E-ISSN: 3025-0994
Halaman 5580-5600
Volume 4 Nomor 4 Tahun 2026

Perancangan Sistem Informasi Laporan Neraca Keuangan dan Kontrol Farm Menggunakan Metode Running Balance Pada BMT Hasanah Ponorogo

Rizky Ali Romadhoni[1], Adi Fajaryanto Cobantoro[2], Ismail Abdurrozzaq Zulkarnain[3]

[1,2,3] Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Ponorogo
e-mail: rizkyalirmd@gmail.com

Abstrak

Pengelolaan laporan keuangan dan kontrol operasional peternakan pada BMT Hasanah Ponorogo masih dilakukan secara manual sehingga menimbulkan keterlambatan penyajian laporan, kesalahan pencatatan transaksi, serta data keuangan dan data *farm* yang tidak terintegrasi dalam satu sumber. Penelitian ini bertujuan merancang dan membangun sistemS informasi laporan neraca keuangan dan kontrol *farm* berbasis *web* yang menerapkan metode *running balance* untuk menghitung saldo secara otomatis dan *real-time* setiap terjadi transaksi. Pengembangan sistem dilakukan menggunakan metode *Waterfall* yang meliputi tahap pengumpulan data, analisis, perancangan, pengkodean, dan pengujian, dengan implementasi menggunakan *framework* Laravel dan basis data MySQL. Pengujian sistem dilakukan menggunakan *Black Box Testing* untuk mengukur fungsionalitas, pengujian akurasi *running balance* dengan membandingkan hasil perhitungan sistem dan manual, pengujian konsistensi data antar modul, serta *System Usability Scale* (SUS) untuk mengukur kemudahan penggunaan. Hasil penelitian menunjukkan bahwa seluruh fitur utama sistem berjalan sesuai kebutuhan, algoritma *running balance* menghasilkan saldo dengan tingkat kesesuaian 100% terhadap perhitungan manual, data pada *dashboard*, *database*, dan laporan konsisten, serta pengujian usability memperoleh skor rata-rata 80,5 yang termasuk kategori Grade B dan *Excellent*. Dengan demikian, sistem yang dikembangkan mampu mendukung pengelolaan keuangan dan kontrol *farm* pada BMT Hasanah Ponorogo secara lebih akurat, efisien, dan terintegrasi.

Kata Kunci: Running Balance, Sistem Informasi Keuangan, Kontrol Farm, Metode Waterfall, Laravel

Abstract

The management of financial reports and livestock operational control at BMT Hasanah Ponorogo is still carried out manually, resulting in delays in report presentation, transaction recording errors, and financial and farm data that are not integrated into a single source. This

study aims to design and build a web-based balance sheet and farm control information system that applies the running balance method to calculate balances automatically and in real time for every transaction. The system was developed using the Waterfall method, covering data collection, analysis, design, coding, and testing, and was implemented using the Laravel framework and a MySQL database. System testing was carried out using Black Box Testing to measure functionality, running balance accuracy testing by comparing system and manual calculation results, data consistency testing between modules, and the System Usability Scale (SUS) to measure ease of use. The results show that all main features of the system work as required, the running balance algorithm produces balances with 100% agreement with manual calculations, the data on the dashboard, database, and reports are consistent, and the usability test obtained an average score of 80.5, which falls into Grade B and the Excellent category. Thus, the developed system is able to support financial management and farm control at BMT Hasanah Ponorogo in a more accurate, efficient, and integrated manner.

Keywords: Running Balance, Financial Information System, Farm Control, Waterfall Method, Laravel

PENDAHULUAN

Perkembangan teknologi informasi yang semakin pesat mendorong berbagai organisasi untuk memanfaatkan sistem informasi berbasis *web* dalam meningkatkan efisiensi dan efektivitas pengelolaan data, khususnya dalam bidang keuangan. Sistem informasi berbasis *web* memungkinkan proses pencatatan, pengolahan, dan penyimpanan data keuangan dilakukan secara lebih cepat, terstruktur, serta dapat diakses secara *real-time* oleh pengguna tanpa dibatasi oleh ruang dan waktu (Widianto & Kurniadi, 2021). Selain itu, penerapan sistem informasi keuangan berbasis *web* juga mampu meningkatkan akurasi data serta meminimalkan kesalahan dalam proses pencatatan transaksi keuangan, terutama pada pengelolaan kas masuk dan kas keluar (Sugiartawan et al., 2023). Dengan adanya sistem yang terkomputerisasi, proses pelaporan keuangan dapat dilakukan secara otomatis sehingga menghasilkan informasi yang lebih akurat dan relevan bagi pengambilan keputusan.

BMT Hasanah Ponorogo sebagai lembaga keuangan mikro syariah memiliki peran penting dalam pengelolaan dana serta pemberdayaan masyarakat, khususnya dalam sektor peternakan. Dalam menjalankan operasionalnya, transparansi dan akuntabilitas dalam pengelolaan keuangan menjadi hal yang sangat penting untuk menjaga kepercayaan anggota dan keberlanjutan program yang dijalankan. Namun, dalam praktiknya, pengelolaan data keuangan masih dilakukan secara manual sehingga menimbulkan berbagai kendala, seperti keterlambatan dalam penyajian laporan, kesalahan pencatatan transaksi, serta kesulitan dalam melakukan pencarian dan pengelolaan data dalam jumlah besar (Suwandhi et al., 2025). Selain itu, sistem manual juga memiliki keterbatasan dalam hal efisiensi dan kecepatan pengolahan data, sehingga kurang mampu mendukung kebutuhan organisasi yang semakin kompleks (Fauzan & Ningsih, 2022).

Permasalahan lain yang dihadapi adalah belum adanya sistem yang terintegrasi antara pengelolaan keuangan dan operasional peternakan. Data keuangan dan data operasional

masih dikelola secara terpisah sehingga menyebabkan proses monitoring menjadi kurang optimal dan informasi yang dihasilkan tidak terpusat (Fadliana & Ahyana, 2025). Kondisi ini mengakibatkan kesulitan dalam melakukan analisis data secara menyeluruh serta menghambat proses pengambilan keputusan yang tepat dan cepat. Keterbatasan dalam penyajian laporan secara otomatis juga menjadi kendala dalam mengevaluasi kinerja keuangan dan operasional secara berkala (Riyanto et al., 2023). Sistem informasi yang terintegrasi mampu meningkatkan efisiensi operasional, mempercepat proses pengolahan data, serta menyediakan informasi yang lebih akurat dan transparan bagi pengguna (Riyanto et al., 2023), dan penerapan sistem informasi akuntansi yang baik juga dapat meningkatkan daya saing organisasi melalui pengelolaan data yang lebih efektif dan sistematis (Budiningrum & Subiyantoro, 2023).

Berdasarkan permasalahan tersebut, diperlukan suatu sistem informasi berbasis *web* yang mampu mengintegrasikan pengelolaan keuangan dan kontrol operasional peternakan dalam satu platform yang terpusat. Sistem ini dirancang dengan menggunakan metode *running balance* untuk menghitung saldo secara otomatis setiap terjadi transaksi, sehingga informasi keuangan dapat disajikan secara *real-time* dan akurat. Dibandingkan dengan metode lain seperti perhitungan periodik atau rekap manual yang membutuhkan penghitungan ulang, *running balance* lebih efisien dan praktis karena saldo langsung terupdate tanpa proses tambahan. Oleh karena itu, metode ini dinilai paling sesuai untuk mendukung sistem informasi keuangan yang membutuhkan kecepatan, ketepatan, dan kemudahan dalam monitoring data.

Rumusan masalah dalam penelitian ini adalah bagaimana (1) merancang dan membangun sistem informasi laporan keuangan berbasis *web* pada BMT Hasanah Ponorogo, (2) mengimplementasikan metode *running balance* untuk menghasilkan perhitungan saldo keuangan secara *real-time* dan akurat, (3) mengintegrasikan pengelolaan data keuangan dengan data operasional peternakan dalam satu sistem yang terpusat, dan (4) meningkatkan efisiensi dan kemudahan dalam proses monitoring keuangan serta kondisi peternakan. Sejalan dengan itu, penelitian ini bertujuan merancang dan membangun sistem informasi laporan keuangan dan kontrol peternakan berbasis *web*, mengimplementasikan metode *running balance*, mengintegrasikan data keuangan dan data peternakan dalam satu sistem yang terstruktur, serta meningkatkan efisiensi dan akurasi pengelolaan data keuangan dan operasional peternakan.

Penelitian ini diharapkan bermanfaat bagi instansi (BMT Hasanah dan BankZiska) dalam mengelola data keuangan secara lebih akurat dan terstruktur, memantau kondisi peternakan secara *real-time*, serta meningkatkan efisiensi dan transparansi pengambilan keputusan; bagi peneliti sebagai pengalaman merancang sistem berbasis *web* dan menerapkan metode *running balance* pada sistem nyata; serta bagi akademik sebagai referensi pengembangan ilmu di bidang sistem informasi. Ruang lingkup penelitian dibatasi pada sistem berbasis *web* (PHP dan *database* MySQL) untuk pengelolaan laporan keuangan dan kontrol peternakan pada BMT Hasanah Ponorogo, dengan data keuangan berupa transaksi kas masuk, kas keluar, dan laporan neraca sederhana serta data peternakan berupa jumlah, kondisi, dan perkembangan ternak. Sistem bersifat *private* dan tidak mencakup integrasi dengan sistem eksternal seperti perbankan atau IoT peternakan.

LANDASAN TEORI

Penelitian Terdahulu

Penelitian terdahulu yang relevan dengan penelitian ini dirangkum pada Tabel 1.

Tabel 1. Penelitian Terdahulu

No	Peneliti & Tahun	Judul Penelitian	Hasil
1	Sugiartawan et al. (2023)	Sistem Informasi Keuangan Pada Koperasi Karya Utama Mandiri	Sistem informasi berbasis *web* mampu mengelola transaksi kas masuk dan kas keluar secara terstruktur serta meningkatkan efisiensi dalam penyusunan laporan keuangan dan monitoring saldo.
2	Suwandhi et al. (2025)	Perancangan Sistem Informasi Keuangan Koperasi Simpan Pinjam Cipta Maju Berbasis *Web*	Sistem yang dikembangkan mampu meningkatkan transparansi dan efisiensi pengelolaan keuangan serta menyediakan akses data secara *real-time* bagi pengguna.
3	Fauzan & Ningsih (2022)	Analisis dan Perancangan Sistem Informasi Akuntansi Simpan Pinjam Koperasi Berbasis *Web*	Sistem mampu mempercepat proses pengolahan data transaksi serta meningkatkan akurasi dalam penyajian laporan keuangan dibandingkan dengan sistem manual.
4	Fadliana & Ahyana (2025)	Perancangan Sistem Informasi Koperasi Online Berbasis *Web* di Lingkungan Kampus	Sistem terintegrasi yang dibangun mampu menghasilkan laporan keuangan secara otomatis dan meningkatkan transparansi serta efektivitas pengelolaan data koperasi.
5	Riyanto et al. (2023)	Sistem Informasi Simpan Pinjam Berbasis *Web* pada Koperasi Jasa Kawan Sejahtera	Sistem membantu pengelolaan data transaksi dan monitoring aktivitas keuangan koperasi secara lebih terstruktur dan efisien.
6	Budiningrum & Subiyantoro (2023)	Sistem Informasi Akuntansi Koperasi Untuk Meningkatkan Daya Saing Pengelolaan Koperasi	Implementasi sistem informasi akuntansi berdampak positif terhadap kinerja organisasi, termasuk efisiensi operasional, kecepatan pengolahan data, serta kualitas informasi untuk pengambilan keputusan.

Penelitian-penelitian tersebut mengembangkan sistem informasi keuangan berbasis *web* yang meningkatkan efisiensi dan akurasi pencatatan transaksi serta penyusunan laporan. Namun, penelitian-penelitian itu masih berfokus pada pengelolaan transaksi keuangan secara umum, belum menerapkan metode perhitungan saldo secara berkelanjutan seperti *running balance*, dan belum mengintegrasikan data operasional lain dalam satu platform terpadu. Penelitian Budiningrum dan Subiyantoro (2023) lebih berfokus pada aspek manajerial dan tidak membahas implementasi sistem secara teknis. Berbeda dengan penelitian-penelitian tersebut, penelitian ini menerapkan metode *running balance* sebagai inti perhitungan saldo keuangan serta mengintegrasikannya dengan sistem kontrol peternakan dalam satu sistem berbasis *web* yang diimplementasikan langsung sebagai aplikasi.

Sistem Informasi dan Teknologi Pengembangan

Sistem informasi adalah suatu sistem di dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi organisasi, serta menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Purnaningsih & Yulianto, 2022). Sistem informasi berbasis *web* banyak digunakan karena kemudahan akses dan fleksibilitasnya dalam mendukung kebutuhan pengguna (Usnaini et al., 2021). *Website* merupakan kumpulan halaman *web* yang saling terhubung dalam sebuah domain atau subdomain pada World Wide *Web* (Setiawan et al., 2022).

PHP (Hypertext Preprocessor) adalah bahasa pemrograman yang banyak digunakan secara luas dan sesuai untuk pengembangan *web* (Fitri, 2021). MySQL adalah sistem manajemen *database* gratis berperforma baik yang menggunakan bahasa SQL (Structured Query Language) untuk pengambilan, manipulasi, dan pengaturan struktur data (Setiawan et al., 2022). *Front end* merupakan bagian *website* yang menampilkan antarmuka kepada pengguna dengan teknologi HTML, CSS, dan JavaScript (Prawastiyo & Hermawan, 2020). Laravel adalah *framework* aplikasi *web* berbasis PHP yang bersifat *open-source* dengan konsep MVC (Model-View-*Controller*) yang memisahkan logika, tampilan, dan data (Septian et al., 2023). Fitur Laravel yang dimanfaatkan dalam penelitian ini meliputi *Eloquent ORM* untuk mengelola *database* MySQL dengan sintaks PHP sederhana, *Blade Templating Engine* untuk membangun tampilan dinamis, *Migration* untuk mengelola skema *database*, fitur keamanan bawaan (proteksi terhadap *SQL Injection*, *Cross-Site Request Forgery*, dan *Cross-Site Scripting*), serta *Artisan CLI* yang mempercepat pembuatan *controller*, model, dan komponen sistem lainnya.

Neraca

Neraca merupakan salah satu laporan keuangan utama yang menggambarkan posisi keuangan suatu entitas pada periode tertentu melalui informasi aset, liabilitas, dan ekuitas. Pada sistem informasi keuangan, neraca berfungsi sebagai output utama dari proses pencatatan transaksi yang dilakukan secara sistematis dan terintegrasi, sehingga penyusunannya dapat dilakukan otomatis berdasarkan data transaksi dalam basis data dan mengurangi potensi kesalahan pencatatan manual (Rahmalia & Komariyah, 2022). Aset (assets) adalah sumber daya ekonomi yang dimiliki entitas dan memiliki manfaat di masa depan; liabilitas (liabilities) adalah kewajiban yang harus dipenuhi kepada pihak lain; dan ekuitas (equity) adalah hak pemilik atas aset setelah dikurangi seluruh kewajiban. Hubungan ketiganya dirumuskan dalam persamaan dasar akuntansi (Trissiyan & Suwandi, 2024):

$$\text{Aset} = \text{Liabilitas} + \text{Ekuitas} \quad (1)$$

Running Balance

Running balance merupakan metode pencatatan keuangan yang menghitung saldo secara terus-menerus setiap terjadi transaksi, sehingga saldo diperbarui otomatis setiap kali terjadi penambahan (debit) atau pengurangan (kredit) dan informasi keuangan selalu terkini (Rahmalia & Komariyah, 2022). Pada pencatatan manual, perhitungan saldo sering dilakukan terpisah setelah transaksi dicatat sehingga berpotensi menimbulkan kesalahan dan keterlambatan informasi; penerapan *running balance* pada sistem berbasis komputer menjadi solusi untuk meningkatkan akurasi dan efisiensi pengolahan data keuangan (Anggraeni, 2023). Secara konsep, saldo awal menjadi dasar perhitungan dan setiap transaksi langsung memengaruhi saldo sebelumnya dengan rumus:

$$\text{Saldo Akhir} = \text{Saldo Sebelumnya} + \text{Debit} - \text{Kredit} \quad (2)$$

Keunggulan metode ini adalah (1) memberikan informasi saldo secara *real-time*, (2) mengurangi risiko kesalahan pencatatan karena perhitungan dilakukan otomatis, (3) meningkatkan efisiensi waktu karena tidak diperlukan perhitungan ulang manual, dan (4) mendukung transparansi dan akuntabilitas karena setiap perubahan saldo dapat ditelusuri berdasarkan riwayat transaksi. Dalam sistem berbasis *web*, metode ini diintegrasikan dengan *database* untuk menyimpan seluruh transaksi secara terstruktur sehingga setiap transaksi yang masuk langsung memperbarui nilai saldo yang tersimpan (Trissiyan & Suwandi, 2024).

Metode Pengujian dan Pemodelan Sistem

Pengujian sistem dilakukan untuk memastikan sistem berjalan sesuai kebutuhan dan menilai kualitasnya dari sudut pandang pengguna. *Black Box Testing* adalah metode pengujian yang berfokus pada fungsionalitas dengan memeriksa kesesuaian masukan (input) dan keluaran (output) tanpa memperhatikan struktur internal program (Amalia et al., 2021), dan efektif menemukan kesalahan pada fungsi sistem (Wulandari et al., 2022). *System Usability Scale* (SUS) adalah metode evaluasi usability berdasarkan persepsi pengguna (Brooke, 1996) melalui kuesioner standar untuk mengukur efektivitas, efisiensi, dan kepuasan pengguna, yang mudah diterapkan dan tidak memerlukan banyak responden (Prabowo & Suprapto, 2021).

Perancangan sistem menggunakan beberapa model. *Use Case Diagram* dalam UML menggambarkan interaksi aktor dengan sistem untuk mendeskripsikan kebutuhan fungsional dari sudut pandang pengguna (Kurniawan, 2018). *Data Flow Diagram* (DFD) menggambarkan aliran data, proses pengolahan, penyimpanan, sumber, dan tujuan data (Ibrahim & Yen, 2010). *Entity Relationship Diagram* (ERD) merupakan model konseptual perancangan basis data yang terdiri atas entitas, atribut, dan relasi, sedangkan *flowchart* merepresentasikan secara grafis urutan langkah proses atau algoritma (Anggraeni, 2023).

METODE PENELITIAN

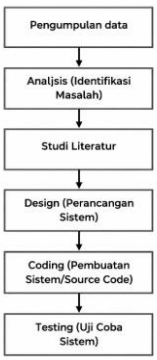
Jenis Penelitian dan Pengumpulan Data

Penelitian ini menggunakan metode rekayasa perangkat lunak (*Software Engineering*) yang berfokus pada perancangan dan pembangunan sistem informasi laporan keuangan dan kontrol *farm* berbasis *web* pada BMT Hasanah Ponorogo, dengan tujuan meningkatkan efisiensi pencatatan transaksi serta akurasi laporan keuangan. Pengumpulan data dilakukan

melalui (1) observasi, yaitu pengamatan langsung terhadap alur transaksi kas masuk dan keluar serta mekanisme pelaporan aset pada unit usaha peternakan kambing; (2) wawancara dengan staf Administrasi Keuangan BMT Hasanah Ponorogo untuk mengetahui sistem yang belum ada dan dibutuhkan; dan (3) studi literatur dari buku, jurnal, dan dokumentasi terkait sistem informasi serta metode *running balance*.

Model Pengembangan Waterfall

Model *Waterfall* dipilih sebagai model pengembangan perangkat lunak karena sifatnya yang sistematis dan berurutan. Setiap fitur kontrol keuangan dan perhitungan neraca dikerjakan secara tuntas pada satu tahap sebelum melangkah ke tahap berikutnya, guna meminimalkan kesalahan fatal dalam perhitungan data keuangan. Tahapan pengembangan ditunjukkan pada Gambar 1.



Gambar 1. Tahapan Perancangan (Model Waterfall)

Tahapan tersebut meliputi: (1) pengumpulan data melalui wawancara dan observasi; (2) analisis masalah, yaitu mempelajari sistem pencatatan yang berjalan dan mengidentifikasi kendala, termasuk data transaksi, kategori biaya, dan inventaris ternak; (3) studi literatur dari buku akuntansi, jurnal terdahulu, dan dokumentasi resmi *framework* Laravel; (4) perancangan sistem menggunakan Aliran Sistem Informasi, Context Diagram, DFD, ERD, dan *flowchart* yang difokuskan pada pemetaan arus kas dan kalkulasi nilai aset unit peternakan kambing; (5) pembuatan sistem (coding) dengan *framework* Laravel, bahasa PHP, dan *database* MySQL, dengan antarmuka *front end* didukung Bootstrap agar responsif; serta (6) pengujian, baik secara lokal maupun melalui deployment ke domain dan hosting, untuk memvalidasi bahwa sistem layak digunakan dan bebas dari error.

Rancangan Pengujian

Pengujian dilakukan dalam empat bentuk. Pertama, *Black Box Testing* terhadap 12 skenario fitur utama (*login*, *dashboard*, transaksi kas, *running balance*, laporan neraca, filter laporan, pengelolaan data kambing, dan *logout*). Kedua, pengujian akurasi *running balance* dengan membandingkan saldo hasil sistem dan perhitungan manual pada 10 transaksi. Ketiga, pengujian konsistensi data antara *dashboard*, *database*, dan laporan. Keempat, pengujian usability dengan *System Usability Scale* (SUS) kepada 20 responden (pengelola, staf administrasi, dan pengguna yang memahami proses pencatatan keuangan serta kontrol *farm*) menggunakan 10 pernyataan berskala Likert 1–5. Skor item ganjil dihitung dari nilai

jawaban dikurangi 1, skor item genap dari 5 dikurangi nilai jawaban, lalu totalnya dikalikan 2,5 sehingga skor SUS berada pada rentang 0–100:

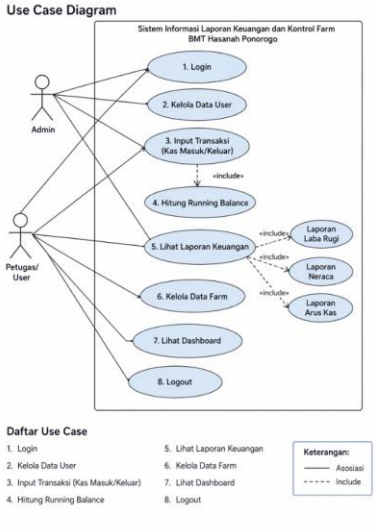
$$\text{Skor SUS} = [\Sigma(Q_{ganjil} - 1) + \Sigma(5 - Q_{genap})] \times 2{,}5 \quad (3)$$

HASIL DAN PEMBAHASAN

Perancangan Sistem

Use Case Diagram

Use Case Diagram sistem ditunjukkan pada Gambar 2. Aktor dalam sistem terdiri atas Admin dan Petugas/User yang memiliki hak akses untuk mengoperasikan sistem sesuai kebutuhan, dengan beberapa fitur yang lebih terfokus pada peran admin. Use case meliputi *login*, *dashboard* (ringkasan total pemasukan, total pengeluaran, saldo bersih, jumlah hewan, dan grafik keuangan), input dan pengelolaan transaksi kas masuk maupun kas keluar yang otomatis diproses dengan metode *running balance*, laporan neraca (aset, kewajiban, dan ekuitas pada periode tertentu), pengelolaan data aset oleh admin, pengelolaan data *farm* (hewan, pakan, dan kondisi kesehatan), riwayat transaksi, grafik keuangan, dan *logout*. Sistem bersifat *private* sehingga hanya pengguna yang memiliki akun yang dapat mengakses seluruh fitur.



Deskripsi Use Case

No	Use Case	Aktor	Deskripsi
1	Login	Admin, Petugas/User	• Pengguna memasukkan username dan password. • Sistem memvalidasi data login. • Jika valid, pengguna masuk ke dashboard sesuai peran.
2	Kelola Data User	Admin	• Admin dapat menambah, mengedit, menghapus, dan melihat data user. • Sistem menyimpan perubahan data user.
3	Input Transaksi (Kas Masuk/Keluar)	Admin, Petugas/User	• Pengguna menginput data transaksi keuangan. • Data yang diinput meliputi tanggal, jenis (masuk/keluar), kategori, jumlah, dan keterangan. • Setelah disimpan, sistem akan memproses running balance.
4	Hitung Running Balance	Sistem	• Sistem mengambil saldo terakhir. • Jika transaksi masuk: saldo baru = saldo sebelumnya + jumlah. • Jika transaksi keluar: saldo baru = saldo sebelumnya - jumlah. • Sistem menyimpan saldo baru ke database.
5	Lihat Laporan Keuangan	Admin, Petugas/User	• Pengguna memilih jenis laporan (Laba Rugi, Neraca, Arus Kas). • Sistem menampilkan laporan sesuai periode yang dipilih. • Laporan dapat dicetak atau diekspor (PDF/Excel).
6	Kelola Data Farm	Admin, Petugas/User	• Pengguna dapat menambah, mengedit, menghapus, dan melihat data farm. • Data farm meliputi informasi hewan, pakan, kesehatan, dan lainnya.
7	Lihat Dashboard	Admin, Petugas/User	• Sistem menampilkan ringkasan data penting seperti total pemasukan, total pengeluaran, saldo bersih, jumlah hewan, dan grafik ringkasan.
8	Logout	Admin, Petugas/User	• Pengguna keluar dari sistem. • Sistem mengakhiri sesi dan kembali ke halaman login.

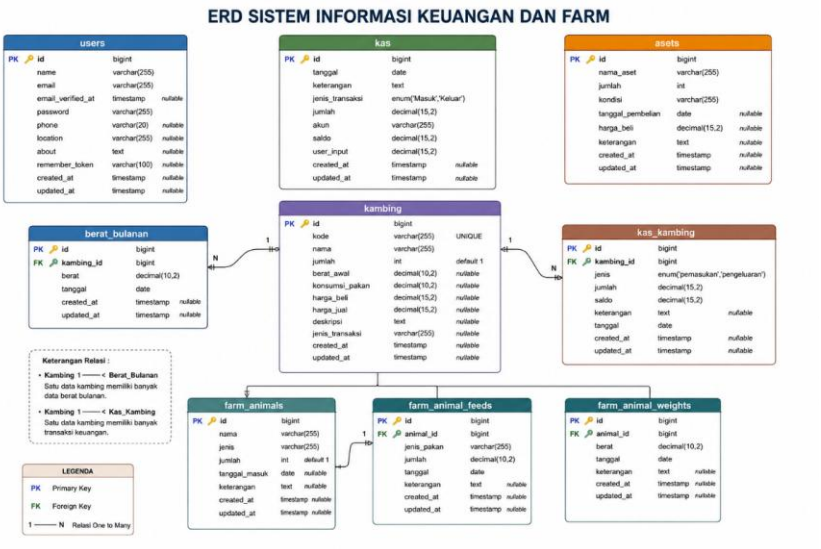
Catatan:
Sistem ini bersifat private. Hanya pengguna yang memiliki username dan password yang telah ditentukan yang dapat mengakses sistem.

Gambar 2. Use Case Diagram Sistem

Entity Relationship Diagram

ERD pada Gambar 3 menggambarkan struktur basis data dan hubungan antar entitas yang terintegrasi untuk mendukung pengelolaan data keuangan dan operasional *farm*. Entitas users menyimpan data pengguna dan berelasi dengan entitas transaksi, yang menyimpan kas masuk dan kas keluar beserta saldo hasil metode *running balance* dan berelasi dengan kategori_transaksi. Entitas saldo_awal menjadi dasar perhitungan saldo berkelanjutan. Pengelolaan aset memakai entitas aset dan kategori_aset. Pada kontrol *farm*, entitas hewan berelasi dengan jenis_hewan serta entitas pendukung pakan, kesehatan_hewan, dan

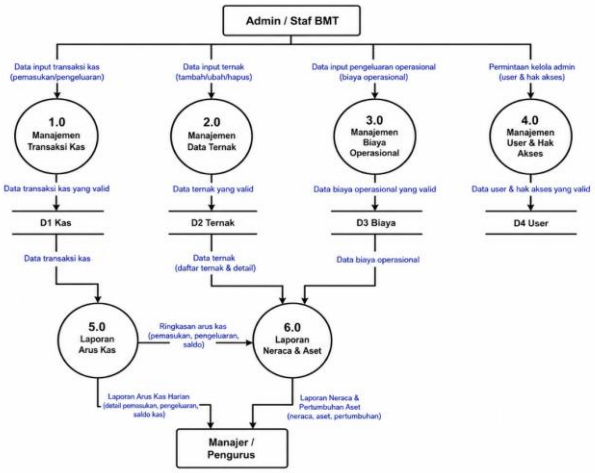
berat_hewan yang mencatat jenis dan stok pakan, riwayat kesehatan, serta perkembangan berat hewan secara berkala.



Gambar 3. Entity Relationship Diagram (ERD)

Data Flow Diagram

DFD pada Gambar 4 menggambarkan aliran data antara pengguna, proses sistem, penyimpanan data, dan laporan. Admin atau staf BMT menginput data transaksi kas, data ternak, biaya operasional, serta pengelolaan user dan hak akses. Data disimpan pada penyimpanan kas, ternak, biaya, dan user, kemudian digunakan sistem untuk menghasilkan laporan arus kas serta laporan neraca dan aset yang disajikan kepada manajer atau pengurus sebagai bahan monitoring dan pengambilan keputusan.



Gambar 4. Data Flow Diagram (DFD)

Flowchart Sistem dan Rancangan Antarmuka

Flowchart sistem menggambarkan alur dari halaman *login*, tempat pengguna memasukkan *username* dan *password* yang divalidasi sistem, hingga *dashboard* yang menampilkan ringkasan dari *database* dan grafik keuangan. Dari *dashboard*, pengguna mengakses menu transaksi kas (sistem langsung menghitung dan menyimpan saldo terbaru dengan *running balance*), laporan neraca (periode dipilih pengguna, dapat dilihat langsung atau diekspor ke PDF/Excel), data aset (dikelola admin), kontrol *farm* (data hewan, pakan, kesehatan, dan berat), hingga *logout*. Rancangan antarmuka dibuat sederhana, intuitif, dan terstruktur dengan menu *dashboard*, transaksi kas, laporan keuangan, data aset, dan kontrol *farm* agar pengguna dapat mengakses fitur dengan cepat.

Implementasi Sistem

Sistem dikembangkan dengan *framework* Laravel (struktur MVC, *routing*, validasi, dan *middleware*) dan *database* MySQL sebagai media penyimpanan data transaksi, saldo, laporan, dan *farm*. Implementasi difokuskan pada pengguna internal, yaitu administrator atau petugas, dan mencakup modul autentikasi, keuangan, kontrol *farm*, laporan, serta algoritma *running balance* sebagai inti perhitungan saldo.

Implementasi Tabel Database

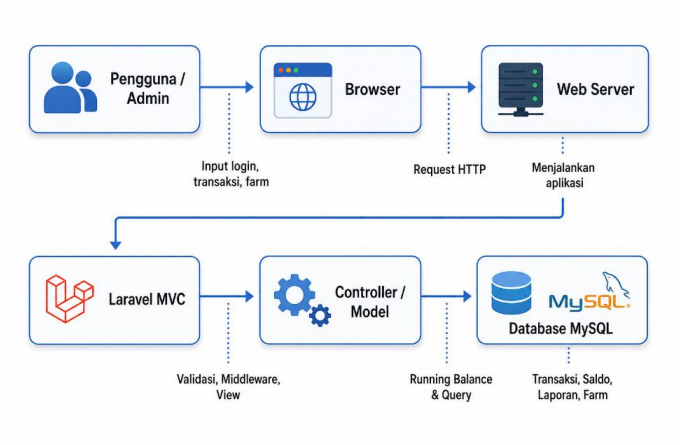
Tabel utama pada *database* ditunjukkan pada Tabel 2. Seluruh tabel menyimpan waktu pembuatan dan pembaruan data (created_at dan updated_at) untuk memudahkan pelacakan perubahan.

Tabel 2. Tabel Database Utama

Tabel	Fungsi
kas	Menyimpan seluruh transaksi keuangan (pemasukan dan pengeluaran): tanggal, jenis transaksi, akun, nominal, keterangan, saldo hasil running balance, serta pengguna yang menginput.
kambing	Menyimpan data master kambing: kode, nama, jumlah, berat awal, konsumsi pakan, harga beli, harga jual, deskripsi, dan jenis transaksi terkait.
kas_kambing	Menyimpan transaksi keuangan kegiatan peternakan kambing (kambing_id, jenis transaksi, nominal, saldo setelah transaksi, keterangan, tanggal) sebagai riwayat keuangan setiap kambing.
farm_animals	Menyimpan data utama hewan ternak: kode, nama, jenis, jenis kelamin, umur, berat terakhir, tanggal masuk, status, kondisi kesehatan, dan foto.
farm_animal_weights	Mencatat perkembangan berat badan hewan secara berkala.
farm_animal_feeds	Mencatat riwayat pemberian pakan hewan.

Arsitektur Sistem

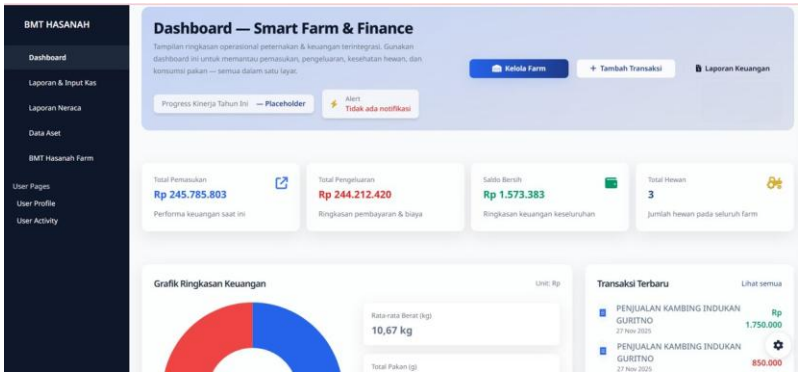
Arsitektur sistem menggunakan pola *client-server* (Gambar 5). Pengguna mengakses sistem melalui browser yang mengirim request HTTP ke *web* server; aplikasi Laravel memproses *routing*, validasi input, autentikasi, dan logika bisnis, serta berkomunikasi dengan *database* MySQL. *Controller* memvalidasi data, memanggil model, dan menjalankan perhitungan *running balance*, sedangkan *middleware* membatasi akses sehingga halaman utama hanya dapat dibuka pengguna yang sudah *login*. Karena data *farm* dan data keuangan berada dalam satu *database*, data yang dikelola lebih konsisten dan monitoring lebih mudah dilakukan.



Gambar 5. Arsitektur Sistem

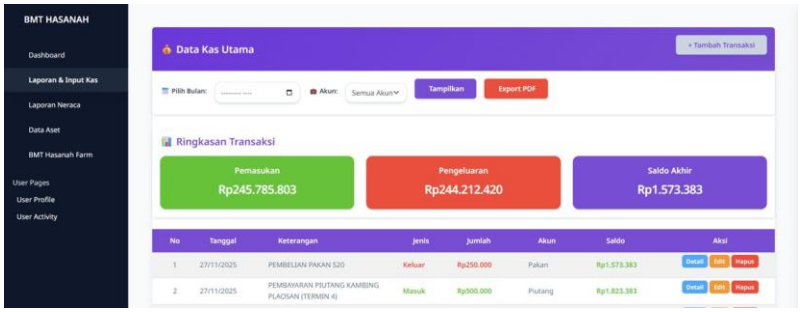
Implementasi Antarmuka

Halaman *login* memastikan hanya pengguna yang memiliki akun dan hak akses yang dapat masuk. Sistem memvalidasi *username* dan *password* melalui *controller* autentikasi; bila sesuai, sistem membuat *session* dan mengarahkan pengguna ke *dashboard*, dan bila tidak sesuai, sistem menampilkan pesan kesalahan dan pengguna tetap berada di halaman *login*. *Dashboard* keuangan (Gambar 6) menampilkan total pemasukan, total pengeluaran, saldo berjalan, dan transaksi terbaru.



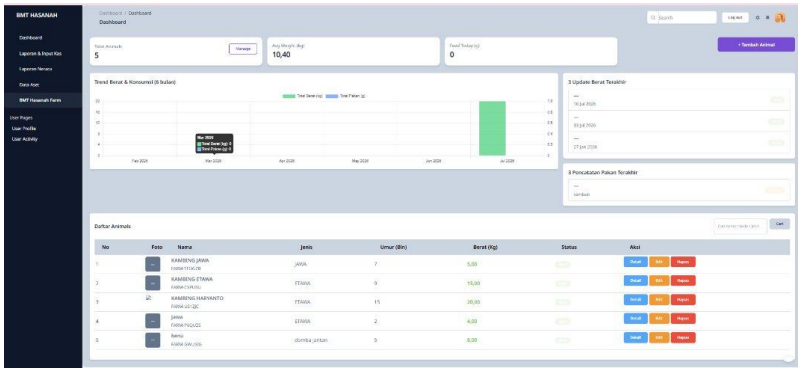
Gambar 6. Halaman Dashboard

Menu input kas (Gambar 7) digunakan untuk mencatat transaksi kas masuk dan kas keluar dengan data tanggal, kategori atau akun, nominal, dan keterangan. Setelah data disimpan, sistem menambah atau mengurangi saldo terakhir sehingga saldo berjalan selalu diperbarui, dengan validasi agar nominal tidak kosong dan sesuai format angka. Riwayat saldo diurutkan berdasarkan tanggal dan ID transaksi karena *running balance* menghitung saldo dari transaksi sebelumnya.



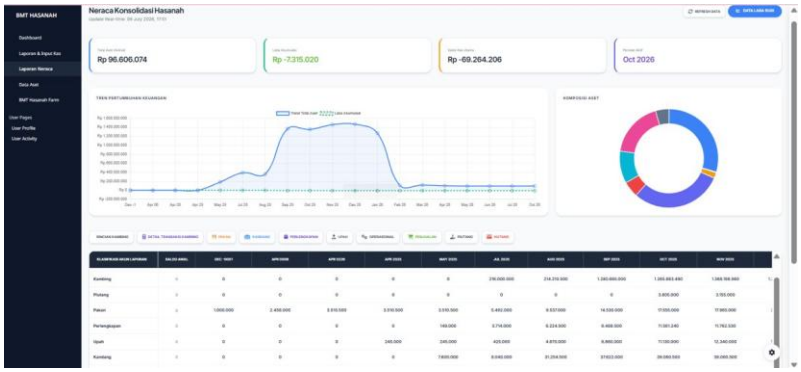
Gambar 7. Halaman Data Kas dan Input Kas

Modul *farm* (Gambar 8) mengelola data kambing (identitas, jumlah, kondisi kesehatan, umur, bobot), kandang, pakan, dan jadwal perawatan, sehingga informasi operasional peternakan tersimpan terpusat dan dapat dikaitkan dengan kebutuhan laporan.



Gambar 8. Halaman Modul Farm

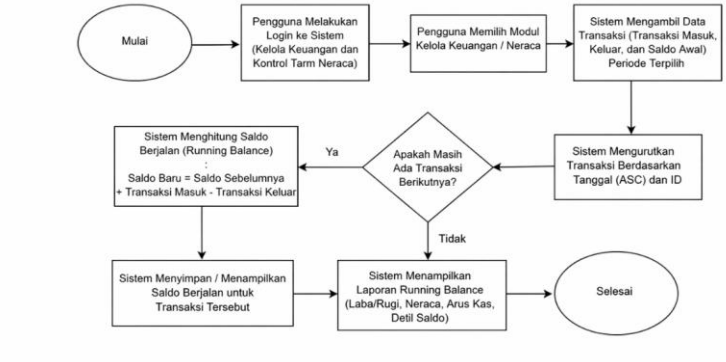
Laporan yang disediakan (Gambar 9) meliputi laporan keuangan, pemasukan, pengeluaran, saldo, neraca, dan *farm*, sehingga pengguna dapat mengevaluasi kondisi keuangan dan operasional dengan lebih mudah.



Gambar 9. Halaman Laporan Neraca

Implementasi Algoritma Running Balance

Alur algoritma *running balance* ditunjukkan pada Gambar 10. Setiap transaksi pemasukan menambah saldo, sedangkan transaksi pengeluaran mengurangi saldo, dan hasilnya langsung disimpan sebagai saldo baru. Berbeda dengan perhitungan manual periodik, saldo tidak perlu dihitung ulang dari awal setiap kali laporan dibuat: sistem cukup mengambil saldo terakhir, membaca jenis transaksi, menghitung saldo baru, lalu menyimpan transaksi beserta saldo barunya. Pada alur lengkapnya, sistem mengambil data transaksi dan saldo awal pada periode terpilih, mengurutkannya berdasarkan tanggal secara menaik dan ID transaksi, menghitung saldo berjalan untuk setiap transaksi hingga seluruh data selesai diproses, kemudian menampilkan laporan laba rugi, neraca, arus kas, dan detail saldo.



Gambar 10. Alur Algoritma Running Balance

Algoritma terdiri atas langkah: ambil saldo terakhir (bernilai 0 bila belum ada transaksi), input jenis dan jumlah transaksi, hitung saldo baru (jika jenis “masuk” maka saldo sebelumnya ditambah jumlah, jika “keluar” maka saldo sebelumnya dikurangi jumlah), lalu simpan transaksi beserta saldo baru. Implementasi pada Laravel/PHP ditunjukkan pada Kode 1.

Kode 1. Implementasi Running Balance pada Controller Laravel

```
public function store(Request $request)
{
      $lastTransaction = Transaksi::latest()->first();
      $saldo_sebelumnya = $lastTransaction ? $lastTransaction->saldo : 0;
      $jumlah = $request->jumlah;
      $jenis  = $request->jenis_transaksi;
      if ($jenis == 'masuk') {
            $saldo_baru = $saldo_sebelumnya + $jumlah;
      } else {
            $saldo_baru = $saldo_sebelumnya - $jumlah;
      }
      Transaksi::create([
            'tanggal' => $request->tanggal,
            'jenis_transaksi' => $jenis,
            'jumlah' => $jumlah,
            'saldo' => $saldo_baru
      ]);
}
```

Pada proses update dan delete, sistem menjaga konsistensi saldo melalui fungsi recalculateBalance, yang mengurutkan transaksi berdasarkan tanggal dan ID, lalu menghitung ulang saldo lama dan saldo baru setiap transaksi setelahnya, karena *running balance* bergantung pada urutan kronologis transaksi. Sebagai contoh, transaksi masuk Rp1.000.000 (1 Jan) menghasilkan saldo Rp1.000.000, transaksi keluar Rp200.000 (2 Jan) menghasilkan saldo Rp800.000, dan transaksi masuk Rp500.000 (3 Jan) menghasilkan saldo Rp1.300.000. Dengan mekanisme ini, setiap perubahan transaksi langsung memengaruhi saldo secara otomatis sehingga meningkatkan efisiensi dan transparansi serta mengurangi kesalahan pencatatan keuangan.

HASIL PENGUJIAN

Pengujian Black Box

Pengujian *black box* dilakukan pada setiap fitur utama berdasarkan input dan output yang diharapkan (Tabel 3). Seluruh fitur utama berjalan sesuai output yang diharapkan, sehingga fungsi *login*, transaksi, *farm*, laporan, dan *logout* dapat digunakan sesuai kebutuhan sistem.

Tabel 3. Hasil Pengujian Black Box

No	Test Case	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Status
1	*Login* Admin	Menginput *username* dan *password* valid	*Username* dan *password* benar	Sistem berhasil masuk ke *dashboard* admin	Berhasil
2	*Login* Gagal	Menginput data *login* salah	Data *login* salah	Sistem menampilkan pesan kesalahan dan tetap pada halaman *login*	Berhasil
3	Menampilkan *Dashboard*	Mengakses halaman utama setelah *login*	-	*Dashboard* menampilkan menu keuangan, *farm*, laporan, *logout*	Berhasil
4	Input Transaksi Masuk	Menambahkan data kas masuk	Nominal, tanggal, keterangan	Data tersimpan dan saldo bertambah (*running balance* update)	Berhasil
5	Input Transaksi Keluar	Menambahkan data kas keluar	Data kas keluar	Data tersimpan dan saldo berkurang otomatis	Berhasil
6	*Running Balance*	Menghitung saldo otomatis	Data transaksi masuk & keluar	Saldo akhir sesuai perhitungan (saldo sebelumnya + debit − kredit)	Berhasil
7	Menampilkan Laporan Neraca	Membuka halaman laporan	-	Sistem menampilkan aset, liabilitas, dan ekuitas secara otomatis	Berhasil
8	Filter Laporan	Mencari laporan berdasarkan tanggal	Rentang tanggal	Data laporan sesuai filter yang dipilih	Berhasil
9	Input Data Kambing	Menambahkan data ternak	Nama, jumlah, kondisi	Data kambing tersimpan dalam sistem	Berhasil

No	Test Case	Skenario Pengujian	Data Masukan	Hasil yang Diharapkan	Status
10	Update Data Kambing	Mengubah kondisi kambing	Status kesehatan / jumlah	Data berhasil diperbarui	Berhasil
11	Hapus Data	Menghapus data transaksi/kambing	ID data	Data berhasil dihapus dari sistem	Berhasil
12	*Logout*	Keluar dari sistem	-	Sistem kembali ke halaman *login*	Berhasil

Pengujian Akurasi Running Balance

Pengujian akurasi membandingkan hasil perhitungan manual dan sistem pada 10 transaksi untuk memastikan saldo yang dihasilkan sistem sama dengan perhitungan manual (Tabel 4). Contohnya, dengan saldo awal Rp5.000.000, pemasukan Rp2.000.000 menghasilkan saldo Rp7.000.000, dan pengeluaran Rp1.000.000 menghasilkan saldo Rp6.000.000. Seluruh hasil perhitungan sistem sama dengan perhitungan manual sehingga tingkat kesesuaian *running balance* adalah 100%.

Tabel 4. Pengujian Akurasi Running Balance

No	Saldo Awal	Jenis	Nominal (Rp)	Saldo Sistem (Rp)	Saldo Manual (Rp)	Status
1	Rp0	Pemasukan	5.000.000	5.000.000	5.000.000	Sesuai
2	Rp5.000.000	Pemasukan	2.000.000	7.000.000	7.000.000	Sesuai
3	Rp7.000.000	Pengeluaran	1.000.000	6.000.000	6.000.000	Sesuai
4	Rp6.000.000	Pemasukan	1.500.000	7.500.000	7.500.000	Sesuai
5	Rp7.500.000	Pengeluaran	750.000	6.750.000	6.750.000	Sesuai
6	Rp6.750.000	Pengeluaran	250.000	6.500.000	6.500.000	Sesuai
7	Rp6.500.000	Pemasukan	3.000.000	9.500.000	9.500.000	Sesuai

No	Saldo Awal	Jenis	Nominal (Rp)	Saldo Sistem (Rp)	Saldo Manual (Rp)	Status
8	Rp9.500.000	Pengeluaran	2.000.000	7.500.000	7.500.000	Sesuai
9	Rp7.500.000	Pemasukan	500.000	8.000.000	8.000.000	Sesuai
10	Rp8.000.000	Pengeluaran	1.250.000	6.750.000	6.750.000	Sesuai

Pengujian Konsistensi Data

Pengujian konsistensi membandingkan data pada *dashboard*, *database*, dan laporan (Tabel 5). Ketiganya menunjukkan nilai yang sama, sehingga sistem mampu menjaga konsistensi data antar modul karena informasi berasal dari sumber data yang sama.

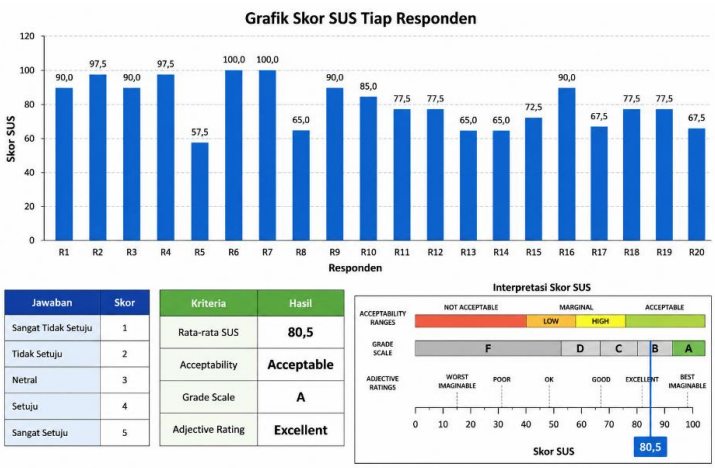
Tabel 5. Pengujian Konsistensi Data

Data	*Dashboard*	*Database*	Laporan	Status
Total pemasukan	Rp12.000.000	Rp12.000.000	Rp12.000.000	Sesuai
Total pengeluaran	Rp5.250.000	Rp5.250.000	Rp5.250.000	Sesuai
Saldo akhir	Rp6.750.000	Rp6.750.000	Rp6.750.000	Sesuai
Jumlah transaksi	10	10	10	Sesuai
Data kambing	25 ekor	25 ekor	25 ekor	Sesuai

Pengujian System Usability Scale (SUS)

Pengujian SUS dilakukan kepada 20 responden dengan komposisi pengelola/administrator 4 orang (20%), staf administrasi 6 orang (30%), dan pengguna/petugas *farm* 10 orang (50%). Komposisi ini menunjukkan bahwa sebagian besar responden berinteraksi langsung dengan operasional sistem.

Pada pernyataan positif, responden memberikan rata-rata nilai 4,30–4,50, yang menunjukkan sistem dinilai mudah digunakan, terintegrasi dengan baik, serta cepat dipelajari. Pernyataan negatif memperoleh rata-rata rendah, yaitu 1,70–1,90, yang mengindikasikan responden tidak merasa sistem terlalu rumit, tidak membutuhkan banyak bantuan teknis, dan tidak mengalami kesulitan berarti, termasuk pada proses *running balance*. Sebagai contoh perhitungan, responden pertama (R1) menjawab Q1 = 4, Q2 = 2, Q3 = 5, Q4 = 2, Q5 = 5, Q6 = 1, Q7 = 4, Q8 = 1, Q9 = 5, dan Q10 = 1, sehingga total skor item ganjil adalah 18 dan item genap 18, dan skor SUS R1 = (18 + 18) × 2,5 = 90. Perhitungan yang sama dilakukan untuk seluruh responden (Gambar 11); jumlah skor 1.610 dibagi 20 responden menghasilkan rata-rata 80,5.



Gambar 11. Skor SUS Tiap Responden dan Interpretasi

Nilai rata-rata 80,5 berada di atas nilai standar SUS yaitu 68, sehingga sistem dinyatakan layak digunakan (Acceptability: *Acceptable*). Berdasarkan interpretasi SUS pada Gambar 11, skor tersebut termasuk kategori Grade B dengan *Adjective Rating Excellent*, yang menunjukkan bahwa sistem mudah digunakan, mudah dipelajari, dan dapat diterima dengan baik oleh pengguna di BMT Hasanah Ponorogo.

PEMBAHASAN

Sistem informasi laporan neraca keuangan dan kontrol *farm* berhasil dibangun sesuai kebutuhan yang dirancang. Sistem menyediakan fitur *login*, *dashboard*, pengelolaan transaksi pemasukan dan pengeluaran, riwayat saldo, laporan keuangan, laporan *farm*, dan pengelolaan data *farm* dalam satu aplikasi berbasis *web*. Algoritma *running balance* menjadi bagian utama karena setiap transaksi langsung memengaruhi saldo: pemasukan ditambahkan ke saldo sebelumnya dan pengeluaran dikurangkan, sehingga saldo selalu diperbarui secara *real-time* dan risiko kesalahan perhitungan manual berkurang.

Pengujian *black box* menunjukkan seluruh fungsi utama berjalan sesuai kebutuhan, mulai dari pembatasan akses melalui *login*, penyimpanan transaksi dan pembaruan saldo, pengelolaan data *farm*, hingga penyajian laporan sesuai periode. Kesesuaian 100% pada 10 transaksi menunjukkan algoritma menghasilkan saldo yang akurat selama urutan transaksi dan nominal yang dimasukkan sesuai, dan uji konsistensi menunjukkan informasi tidak berbeda antar halaman karena bersumber dari data yang sama. Skor SUS 80,5 (Grade B, *Excellent*) didukung tampilan yang sederhana, menu yang jelas, dan proses input yang terstruktur.

Dibandingkan penelitian terdahulu yang umumnya berfokus pada pencatatan transaksi dan penyusunan laporan keuangan, sistem ini memiliki perbedaan pada penerapan metode *running balance* dan integrasi dengan modul kontrol *farm*. Kelebihan sistem adalah kemampuannya mengelola data keuangan secara *real-time*, menghitung saldo otomatis, menyajikan laporan, dan mengintegrasikan data *farm*, serta akses berbasis *web* yang mudah

selama pengguna memiliki perangkat dan jaringan yang mendukung. Keterbatasannya, sistem belum terintegrasi dengan sistem perbankan, belum menggunakan perangkat IoT untuk monitoring *farm* otomatis, akses masih difokuskan pada pengguna internal, dan masih memerlukan pengembangan untuk level hak akses yang lebih beragam serta laporan keuangan yang lebih kompleks. Secara keseluruhan, sistem telah menjawab rumusan masalah dan tujuan penelitian, yaitu merancang sistem informasi laporan keuangan berbasis *web*, menerapkan *running balance* untuk menghitung saldo otomatis, mengintegrasikan data keuangan dengan data *farm*, dan meningkatkan kemudahan monitoring keuangan serta kondisi peternakan pada BMT Hasanah Ponorogo.

PENUTUP

Kesimpulan

Berdasarkan hasil penelitian, dapat disimpulkan bahwa: (1) sistem informasi laporan keuangan berbasis *web* pada BMT Hasanah Ponorogo berhasil dirancang dan dibangun sehingga mampu mendukung pengelolaan data keuangan dan kontrol *farm* dalam satu sistem; (2) metode *running balance* berhasil diimplementasikan sehingga sistem dapat menghitung dan memperbarui saldo secara otomatis, *real-time*, dan akurat pada setiap transaksi; (3) data keuangan dan data operasional peternakan berhasil diintegrasikan ke dalam satu basis data sehingga memudahkan pengelolaan dan monitoring secara terpusat; dan (4) sistem mampu meningkatkan efisiensi serta kemudahan monitoring keuangan dan kondisi peternakan karena pencatatan, pengolahan data, dan penyajian laporan dilakukan secara terkomputerisasi. Dengan demikian, seluruh rumusan masalah dan tujuan penelitian telah tercapai.

Saran

Sistem dapat dikembangkan lebih lanjut dengan (1) menambahkan integrasi dengan layanan perbankan atau pembayaran digital, (2) mengembangkan modul kontrol *farm* menggunakan teknologi Internet of Things (IoT), (3) melengkapi sistem dengan pengaturan hak akses pengguna yang lebih rinci, dan (4) mengembangkan fitur laporan keuangan menjadi lebih lengkap sesuai kebutuhan BMT Hasanah Ponorogo secara lebih mendalam guna mendukung pengambilan keputusan manajerial yang lebih komprehensif.

DAFTAR PUSTAKA

Amalia, F., Kurniawan, M., & Setiyawan, D. T. (2021). Pengujian Blackbox Pada Desain Antarmuka Sistem Informasi Traceability Rantai Pasok Apel. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIIK), 8*(5), 853–860. Https://Doi.Org/10.25126/Jtiik.202183487

Anggraeni, D. P. (2023). Pengembangan Sistem Informasi Laporan Keuangan Digital Berbasis Web Untuk Efisiensi Manajemen Keuangan. *Jurnal Penelitian Sistem Informasi, 1*(1), 17–22. Https://Doi.Org/10.54066/Jpsi.V2i4.2629

Brooke, J. (1996). *SUS: A Quick And Dirty Usability Scale*. Taylor & Francis.

Budiningrum, E. W., & Subiyantoro, A. (2023). Sistem Informasi Akuntansi Koperasi Untuk Meningkatkan Daya Saing Pengelolaan Koperasi. *Jurnal Economina, 2*(3), 25–37.

Fadliana, N., & Ahyana, N. (2025). Perancangan Sistem Informasi Koperasi Online Berbasis Web Di Lingkungan Kampus. *Jurnal Minfo Polgan, 14*(2), 3643–3649. Https://Doi.Org/10.33395/Jmp.V14i2.15948

Fauzan, H., & Ningsih, E. (2022). Analisis Dan Perancangan Sistem Informasi Akuntansi Simpan Pinjam Koperasi Berbasis Web. *Jurnal Pustaka AI, 2*(3), 230–245.

Fitri. (2021). Pembuatan Sistem Informasi Inventaris Politeknik Kampar Berbasis Web Menggunakan PHP Dan Mysql Dengan Framework Codeigniter. *Jurnal Sains Dan Ilmu Terapan, 4*(2), 156–170.

Ibrahim, R., & Yen, S. Y. (2010). Formalization Of The Data Flow Diagram Rules For Consistency Check. *International Journal Of Software Engineering & Applications, 1*(4), 95–111. Https://Doi.Org/10.5121/Ijsea.2010.1406

Kurniawan, T. A. (2018). Pemodelan Use Case (UML): Evaluasi Terhadap Beberapa Kesalahan Dalam Praktik. *Jurnal Teknologi Informasi Dan Ilmu Komputer, 5*(1), 77–86. Https://Doi.Org/10.25126/Jtiik.201851610

Prabowo, M., & Suprapto, A. (2021). Usability Testing Pada Sistem Informasi Akademik IAIN Salatiga Menggunakan Metode System Usability Scale. *JANUARI, 6*(1), 38–49.

Prawastiyo, C. A., & Hermawan, I. (2020). Pengembangan Front-End Website Perpustakaan Politeknik Negeri Jakarta Dengan Menggunakan Metode User Centered Design. *Jurnal Teknologi Terpadu, 6*(4), 89–95.

Purnaningsih, P. K., & Yulianto, A. (2022). Perancangan Sistem Informasi Manajemen Berbasis Web Dalam Pengelolaan Data Siswa. *Remik, 6*(4), 738–753. Https://Doi.Org/10.33395/Remik.V6i4.11818

Rahmalia, H. L., & Komariyah, F. (2022). Analisis Sistem Pencatatan Laporan Keuangan Manual. *Revenue: Jurnal Ilmiah Ilmu Akuntansi, 3*(1). Https://Doi.Org/10.46306/Rev.V3i1

Riyanto, S., Rahmawati, E., Hertyana, H., & Mufida, E. (2023). Sistem Informasi Simpan Pinjam Berbasis Web Pada Koperasi Jasa Kawan Sejahtera Fakultas Teknik Universitas Pancasila. *Jurnal Asiimetrik: Jurnal Ilmiah Rekayasa Dan Inovasi, 5*(3), 63–70.

Septian, Y., Kurniawan, K., Damayanti, N. R., & Fatmasari, F. (2023). Pengembangan Sistem Informasi Teritorial Berbasis Web Menggunakan Framework Laravel Untuk Pemantauan Sumber Daya Di Korem 044/Gapo. *Jurnal Teknologi Informatika Dan Komputer, 9*(2), 1118–1135. Https://Doi.Org/10.37012/Jtik.V9i2.1809

Setiawan, Y., Fitri, I., & Sholihatih, I. D. (2022). Electronic Customer Relationship Management Menggunakan Framework Codeigniter Dengan Metode Waterfall Berbasis Website. *Jurnal Teknologi Informasi Dan Komunikasi, 6*(2).

Sugiartawan, P., Suryawan, I. G. T., & Indawan, I. G. A. (2023). Sistem Informasi Keuangan Pada Koperasi Karya Utama Mandiri. *Jurnal Sistem Informasi Dan Komputer Terapan Indonesia (JSIKTI), 5*(2), 97–108. Https://Doi.Org/10.33173/Jsikti.181

Suwandhi, A., Lumbanbatu, M. N., & Ibbi, U. (2025). Perancangan Sistem Informasi Keuangan Koperasi Simpan Pinjam Cipta Maju Berbasis Web. *Jurnal Publikasi Pengabdian Kepada Masyarakat, 5*(1).

Trissiyan, N. P. P., & Suwandi. (2024). Pengaruh Penerapan Sistem Informasi Akuntansi Dan Good Corporate Governance Terhadap Kualitas Laporan Keuangan. *Neraca: Jurnal Ekonomi, Manajemen Dan Akuntansi, 2*(5), 346–351.

Usnaini, M., Yasin, V., & Sianipar, A. Z. (2021). Perancangan Sistem Informasi Inventarisasi Aset Berbasis Web Menggunakan Metode Waterfall. *Jurnal Manajemen Informatika Jayakarta, 1*(1), 36. Https://Doi.Org/10.52362/Jmijayakarta.V1i1.415

Widianto, E., & Kurniadi, D. (2021). Rancang Bangun Sistem Informasi Manajemen Keuangan RT/RW Berbasis Web. *Jurnal Algoritma, 18*(3), 246–253. Https://Doi.Org/10.33364/Algoritma/V.18-1.838

Wulandari, A. S., Saepudin, A., Kinanti, M. P., Sudesi, Z., Saifudin, A., & Yulianti, Y. (2022). Pengujian Aplikasi Sistem Informasi Akademik Berbasis Web Menggunakan Metode Black Box Testing Equivalence Partitioning. *Jurnal Teknologi Sistem Informasi Dan Aplikasi, 5*(2), 102. Https://Doi.Org/10.32493/Jtsi.V5i2.17561