengelola laporan keuangan mereka. Anda dapat melakukan
berbagai hal seperti
                    membuat laporan keuangan. Aplikasi ini juga
dilengkapi dengan fitur untuk mencetak laporan keuangan dalam
format PDF.
                    <span class="text-red-500"
                        >Hanya dengan membuat jurnal umum data
yang lain akan tergenerate sesuai pedoman pembuatan laporan
keuangan</span
                    >
                </div>
            </div>
        </div>
    </AppLayout>
</template>
```

3) Halaman Jurnal Umum

Berikut Berikut adalah tampilan Jurnal Umum pada Aplikasi sistem laporan keuangan UMKM. Halaman Jurnal Umum dalam aplikasi ini berfungsi untuk menampilkan daftar Jurnal Umum, Menambahkan data serta menyediakan menu print pada halaman Jurnal Umum.



Gambar 4 24. Halaman Jurnal Umum

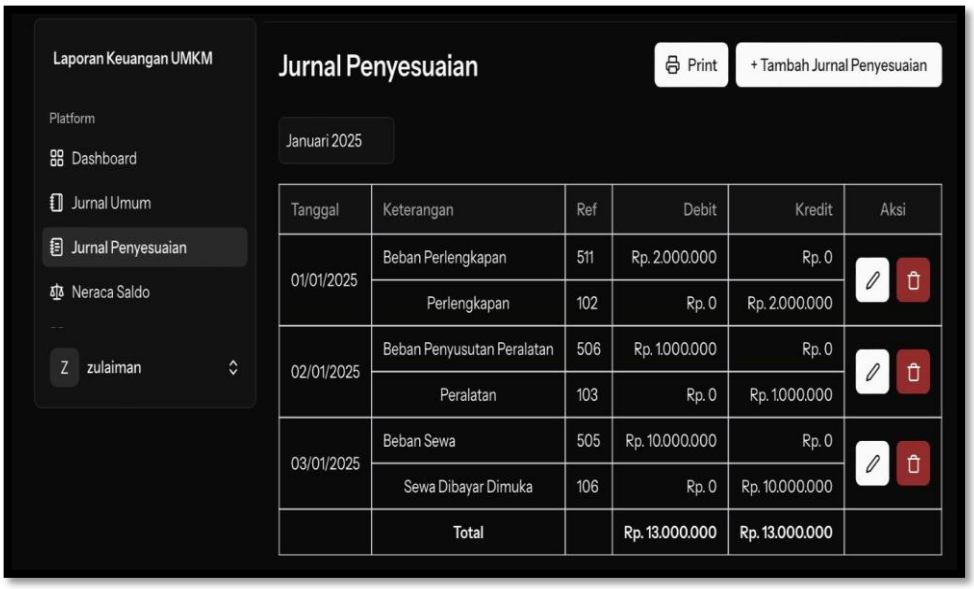
```
File: resources/js/Pages/JurnalUmum/Index.vue
<template>
    <AppLayout :breadcrumbs="breadcrumbs">
        <div class="p-4">
            <div class="mb-4 flex items-center justify-between">
                <h1 class="text-2xl font-bold">
                    Jurnal Umum <span class="hidden
print:inline">- {{ formatMonth(month as string) }}</span>
                </h1>
                <div class="flex items-center gap-2">
                    <PrintButton />
                    <Link
                        :href="
                            route('jurnal.create', {
                                month: month,
                            })
                   class="print:hidden"
```

```
                        <Button>+ Tambah Jurnal Umum</Button>
                    </Link>
                </div>
            </div>
            <div class="mt-4 flex items-center gap-2
print:hidden">
                <Input
                    type="month"
                    placeholder="Cari berdasarkan keterangan,
ref, debit, atau kredit"
                    class="mb-4 w-max"
                    v-model="month"
                    @input="filterMonth"
                />
            </div>
            <Table v-if="items.length > 0" class="w-full">
                <TableHeader>
                    <TableRow>
                        <TableHead>Tanggal</TableHead>
                        <TableHead>Keterangan</TableHead>
                        <TableHead>Ref</TableHead>
                        <TableHead class="text-
right">Debit</TableHead>
                        <TableHead class="text-
right">Kredit</TableHead>
                        <TableHead class="text-center
print:hidden">Aksi</TableHead>
                    </TableRow>
                </TableHeader>
                <TableBody>
                    <TableRow v-for="(item, index) in items"
:key="item.id">
                        <TableCell rowspan="2" v-if="index % 2
== 0">{{ item.formatted_date }}</TableCell>
                        <TableCell :class="{ 'text-center':
index % 2 == 1 }">{{ item.keterangan ?? item.akun?.nama_akun ??
'-' }}</TableCell>
                        <TableCell>{{ item.kode_akun || '-'
}}</TableCell>
                        <TableCell class="text-right">{{
formatRupiah(item.debit) }}</TableCell>
                        <TableCell class="text-right">{{
formatRupiah(item.kredit) }}</TableCell>
                        <TableCell rowspan="2" v-if="index % 2
== 0" class="text-center print:hidden">
                            <div class="flex justify-center gap-
2">
                                <Link
:href="route('jurnal.edit', item.jurnal_id ?? item.id) +
`?month=${month}`">
                                    <Button size="icon"><Pen
/></Button>
                                </Link>
                                <Button
@click="hapus(item.jurnal_id ?? item.id)" variant="destructive"
size="icon"><Trash /></Button>
```

```
                            </div>
                        </TableCell>
                    </TableRow>
                    <TableRow>
                        <TableCell></TableCell>
                        <TableCell class="text-center font-
bold">Total</TableCell>
                        <TableCell></TableCell>
                        <TableCell class="text-right font-
bold">{{ formatRupiah(sumJurnalUmum('debit')) }}</TableCell>
                        <TableCell class="text-right font-
bold">{{ formatRupiah(sumJurnalUmum('kredit')) }}</TableCell>
                        <TableCell
class="print:hidden"></TableCell>
                    </TableRow>
                </TableBody>
            </Table>
            <Alert v-else variant="destructive">
                <InfoIcon class="h-4 w-4" />
                <AlertTitle>Tidak ada data Jurnal
Umum.</AlertTitle>
                <AlertDescription> Silakan tambahkan transaksi
jurnal umum baru untuk bulan {{ formatMonth(month as string) }}.
</AlertDescription>
            </Alert>
        </div>
    </AppLayout>
</template>
<script setup lang="ts">
import { Alert, AlertDescription, AlertTitle } from
'@/components/ui/alert';
import { Button } from '@/components/ui/button';
import PrintButton from
'@/components/ui/button/PrintButton.vue';
import { Input } from '@/components/ui/input';
import { Table, TableBody, TableCell, TableHead, TableHeader,
TableRow } from '@/components/ui/table';
import AppLayout from '@/layouts/AppLayout.vue';
import { formatMonth } from '@/lib/utils';
import { BreadcrumbItem } from '@/types';
import { Link, router } from '@inertiajs/vue3';
import { InfoIcon, Pen, Trash } from 'lucide-vue-next';
import { onMounted, ref } from 'vue';
import { JurnalUmum } from './types';
const props = defineProps<{ data: any[]; month: string }>();
const formatRupiah = (val: number | string): string => {
    const n = parseInt(String(val).replace(/[^\d]/g, '')) || 0;
    return 'Rp. ' + n.toLocaleString('id-ID');
};
const hapus = (id: string) => {
    if (confirm('Yakin ingin menghapus data ini?')) {
        router.delete(route('jurnal.destroy', id), {
            data: {
```

4) Halaman Jurnal Penyesuaian

Berikut Berikut adalah tampilan Jurnal Penyesuaian pada Aplikasi sistem laporan keuangan UMKM. Halaman Jurnal Penyesuaian dalam aplikasi ini berfungsi untuk menampilkan daftar Jurnal Penyesuaian, Menambahkan data serta menyediakan menu print pada halaman Jurnal Penyesuaian.



Gambar 4 25 Halaman Jurnal Penyesuaian

```
File: resources/js/Pages/JurnalUmum/Index.vue
<template>
    <AppLayout :breadcrumbs="breadcrumbs">
        <div class="p-4">
            <div class="mb-4 flex items-center justify-between">
                <h1 class="text-2xl font-bold">
                    Jurnal Penyesuaian <span class="hidden
print:inline">- {{ formatMonth(month as string) }}</span>
                </h1>
                <div class="flex items-center gap-2">
                    <PrintButton />
                    <Link
                        :href="
                            route('jurnal-penyesuaian.create', {
```

```
                        <Button>+ Tambah Jurnal
Penyesuaian</Button>
                    </Link>
                </div>
            </div>
            <div class="mt-4 flex items-center gap-2
print:hidden">
                <Input
                    type="month"
                    placeholder="Cari berdasarkan keterangan,
ref, debit, atau kredit"
                    class="mb-4 w-max"
                    v-model="month"
                    @input="filterMonth"
                />
            </div>
            <Table v-if="items.length > 0" class="w-full">
                <TableHeader>
                    <TableRow>
                        <TableHead>Tanggal</TableHead>
                        <TableHead>Keterangan</TableHead>
                        <TableHead>Ref</TableHead>
                        <TableHead class="text-
right">Debit</TableHead>
                        <TableHead class="text-
right">Kredit</TableHead>
                        <TableHead class="text-center
print:hidden">Aksi</TableHead>
                    </TableRow>
                </TableHeader>
                <TableBody>
                    <TableRow v-for="(item, index) in items"
:key="item.id">
                        <TableCell rowspan="2" v-if="index % 2
== 0">{{ item.formatted_date }}</TableCell>
                        <TableCell :class="{ 'text-center':
index % 2 == 1 }">{{ item.keterangan ?? item.akun?.nama_akun ??
'-' }}</TableCell>
                        <TableCell>{{ item.kode_akun || '-'
}}</TableCell>
                        <TableCell class="text-right">{{
formatRupiah(item.debit) }}</TableCell>
                        <TableCell class="text-right">{{
formatRupiah(item.kredit) }}</TableCell>
                        <TableCell rowspan="2" v-if="index % 2
== 0" class="text-center print:hidden">
                            <div class="flex justify-center gap-
2">
                                <Link :href="route('jurnal-
penyesuaian.edit', item.jurnal_id ?? item.id) +
`?month=${month}`">
                                    <Button size="icon"><Pen
/></Button>
                                </Link>
                                <Button
@click="hapus(item.jurnal_id ?? item.id)" variant="destructive"
```

```
size="icon"><Trash /></Button>
                                </div>
                            </TableCell>
                        </TableRow>
                        <TableRow>
                            <TableCell></TableCell>
                            <TableCell class="text-center font-
bold">Total</TableCell>
                            <TableCell></TableCell>
                            <TableCell class="text-right font-
bold">{{ formatRupiah(sumJurnalUmum('debit')) }}</TableCell>
                            <TableCell class="text-right font-
bold">{{ formatRupiah(sumJurnalUmum('kredit')) }}</TableCell>
                            <TableCell
class="print:hidden"></TableCell>
                        </TableRow>
                    </TableBody>
                </Table>
                <Alert v-else variant="destructive">
                    <InfoIcon class="h-4 w-4" />
                    <AlertTitle>Tidak ada data Jurnal
Penyesuaian.</AlertTitle>
                    <AlertDescription>
                        Silakan tambahkan transaksi Jurnal
Penyesuaian baru untuk bulan {{ formatMonth(month as string) }}.
                    </AlertDescription>
                </Alert>
            </div>
        </AppLayout>
</template>
<script setup lang="ts">
import { Alert, AlertDescription, AlertTitle } from
'@/components/ui/alert';
import { Button } from '@/components/ui/button';
import PrintButton from
'@/components/ui/button/PrintButton.vue';
import { Input } from '@/components/ui/input';
import { Table, TableBody, TableCell, TableHead, TableHeader,
TableRow } from '@/components/ui/table';
import AppLayout from '@/layouts/AppLayout.vue';
import { formatMonth } from '@/lib/utils';
import { BreadcrumbItem } from '@/types';
import { Link, router } from '@inertiajs/vue3';
import { InfoIcon, Pen, Trash } from 'lucide-vue-next';
import { onMounted, ref } from 'vue';
import { JurnalUmum } from './types';
const props = defineProps<{ data: any[]; month: string }>();
const formatRupiah = (val: number | string): string => {
    const n = parseInt(String(val).replace(/[^\d]/g, '')) || 0;
    return 'Rp. ' + n.toLocaleString('id-ID');
};
```

5) Halaman Neraca Saldo

Berikut Berikut adalah tampilan Neraca Saldo pada Aplikasi sistem laporan keuangan UMKM. Halaman Neraca Saldo dalam aplikasi ini berfungsi untuk menampilkan daftar Neraca Saldo, Menambahkan data serta menyediakan menu print pada halaman Neraca Saldo.



Gambar 4 26 Halaman Neraca Saldo

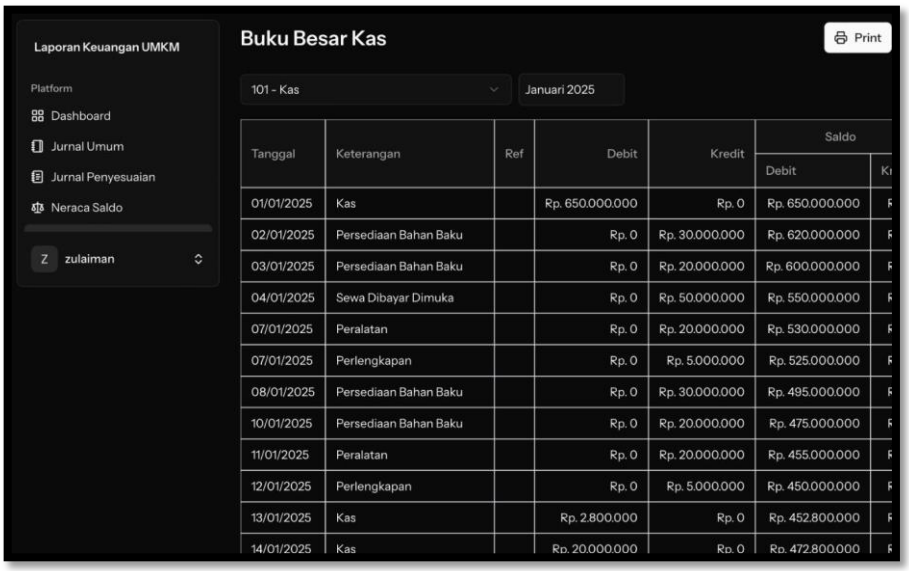
```
<template>
    <AppLayout :breadcrumbs="breadcrumbs">
        <div class="w-full p-4">
            <div class="mb-4 flex items-center justify-between">
                <h1 class="text-2xl font-bold">
                    Data Neraca Saldo <span class="hidden
print:inline"> - {{ formatMonth(bulan as string) }}</span>
                </h1>
                <PrintButton />
            </div>
            <div class="mt-4 flex items-center gap-2
print:hidden">
                <Input
                    type="month"
                    placeholder="Cari berdasarkan keterangan,
ref, debit, atau kredit"
                    class="mb-4 w-max"
                    @update:model-value="filterMonth"
                    v-model="month"
                />
```

```
  </div>
            <Table v-if="items.length > 0">
                <TableHeader>
                    <TableRow>
                        <TableHead>Kode Akun</TableHead>
                        <TableHead>Nama Akun</TableHead>
                        <TableHead class="text-
right">Debit</TableHead>
                        <TableHead class="text-
right">Kredit</TableHead>
                    </TableRow>
                </TableHeader>
                <TableBody>
                    <TableRow v-for="item in items"
:key="item.id">
                        <TableCell>{{ item.kode_akun || '-'
}}</TableCell>
                        <TableCell>{{ item.nama_akun
}}</TableCell>
                        <TableCell class="text-right">{{
formatRupiah(item.debit) }}</TableCell>
                        <TableCell class="text-right">{{
formatRupiah(item.kredit) }}</TableCell>
                    </TableRow>
                    <TableRow>
                        <TableCell></TableCell>
                        <TableCell class="text-center font-
bold">Total</TableCell>
                        <TableCell class="text-right font-
bold">{{
                            formatRupiah(items.reduce((acc,
item) => acc + parseInt(item.debit ?? 0), 0))
                        }}</TableCell>
                        <TableCell class="text-right font-
bold">{{
                            formatRupiah(items.reduce((acc,
item) => acc + parseInt(item.kredit ?? 0), 0))
                        }}</TableCell>
                    </TableRow>
                </TableBody>
            </Table>
            <Alert v-else variant="destructive">
                <InfoIcon class="h-4 w-4" />
                <AlertTitle>Tidak ada data Neraca
Saldo.</AlertTitle>
                <AlertDescription> Silakan tambahkan transaksi
jurnal umum baru untuk bulan {{ formatMonth(bulan as string) }}.
</AlertDescription>
            </Alert>
        </div>
    </AppLayout>
</template>
<script setup lang="ts">
import { Alert, AlertDescription, AlertTitle } from
'@/components/ui/alert';
import PrintButton from
```

```
'@/components/ui/button/PrintButton.vue';
import { Input } from '@/components/ui/input';
import { Table, TableBody, TableCell, TableHead, TableHeader,
TableRow } from '@/components/ui/table';
import AppLayout from '@/layouts/AppLayout.vue';
import { formatMonth, getCurrentYearMonth } from '@/lib/utils';
import { BreadcrumbItem } from '@/types';
import { router } from '@inertiajs/vue3';
import { InfoIcon } from 'lucide-vue-next';
import { AcceptableValue } from 'reka-ui';
import { ref } from 'vue';
import { type NeracaSaldo } from './types';
const { data, bulan } = defineProps<{ data: NeracaSaldo[];
bulan?: string }>();
const items = ref([...data]);
function formatRupiah(val: number | string | null) {
    const n = Number(val ?? 0);
    return 'Rp. ' + n.toLocaleString('id-ID');
}
const breadcrumbs: BreadcrumbItem[] = [
    {
        title: 'Dashboard',
        href: '/dashboard',
    },
    {
        title: 'Neraca Saldo',
        href: '/neraca-saldo',
    },
];
const month = ref<string>(bulan ?? getCurrentYearMonth());
const filterMonth = (value: AcceptableValue) => {
    router.get(
        route('neraca.index'),
        { bulan: value },
        {
            preserveScroll: true,
        },
    );
};
</script>
```

6) Halaman Buku Besar

Berikut Berikut adalah tampilan Buku Besar pada Aplikasi sistem laporan keuangan UMKM. Halaman Buku Besar dalam aplikasi ini berfungsi untuk menampilkan daftar Buku Besar, Menambahkan data serta menyediakan menu print pada halaman Buku Besar.



Gambar 4 27 Halaman Buku Besar

```
// □ File: resources/js/Pages/JurnalUmum/Index.vue
<template>
    <AppLayout :breadcrumbs="breadcrumbs">
        <div class="p-4">
            <div class="mb-4 flex items-center justify-between">
                <h1 class="text-2xl font-bold">
                    Buku Besar {{ items?.nama_akun }} <span
class="hidden print:inline">- {{ formatMonth(bulan as string)
}}</span>
                </h1>
                <PrintButton />
            </div>
            <div class="mt-4 flex items-center gap-2
print:hidden">
                <Select @update:modelValue="filterAkuns" v-
model="akun">
                    <SelectTrigger class="mt-2 mb-4 w-full max-
w-xs">
                        <SelectValue placeholder="Keterangan" />
                    </SelectTrigger>
                    <SelectContent>
```

```
                        <SelectItem selected v-for="akun in
akuns" :key="akun.kode_akun" :value="akun.kode_akun">
                            {{ akun.kode_akun }} - {{
akun.nama_akun }}
                        </SelectItem>
                    </SelectContent>
                </Select>
                <Input
                    type="month"
                    placeholder="Cari berdasarkan keterangan,
ref, debit, atau kredit"
                    class="mb-4 w-max"
                    @update:model-value="filterMonth"
                    v-model="month"
                />
            </div>
            <Table v-if="items && items.jurnal_umum.length > 0"
class="h-60 w-full">
                <TableHeader>
                    <TableRow>
                        <TableHead
rowspan="2">Tanggal</TableHead>
                        <TableHead
rowspan="2">Keterangan</TableHead>
                        <TableHead rowspan="2">Ref</TableHead>
                        <TableHead rowspan="2" class="text-
right">Debit</TableHead>
                        <TableHead rowspan="2" class="text-
right">Kredit</TableHead>
                        <TableHead colspan="2" class="text-
center">Saldo</TableHead>
                    </TableRow>
                    <TableRow>
                        <TableHead>Debit</TableHead>
                        <TableHead>Kredit</TableHead>
                    </TableRow>
                </TableHeader>
                <TableBody>
                    <TableRow v-for="(jurnal_umum, index) in
items.jurnal_umum" :key="index">
                        <TableCell>{{ jurnal_umum.formatted_date
}}</TableCell>
                        <TableCell>{{
jurnal_umum.jurnal_parent.keterangan ??
jurnal_umum.jurnal_parent.akun?.nama_akun }}</TableCell>
                        <TableCell>{{ jurnal_umum.ref
}}</TableCell>
                        <TableCell class="text-right">{{
formatRupiah(jurnal_umum.debit) }}</TableCell>
                        <TableCell class="text-right">{{
formatRupiah(jurnal_umum.kredit) }}</TableCell>
                        <TableCell class="text-right">{{
formatRupiah(saldo(jurnal_umum, index).debit) }}</TableCell>
                        <TableCell class="text-right">{{
formatRupiah(saldo(jurnal_umum, index).kredit) }}</TableCell>
                    </TableRow>
```

```
                </TableBody>
            </Table>
            <Alert v-else variant="destructive">
                <InfoIcon class="h-4 w-4" />
                <AlertTitle>Tidak ada data jurnal
umum.</AlertTitle>
                <AlertDescription>
                    <div>Silakan tambahkan <Link
class="underline" :href="route('jurnal.index')"> Jurnal umum
baru</Link>.</div>
                </AlertDescription>
            </Alert>
        </div>
    </AppLayout>
</template>
<script setup lang="ts">
import { Alert, AlertDescription, AlertTitle } from
'@/components/ui/alert';
import PrintButton from
'@/components/ui/button/PrintButton.vue';
import { Input } from '@/components/ui/input';
import { Select, SelectContent, SelectItem, SelectTrigger,
SelectValue } from '@/components/ui/select';
import { Table, TableBody, TableCell, TableHead, TableHeader,
TableRow } from '@/components/ui/table';
import AppLayout from '@/layouts/AppLayout.vue';
import { formatMonth, getCurrentYearMonth } from '@/lib/utils';
import { BreadcrumbItem } from '@/types';
import { Link, router } from '@inertiajs/vue3';
import { InfoIcon } from 'lucide-vue-next';
import { AcceptableValue } from 'reka-ui';
import { onMounted, ref } from 'vue';
import { Akun, JurnalUmum } from './types';
const props = defineProps<{ akuns: Akun[]; bulan: string;
allAkuns: Akun[] }>();
const formatRupiah = (val: number | string): string => {
    const n = parseInt(String(val).replace(/[^\d]/g, '')) || 0;
    return 'Rp. ' + n.toLocaleString('id-ID');
};
const breadcrumbs: BreadcrumbItem[] = [
    {
        title: 'Dashboard',
        href: '/dashboard',
    },
    {
```

7) Halaman Neraca Jalur

Berikut Berikut adalah tampilan Neraca Jalur pada Aplikasi sistem laporan keuangan UMKM. Halaman Neraca Jalur aplikasi ini berfungsi untuk menampilkan daftar Neraca Jalur, Menambahkan data serta menyediakan menu print pada halaman Neraca Jalur.



Gambar 4 28 Halaman Neraca Jalur

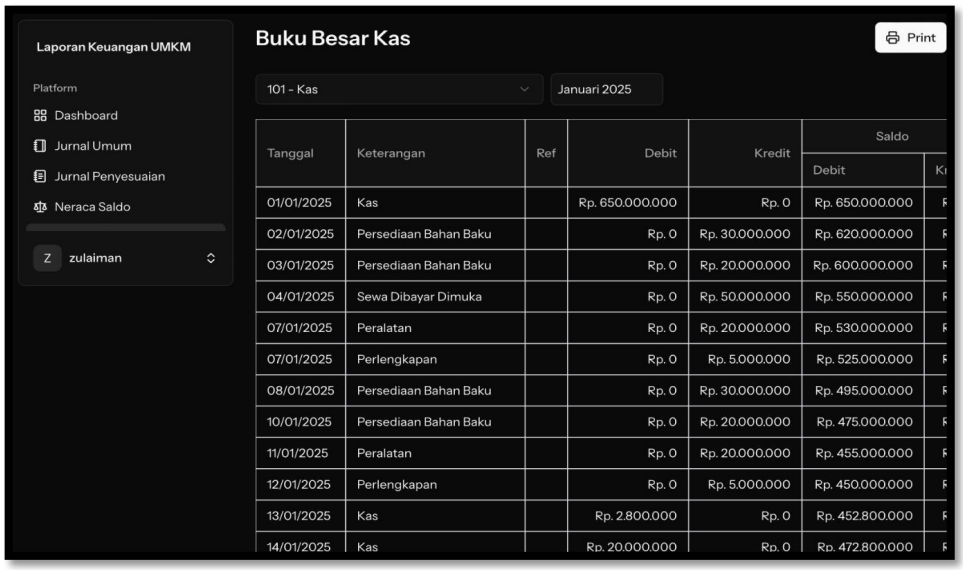
```
// □ File: resources/js/Pages/JurnalUmum/Index.vue
<template>
    <AppLayout :breadcrumbs="breadcrumbs">
        <div class="p-4">
            <div class="mb-4 flex items-center justify-between">
                <h1 class="text-2xl font-bold">
                    Buku Besar {{ items?.nama_akun }} <span
class="hidden print:inline">- {{ formatMonth(bulan as string)
}}</span>
                </h1>
                <PrintButton />
            </div>
            <div class="mt-4 flex items-center gap-2
print:hidden">
                <Select @update:modelValue="filterAkuns" v-
model="akun">
                    <SelectTrigger class="mt-2 mb-4 w-full max-
w-xs">
                        <SelectValue placeholder="Keterangan" />
                    </SelectTrigger>
                    <SelectContent>
```

```
                        <SelectItem selected v-for="akun in
akuns" :key="akun.kode_akun" :value="akun.kode_akun">
                            {{ akun.kode_akun }} - {{
akun.nama_akun }}
                        </SelectItem>
                    </SelectContent>
                </Select>
                <Input
                    type="month"
                    placeholder="Cari berdasarkan keterangan,
ref, debit, atau kredit"
                    class="mb-4 w-max"
                    @update:model-value="filterMonth"
                    v-model="month"
                />
            </div>
            <Table v-if="items && items.jurnal_umum.length > 0"
class="h-60 w-full">
                <TableHeader>
                    <TableRow>
                        <TableHead
rowspan="2">Tanggal</TableHead>
                        <TableHead
rowspan="2">Keterangan</TableHead>
                        <TableHead rowspan="2">Ref</TableHead>
                        <TableHead rowspan="2" class="text-
right">Debit</TableHead>
                        <TableHead rowspan="2" class="text-
right">Kredit</TableHead>
                        <TableHead colspan="2" class="text-
center">Saldo</TableHead>
                    </TableRow>
                    <TableRow>
                        <TableHead>Debit</TableHead>
                        <TableHead>Kredit</TableHead>
                    </TableRow>
                </TableHeader>
                <TableBody>
                    <TableRow v-for="(jurnal_umum, index) in
items.jurnal_umum" :key="index">
                        <TableCell>{{ jurnal_umum.formatted_date
}}</TableCell>
                        <TableCell>{{
jurnal_umum.jurnal_parent.keterangan ??
jurnal_umum.jurnal_parent.akun?.nama_akun }}</TableCell>
                        <TableCell>{{ jurnal_umum.ref
}}</TableCell>
                        <TableCell class="text-right">{{
formatRupiah(jurnal_umum.debit) }}</TableCell>
                        <TableCell class="text-right">{{
formatRupiah(jurnal_umum.kredit) }}</TableCell>
                        <TableCell class="text-right">{{
formatRupiah(saldo(jurnal_umum, index).debit) }}</TableCell>
                        <TableCell class="text-right">{{
formatRupiah(saldo(jurnal_umum, index).kredit) }}</TableCell>
                    </TableRow>
```

```
                </TableBody>
            </Table>
            <Alert v-else variant="destructive">
                <InfoIcon class="h-4 w-4" />
                <AlertTitle>Tidak ada data jurnal
umum.</AlertTitle>
                <AlertDescription>
                    <div>Silakan tambahkan <Link
class="underline" :href="route('jurnal.index')"> Jurnal umum
baru</Link>.</div>
                </AlertDescription>
            </Alert>
        </div>
    </AppLayout>
</template>
<script setup lang="ts">
import { Alert, AlertDescription, AlertTitle } from
'@/components/ui/alert';
import PrintButton from
'@/components/ui/button/PrintButton.vue';
import { Input } from '@/components/ui/input';
import { Select, SelectContent, SelectItem, SelectTrigger,
SelectValue } from '@/components/ui/select';
import { Table, TableBody, TableCell, TableHead, TableHeader,
TableRow } from '@/components/ui/table';
import AppLayout from '@/layouts/AppLayout.vue';
import { formatMonth, getCurrentYearMonth } from '@/lib/utils';
import { BreadcrumbItem } from '@/types';
import { Link, router } from '@inertiajs/vue3';
import { InfoIcon } from 'lucide-vue-next';
import { AcceptableValue } from 'reka-ui';
import { onMounted, ref } from 'vue';
import { Akun, JurnalUmum } from './types';
const props = defineProps<{ akuns: Akun[]; bulan: string;
allAkuns: Akun[] }>();
const formatRupiah = (val: number | string): string => {
    const n = parseInt(String(val).replace(/[^\d]/g, '')) || 0;
    return 'Rp. ' + n.toLocaleString('id-ID');
};
const breadcrumbs: BreadcrumbItem[] = [
    {
        title: 'Dashboard',
        href: '/dashboard',
    },
    {
```

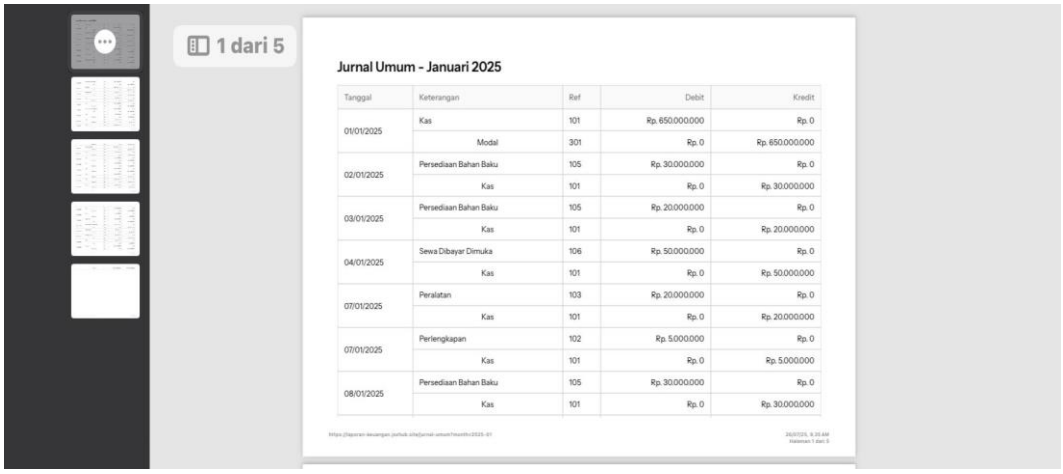
8) Halaman Laporan Keuangan

Berikut Berikut adalah tampilan Laporan Keuanganpada Aplikasi sistem laporan keuangan UMKM. Halaman Laporan Keuanganaplikasi ini berfungsi untuk menampilkan daftar Laporan Keuangan, Menambahkan data serta menyediakan menu print pada halaman Laporan Keuangan.



Gambar 4 29 Halaman Laporan Keuangan

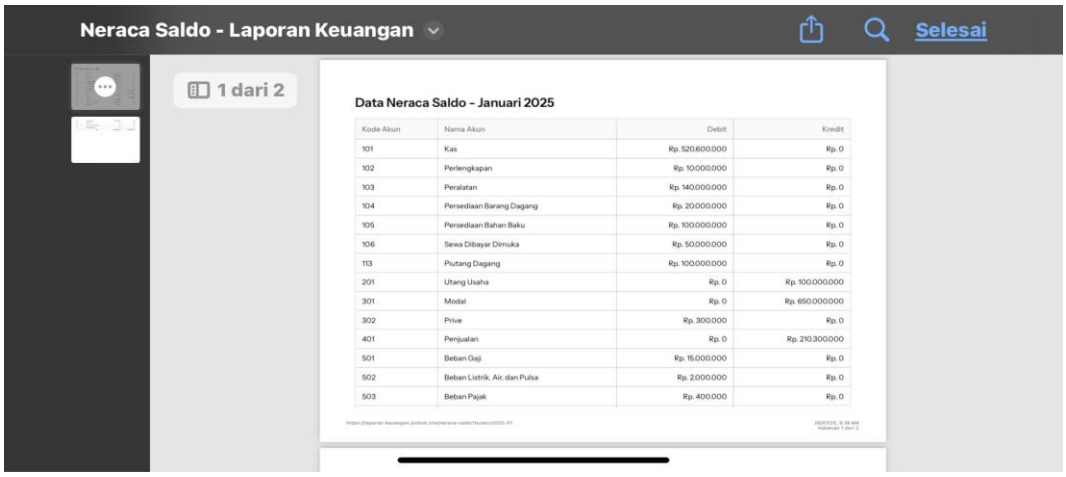
9) Halaman Hasil Print



Hasil Jurnal Umum



Hasil Jurnal Penyesuaian



Hasil Neraca Saldo



Hasil Laporan Keuangan

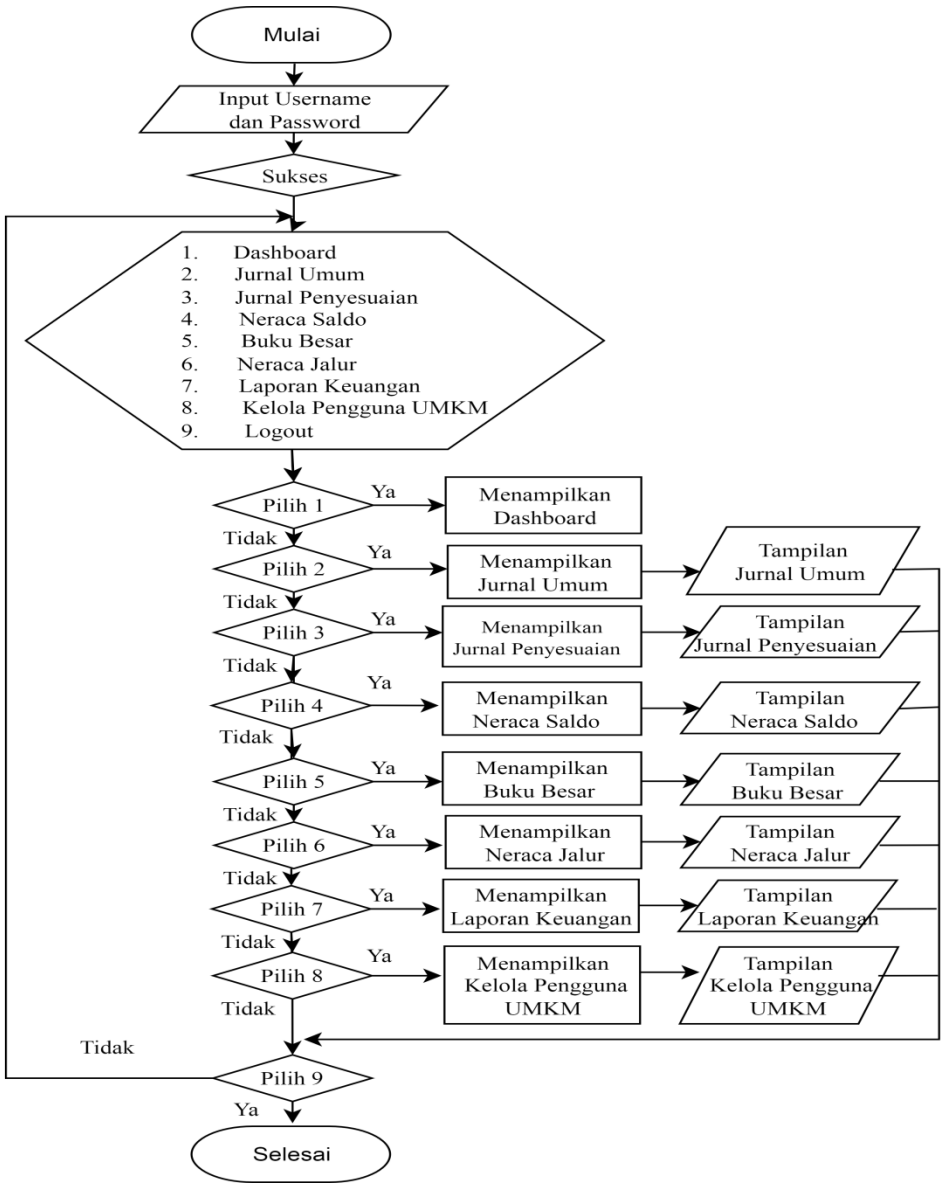
Gambar 4 30 Halaman Hasil Print

6. Pengujian Sistem

a. *WhiteBox*

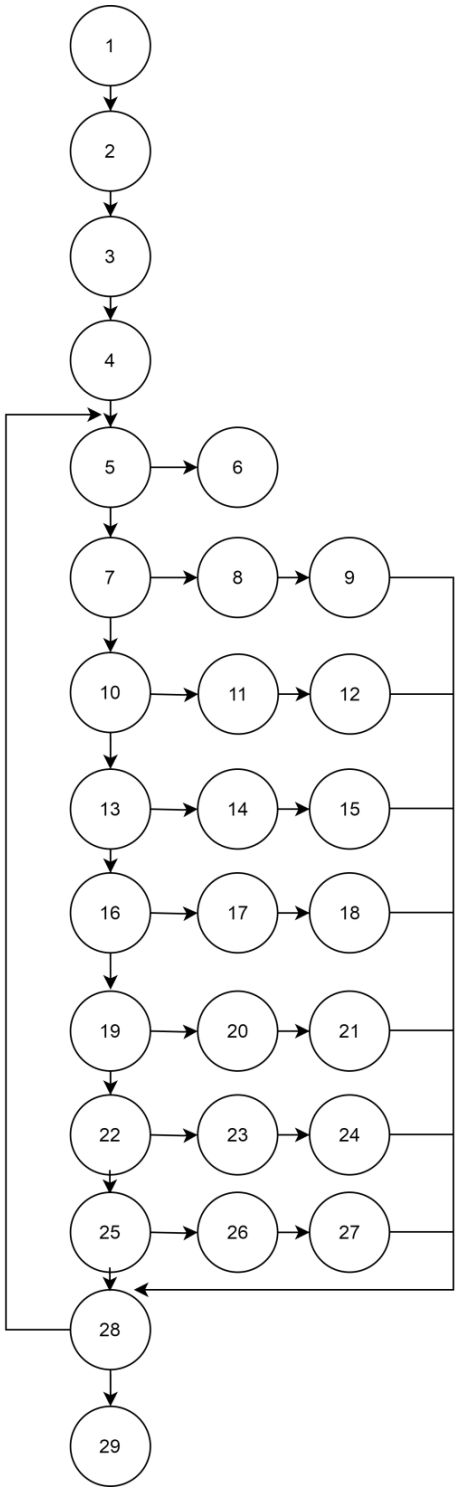
Pengujian aplikasi dilakukan dengan cara pengujian *WhiteBox*. *WhiteBox* adalah metode pengujian perangkat lunak yang menguji struktur internal atau cara kerja aplikasi.

1) Flowchart dan *Flowgraph* Aktivitas *Admin* dan Pengguna UMKM



Gambar 4 14. *Flowchart* Aktivitas *Admin* dan Pengguna UMKM

Dari Flowchart yang digunakan untuk pengujian perangkat lunak, maka ditentukan Flowgraph sebagai berikut:



Gambar 4. 32. *Flowgraph* Aktivitas Admin dan Pengguna UMKM

Dari *Flowgraph* aktivitas *admin* dan Pengguna UMKM diatas dapat dilakukan proses perhitungan sebagai berikut:

a) Menghitung *Cyclomatic Complexcity* V(G) dari *Egde* dan *Node*:

Diketahui :

N = 22 E = 29 R = 2

Penyelasaian :

V(G) = (E- N) + 2

= (29 – 22) + 2 = 9

Independent Path :

Path 1 = 1 – 2 – 3 – 4 – 5 –6

Path 2 = 1 – 2 – 3 – 5 – 7 – 8 –9 – 29

Path 3 = 1 – 2 – 3 – 5 – 7 – 10 –11 –12– 29

Path 4 = 1 – 2 – 3 – 5 – 7 – 10 –13 –14 – 15 –29

Path 5 = 1 – 2 – 3 – 5 – 7 – 10 –13 –16 – 17– 18 –29

Path 6 = 1 – 2 – 3 – 5 – 7 – 10 –13 –16 – 19– 20 –21–29

Path 7 = 1 – 2 – 3 – 5 – 7 – 10 –13 –16 – 19– 22 –23–24–29

Path 8 = 1 – 2 – 3 – 5 – 7 – 10 –13 –16 – 19– 22 –25–26–27–29

Path 9 = 1 – 2 – 3 – 5 – 7 – 10 –13 –16 – 19– 22–25–28–29

Tabel 4 1. *White Box* Perhitungan *menu Admin* dan *User*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	E-1
1		1																			1-1=0
2			1																		1-1=0
3				1																	1-1=0
4					1																1-1=0
5						1	1														2-1=1
6																					0
7								1		1											2-1=1
8									1												1-1=0
9																					1-1=0
10											1		1								2-1=1

11											1									1-1=0
12																				1-1=0
13													1		1					2-1=1
14														1						1-1=0
15																				1-1=0
16																1		1		2-1=1
17																	1			1-1=0
18																				1-1=0
19																			1	2-1=1
20																				0
SUM(E+1)																				8+1=9

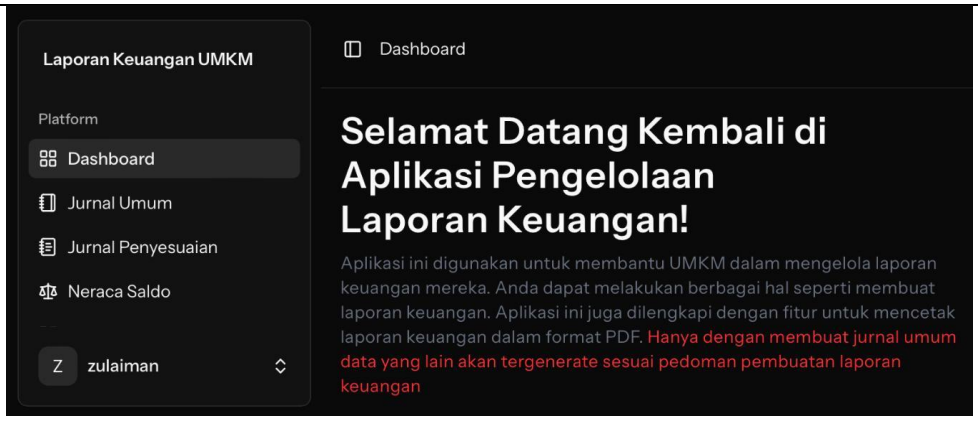
b. *Black box*

Pengujian *BlackBox* digunakan untuk menguji sistem. *BlackBox* adalah metode pengujian perangkat lunak yang memeriksa fungsionalitas aplikasi tanpa mempelajari struktur atau cara kerja internalnya. Pendekatan pengujian ini berlaku untuk hampir semua tingkat pengujian perangkat lunak, termasuk unit, integrasi, sistem, dan penerimaan.

1) Black Box Pengguna UMKM

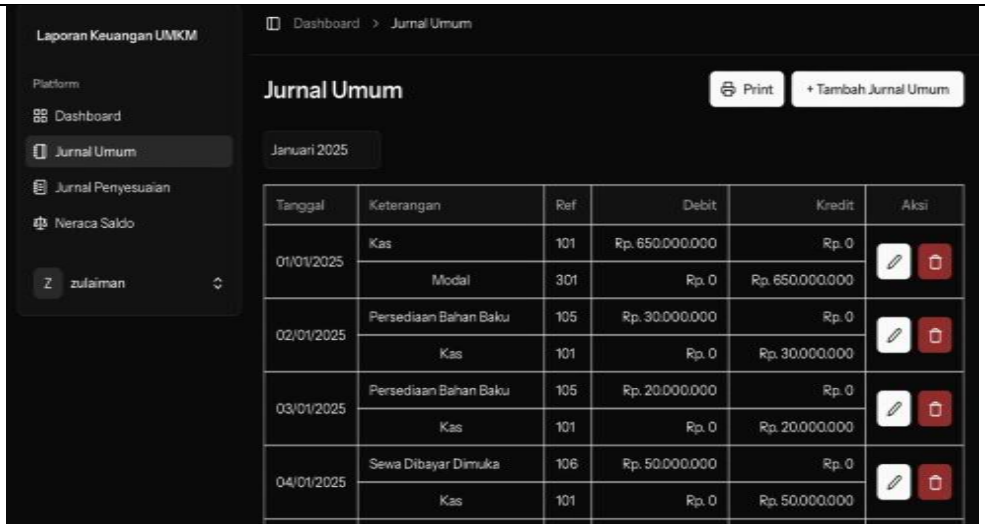
Tabel 4 19. *BlackBox* Halaman Utama

Test Faktor	Hasil	Kesimpulan
Jika memasukkan username dan password dengan benar.	✓	Informasi, tampil halaman utama pengguna UMKM.
Screen Shot		



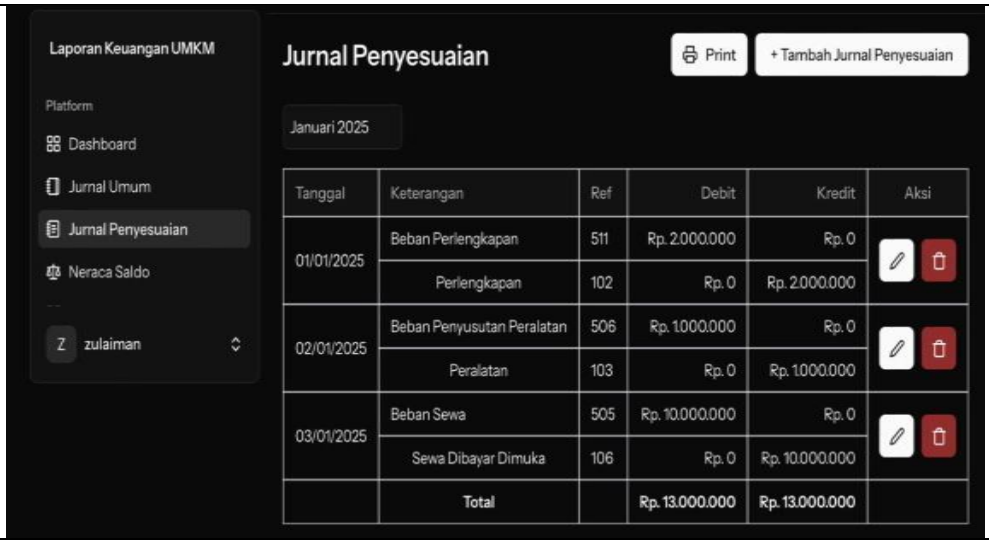
Tabel 4 2. *BlackBox Form* Jurnal Umum

Test Faktor	Hasil	Kesimpulan
Jika memilih menu Jurnal Umum pada halaman utama.	✓	Informasi, tampil form Jurnal Umum
Screen Shot		



Tabel 4 3. *BlackBox Form* Jurnal Penyesuaian

Test Faktor	Hasil	Kesimpulan
Jika memilih menu Jurnal Penyesuaian pada halaman utama.	✓	Informasi, tampil form Jurnal Penyesuaian.
Screen Shot		



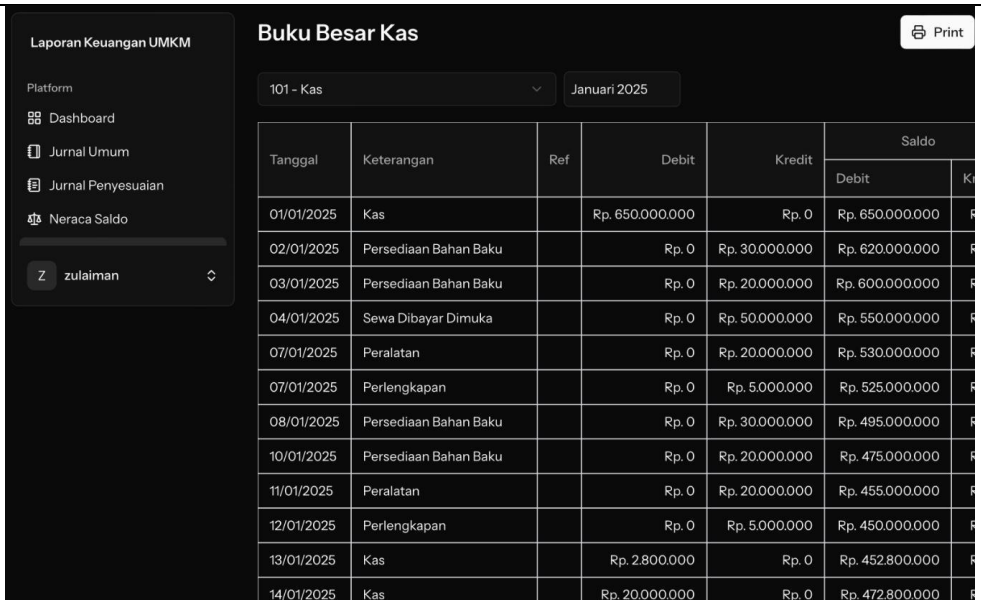
Tabel 4 4. *BlackBox Form* Neraca Saldo

Test Faktor	Hasil	Kesimpulan
Jika memilih menu chat pada Neraca Saldo	✓	Informasi, tampil form Neraca Saldo.
Screen Shot		



Tabel 4 5. *BlackBox Form* Buku Besar Kas

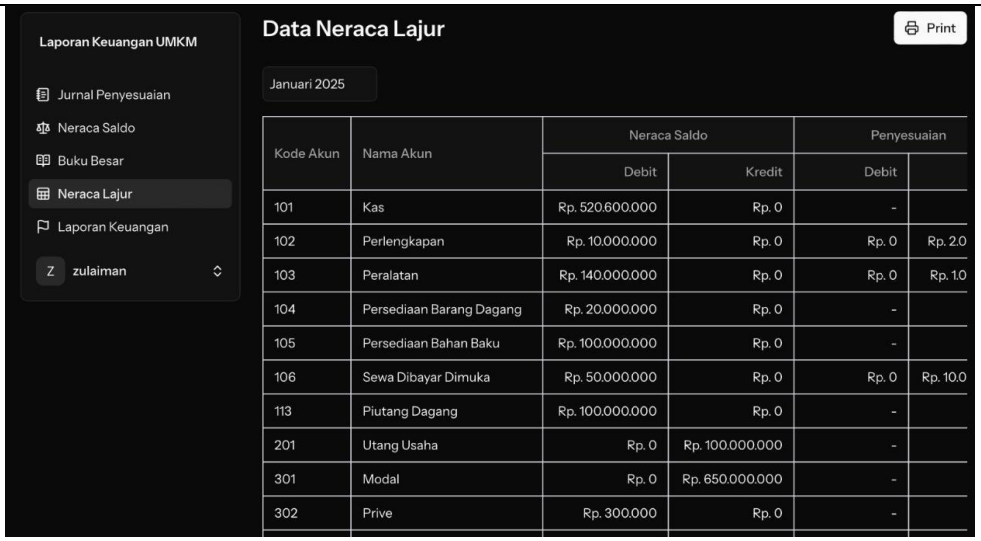
Test Faktor	Hasil	Kesimpulan
Jika memilih menu chat pada Buku Besar Kas	✓	Informasi, tampil form Buku Besar Kas.
Screen Shot		



Tabel 4 6. *BlackBox Form* Neraca Jalur

Test Faktor	Hasil	Kesimpulan
Jika memilih menu chat pada Neraca Jalur	✓	Informasi, tampil form Neraca Jalur.

Screen Shot

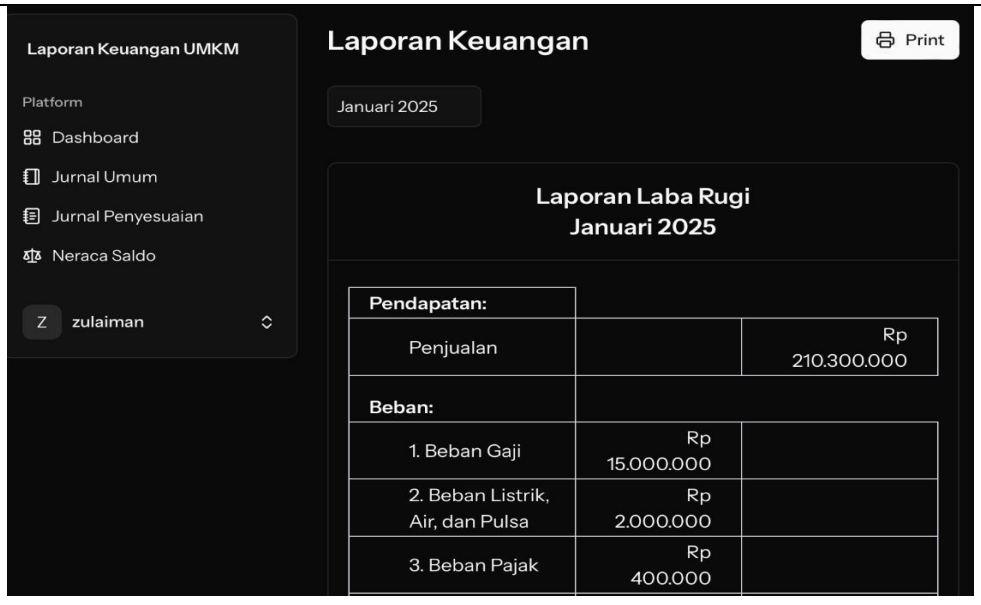


Kode Akun	Nama Akun	Neraca Saldo		Penyesuaian	
		Debit	Kredit	Debit	
101	Kas	Rp. 520.600.000	Rp. 0	-	
102	Perlengkapan	Rp. 10.000.000	Rp. 0	Rp. 0	Rp. 2.0
103	Peralatan	Rp. 140.000.000	Rp. 0	Rp. 0	Rp. 1.0
104	Persediaan Barang Dagang	Rp. 20.000.000	Rp. 0	-	
105	Persediaan Bahan Baku	Rp. 100.000.000	Rp. 0	-	
106	Sewa Dibayar Dimuka	Rp. 50.000.000	Rp. 0	Rp. 0	Rp. 10.0
113	Piutang Dagang	Rp. 100.000.000	Rp. 0	-	
201	Utang Usaha	Rp. 0	Rp. 100.000.000	-	
301	Modal	Rp. 0	Rp. 650.000.000	-	
302	Prive	Rp. 300.000	Rp. 0	-	

Tabel 4 7. *BlackBox Form* Laporan Keuangan

Test Faktor	Hasil	Kesimpulan
Jika memilih menu chat pada Laporan Keuangan	✓	Informasi, tampil form Laporan Keuangan.

Screen Shot

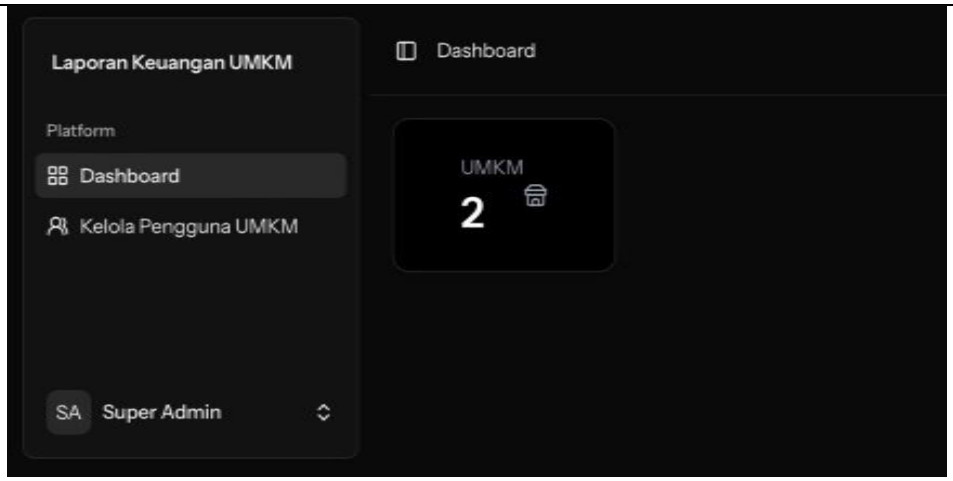


Pendapatan:		
Penjualan		Rp 210.300.000
Beban:		
1. Beban Gaji	Rp 15.000.000	
2. Beban Listrik, Air, dan Pulsa	Rp 2.000.000	
3. Beban Pajak	Rp 400.000	

2) Black Box Admin

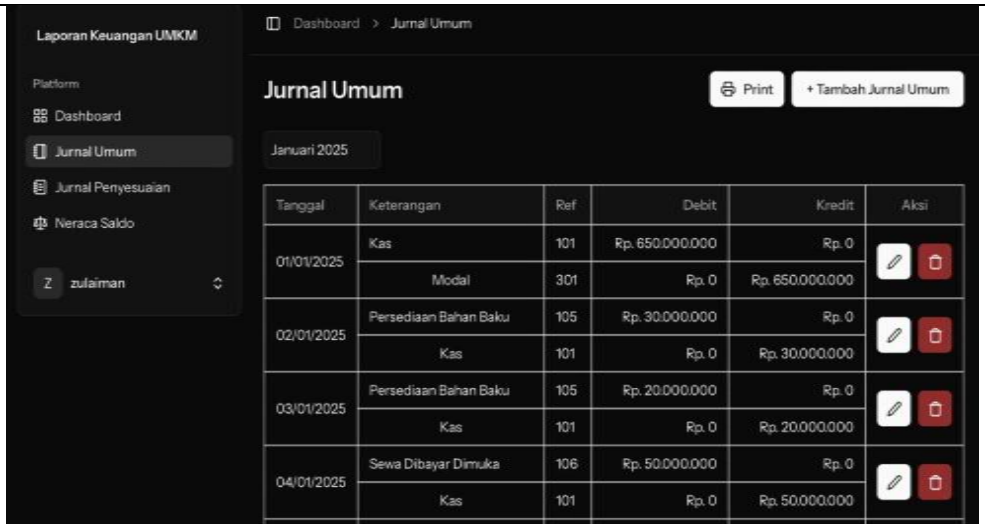
Tabel 4 19. *BlackBox* Halaman Utama Admin

Test Faktor	Hasil	Kesimpulan
Jika memasukkan username dan password dengan benar.	✓	Informasi, tampil halaman utama *admin*.
Screen Shot		



Tabel 4 8. *BlackBox Form* Kelola Pengguna UMKM

Test Faktor	Hasil	Kesimpulan
Jika memilih menu Jurnal Umum pada halaman Kelola Pengguna UMKM.	✓	Informasi, tampil form Kelola Pengguna UMKM
Screen Shot		



BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan sebagai beriku:

1. Fitur Lengkap dan Terintegrasi**,** Sistem dilengkapi dengan fitur utama seperti pencatatan jurnal umum, jurnal penyesuaian, neraca saldo, buku besar, neraca lajur, serta pembuatan laporan keuangan otomatis, serta laporan dapat di print setiap bulannya.
2. Sifat *Application Service*, Aplikasi bersifat layanan aplikasi (*application service*), di mana setiap UMKM dapat mendaftar dan akun akan diaktifkan setelah dikonfirmasi oleh admin.
3. Aksesibilitas Tinggi, Karena berbasis web, sistem dapat diakses kapan saja dan dari perangkat apa pun yang terhubung internet, dan aplikasi diperuntukkan untuk semua pelaku UMKM.
4. Semua hasil laporan dari jurnal umum, jurnal penyesuaian, neraca saldo, buku besar, neraca lajur, serta laporan keuangan dapat di download dalam bentuk pdf. https://laporan-keuangan.joxhub.site/

B. Saran

Untuk pengembangan lebih lanjut, berikut adalah beberapa saran yang dapat dipertimbangkan:

1. Penambahan Fitur Pendukung Usaha Disarankan agar sistem ini ke depannya dapat dikembangkan dengan menambahkan fitur tambahan seperti pengelolaan stok barang**,** manajemen hutang piutang**, serta** pengingat transaksi **(**reminder) yang akan sangat membantu pelaku UMKM dalam mengatur operasional usaha secara lebih menyeluruh, tidak hanya terbatas pada pencatatan keuangan.
2. Peningkatan Tampilan dan Aksesibilitas Tampilan antarmuka (UI/UX) sebaiknya terus disempurnakan agar lebih menarik dan mudah dipahami, terutama bagi pengguna yang tidak memiliki latar belakang teknologi. Selain itu, penting juga untuk mengembangkan versi mobile responsive agar sistem dapat diakses dengan nyaman melalui smartphone, mengingat sebagian besar pelaku UMKM lebih sering menggunakan perangkat mobile dalam aktivitas sehari-hari.
3. Penguatan Keamanan dan Backup Data Mengingat sistem ini menyimpan data keuangan yang bersifat sensitif, pengembang disarankan untuk meningkatkan keamanan data melalui sistem login yang lebih aman (seperti OTP atau autentikasi dua faktor), serta menyediakan fitur backup otomatis secara berkala agar pengguna tidak kehilangan data apabila terjadi gangguan teknis atau kerusakan perangkat.

DAFTAR PUSTAKA

Adi Nugroho.(2020), *Rekayasa Peragkat Lunak Menggunakan UML dan Java, Pustaka Belajar.* Yogyakarta.

Ardi Rahmawan,(2023). SISTEM INFORMASI PEMASARAN PERUMAHAN BERBASIS WEB, INFONTIKA: Jurnal Pendidikan Informatika Volume 02 Nomor 01.

Ardi Rahmawan,(2023). *SISTEM INFORMASI PEMASARAN PERUMAHAN BERBASIS WEB, INFONTIKA: Jurnal Pendidikan Informatika* Volume 02 Nomor 01.

Awaluddin. (2022). *Simbol simbol pada Flowchart dan penjelasannya*. Fakultas Teknik. Universitas Muhammdiyah Parepare.

Brenda Natalia, (2021), *Pengembangan Aplikasi Solusi Akuntansi Keuangan Umkm (Saku) Berbasis Web Menggunakan Framework Laravel,* Universitas Lampung.

Dedi Riswandi. (2019), *transaksi on-line (e-commerce) : peluang dan tantangan dalam perspektif ekonomi islam,* Jurnal Econetica, Vol.1 No.1.

Endra, Aprilita. (2018). *Pengertian HTML (Hypertext Markup Language)* .

Gilang Ramadhan. (2023). *Contoh USE CASE Diagram Legkap Penjelasan, Dicoding Intren*. (Online), https://www.dicoding.com/blog/contoh-use-case-diagram

Hasan. (2023, *Sistem Informasi PT Calidad Coco Indonesia*. Fakultas Teknik. Universitas Muhammdiyah Parepare.

Kementerian Koperasi dan UKM (2023). "Kontribusi UMKM terhadap Perekonomian Indonesia." Diakses dari www.kemenkopukm.go.id.

Mozilla Developer Network. "Progressive Web Apps."https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps

Muh. Ade Candra (2017) Aplikasi Sistem Evaluasi Dan Pelaporan Alokasi Dana Zis Pada Baznas Berbasis Web. Fakultas Teknik. Universitas Muhammdiyah Parepare.

Rahmah, Z.Z. (2022), *Strategi Peningkatan Pemasaran Melalui Media Sosial Terhadap Umkm Di Desa Kintelan,, Jurnal Budimas,* Vol. 04, No. 01.

Rizki, A., Santoso, T., & Suryana, D. (2022). "Penerapan Sistem Informasi Akuntansi Berbasis Web pada UMKM." *Jurnal Akuntansi Digital*, 5(1). Diakses dari jurnal.undiknas.ac.id.

Rukin, (2019). *Metodologi penelitian kualitatif. Takalar* : Yayasan Ahmar Cendikia Indonesia.

Suryana, Nugroho. (2020). Pengenalan *MySQL, Xampp* Dan *Database*. *Journal of Educational Learning and Innovation*, Volume 2 Number, Halaman 42-72.