

Perancangan Sistem Informasi Penjualan dan Manajemen Stok Pada Toko Tingwe Tembakau Menggunakan Metode Agile Modeling

Sales and Inventory Information System at Tingwe Tobacco Using Agile Modeling

Ananda Sabitha Khairani Putri¹, Muhammad Aflah Masdarul Makarim²,
Muhammad Ilham Arif³, Mohamad Yasin⁴

^{1,2,3}Universitas Negeri Malang

³Program Studi S1 Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Malang

*e-mail: ananda.sabitha.2303126@students.um.ac.id¹, muhammad.aflah.2303126@students.um.ac.id

²muhammad.ilham.2303126@students.um.ac.id³ mohamad.yasin.fmipa@um.ac.id⁴

Received:
19.05.2026

Revised:
17.08.2026

Accepted:
21.09.2026

Available online:
06.10.2026

Abstrak: Perkembangan teknologi informasi mendorong sektor UMKM beralih dari sistem manual menuju sistem digital. Toko Tingwe Tembakau saat ini masih mengandalkan pencatatan manual dalam operasional penjualan dan manajemen persediaan, yang rentan terhadap kesalahan, kehilangan data stok, dan keterlambatan evaluasi keuangan. Penelitian ini bertujuan merancang dan mengimplementasikan Sistem Informasi Penjualan dan Manajemen Stok berbasis web menggunakan metode Agile Modeling. Pendekatan iteratif ini berpusat pada pengguna untuk menghasilkan sistem yang fungsional dan adaptif. Sistem dibangun dengan framework Laravel 12 dan MySQL, serta mengintegrasikan fitur inovatif berupa perhitungan Zakat Mal harian berdasarkan keuntungan usaha. Pengujian sistem meliputi White Box Testing menggunakan SonarQube, Black Box Testing, dan Usability Testing dengan kuesioner System Usability Scale (SUS). Hasil pengujian menunjukkan kualitas kode sangat baik dengan predikat A pada aspek keamanan dan keterawatan, tingkat keberhasilan fungsionalitas mencapai 100% dari 86 test case, dan tingkat kemudahan penggunaan memperoleh skor rata-rata 72,67 (kategori Good). Sistem ini berhasil mendigitalisasi operasional toko secara efisien dan akurat, serta memberikan nilai tambah melalui penggabungan operasional komersial dengan aspek sosial-religius.

Kata Kunci: Sistem Informasi, Penjualan, Manajemen Stok, Agile Modeling.

Abstract: Information technology advancements encourage the MSME sector to transition from manual to digital systems. Currently, Tingwe Tobacco Shop relies on manual recording for sales operations and inventory management, which is prone to errors, unmonitored stock depletion, and delayed financial evaluations. This study aims to design and implement a web-based Sales and Inventory Management Information System using the Agile Modeling method. This iterative approach is user-centered to produce a functional and adaptive system. The system was developed using the Laravel 12 framework and MySQL, integrating a specialized feature for daily Zakat Mal calculation based on business profits. System testing included White Box Testing using SonarQube, Black Box Testing, and Usability Testing using the System Usability Scale (SUS) questionnaire. The evaluation results demonstrated excellent code quality, achieving a grade A in security and maintainability aspects, a 100% functional success rate across 86 test cases, and an average SUS score of 72.67 (Good category) for ease of use. This system successfully digitalizes the store's operations efficiently and accurately, providing an innovative integration of commercial operations with socio-religious aspects.

Keywords: Information System, Sales, Inventory Management, Agile Modeling

1. PENDAHULUAN

Perkembangan teknologi informasi telah mendorong berbagai sektor Usaha Mikro, Kecil, dan Menengah (UMKM) untuk beralih dari sistem manual menuju sistem digital. Teknologi informasi berperan penting dalam mendukung berbagai aktivitas manusia, termasuk kegiatan bisnis, melalui penyediaan akses terhadap informasi yang lebih cepat, akurat, dan terintegrasi [4]. Kebutuhan terhadap sistem informasi yang terintegrasi juga dirasakan oleh sektor ritel, termasuk Toko Tingwe Tembakau. Toko tersebut merupakan usaha ritel yang bergerak dalam penjualan produk tembakau dan perlengkapan melinting, seperti tembakau kemasan atau onsan, kertas rokok, filter atau gabus, alat linting, cengkeh, semprot rasa, cerutu, dan tepak tembakau. Saat ini, seluruh proses operasional dan transaksi penjualan di Toko Tingwe Tembakau masih dilakukan secara langsung (*face-to-face*) tanpa dukungan sistem komputerisasi.

Dalam praktiknya, pencatatan transaksi penjualan sehari-hari di Toko Tingwe masih mengandalkan buku tulis secara manual. Metode konvensional ini rentan terhadap kesalahan manusia (*human error*), terutama saat penjual sedang terburu-buru sehingga lupa mencatat atau salah menuliskan barang dan harga yang terjual. Selain itu, pengelolaan persediaan juga masih dilakukan dengan pengecekan fisik tanpa adanya pencatatan stok minimum maupun notifikasi otomatis. Akibatnya, toko sering mengalami kehabisan barang tanpa disadari. Kondisi serupa juga dilaporkan pada beberapa UMKM ritel yang masih menggunakan pencatatan manual, di mana kesalahan input data dan ketidaktertiban arsip menjadi penyebab utama hilangnya informasi penjualan dan stok. Untuk mengatasi hal ini, diperlukan transisi menuju digitalisasi. Sistem informasi penjualan merupakan serangkaian prosedur yang dirancang secara khusus untuk menjalankan, merekam, menghitung, menghasilkan dokumen, dan menyediakan informasi terkait penjualan [2].

Di sisi lain, proses manajerial keuangan juga belum terkelola dengan optimal. Perhitungan keuntungan harian dilakukan secara manual dengan membandingkan harga beli produk dari pemasok dengan harga jual kepada pelanggan. Meskipun rekapitulasi penjualan diupayakan hampir setiap hari, format pencatatan yang masih berupa buku catatan ini sangat menyulitkan pemilik dalam melakukan evaluasi keuangan bulanan secara cepat. Selain itu, pemilik Toko Tingwe memiliki visi sosial-religius yang membutuhkan dukungan komputasi khusus; beliau ingin agar laporan tidak hanya menampilkan keuntungan finansial semata, tetapi juga memuat perhitungan zakat mal harian yang langsung diakumulasikan ke total bulanan, sehingga proses berbisnis tidak hanya sekedar mengumpulkan harta, namun juga mensucikannya. Kebutuhan integrasi fitur zakat ke dalam sistem informasi bisnis sejalan dengan tren pengembangan sistem zakat digital berbasis web dengan pendekatan Agile untuk meningkatkan akurasi perhitungan [24].

Kompleksitas permasalahan dan kebutuhan spesifik pengguna tersebut menunjukkan perlunya pengembangan sistem digital yang adaptif. Dalam dunia rekayasa perangkat lunak, metode Agile telah menarik perhatian luas sebagai pendekatan manajemen proyek yang mampu menangani tantangan unik dalam pengembangan sistem informasi [1]. Oleh karena itu, Sistem Informasi Penjualan dan Manajemen Stok pada Toko Tingwe Tembakau akan dirancang dan dikembangkan menggunakan pendekatan Agile Modeling. Penggunaan metode ini sejalan dengan tujuan utama sistem, yaitu beralih menuju operasional digital untuk meminimalkan kesalahan manusia, mengelola pembaruan stok secara otomatis, menampilkan peringatan stok minimum, serta menghasilkan laporan laba dan zakat secara instan.

2. TINJAUAN PUSTAKA

2.1. *Pengertian Agile Development*

Agile Software Development merupakan pendekatan pengembangan perangkat lunak yang menekankan proses iteratif, penyampaian perangkat lunak secara bertahap, dan kemampuan beradaptasi terhadap perubahan kebutuhan selama proses pengembangan. Kajian sistematis terhadap penelitian empiris menunjukkan bahwa penelitian Agile mencakup tema pengenalan dan adopsi, faktor manusia dan sosial, persepsi terhadap metode Agile, serta perbandingan metode pengembangan [11]. Perkembangan penelitian selama satu dekade juga menunjukkan bahwa Agile Software Development mencakup beragam metodologi dan praktik, sehingga tidak dapat dipahami sebagai satu metode tunggal [12]. Dalam penerapannya, proses iteratif memungkinkan kebutuhan dan solusi dikembangkan secara bertahap melalui evaluasi serta umpan balik selama pengembangan [13]. Keterlibatan pelanggan atau pengguna juga berperan dalam meningkatkan kemampuan proyek untuk merespons kebutuhan yang berubah dan mendukung keberhasilan pengembangan sistem [14].

Agile Modeling merupakan pendekatan pemodelan yang digunakan dalam konteks Agile Software Development. Pendekatan ini tidak menggantikan keseluruhan metodologi pengembangan perangkat lunak, tetapi menyediakan praktik untuk membuat, menggunakan, dan memperbarui model sesuai kebutuhan pengembangan. Pemodelan dilakukan secukupnya untuk membantu memahami kebutuhan, menganalisis permasalahan, merancang solusi, dan mengomunikasikan keputusan teknis tanpa mengharuskan seluruh model sistem diselesaikan secara

rinci sejak awal. Model dapat disesuaikan atau dikembangkan kembali ketika kebutuhan dan pemahaman terhadap sistem mengalami perubahan [15]. Penelitian mengenai Agile Modeling masih didominasi oleh studi kasus, analisis komparatif, dan laporan pengalaman, sedangkan penelitian empiris yang secara khusus menguji Agile Modeling masih terbatas [16].

Dalam penelitian ini, Agile Modeling digunakan untuk mendukung perancangan dan pengembangan Sistem Informasi Penjualan dan Manajemen Stok pada Toko Tingwe Tembakau. Pendekatan tersebut relevan karena rancangan sistem perlu disusun berdasarkan kebutuhan pengguna dan dapat dikembangkan secara bertahap. Keterlibatan pengguna dalam proses Agile dapat membantu mengarahkan pengembangan sistem agar tetap responsif terhadap kebutuhan yang teridentifikasi selama proses pengembangan [14]. Oleh karena itu, pemodelan sistem digunakan sebagai sarana untuk memahami kebutuhan operasional toko, mengomunikasikan rancangan, dan melakukan penyesuaian terhadap fitur yang dikembangkan. Penerapan ini diarahkan pada perancangan fitur pencatatan transaksi, pembaruan stok, peringatan stok minimum, serta penyajian laporan laba dan zakat.

2.2. Nilai-Nilai Agile (Agile Values)

Dalam Agile Modeling terdapat empat nilai utama yang menjadi dasar dalam proses pengembangan sistem. Nilai-nilai ini dikembangkan oleh Kent Beck dalam konsep Extreme Programming (XP). Empat nilai utama tersebut meliputi:

2.2.1 Communication

Komunikasi merupakan faktor penting dalam pengembangan sistem. Komunikasi yang baik antara pengembang sistem dan pengguna akan membantu mengurangi kesalahan dalam memahami kebutuhan sistem.

2.2.2 Simplicity

Simplicity berarti pengembangan sistem harus dimulai dari solusi yang paling sederhana terlebih dahulu. Sistem tidak perlu langsung dibuat kompleks, tetapi dikembangkan secara bertahap sesuai kebutuhan.

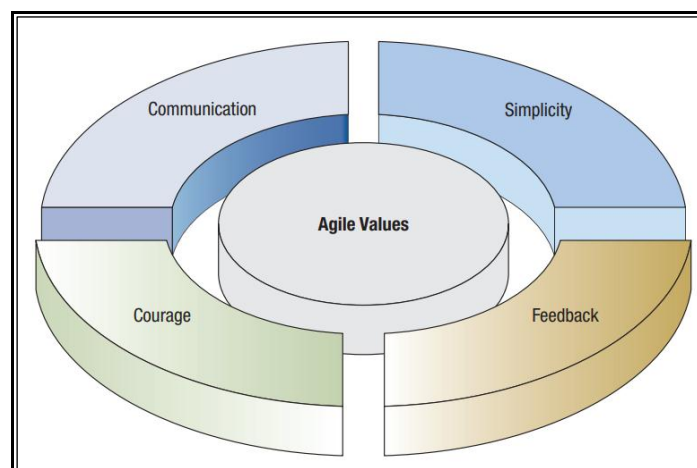
2.2.3 Feedback

Feedback merupakan umpan balik yang diberikan oleh pengguna maupun tim pengembang terhadap sistem yang sedang dikembangkan. Umpan balik ini digunakan untuk melakukan perbaikan dan penyempurnaan sistem.

2.2.4 Courage

Courage berarti tim pengembang harus memiliki keberanian untuk melakukan perubahan apabila terdapat kesalahan atau solusi yang kurang tepat dalam sistem yang dikembangkan.

Keempat nilai tersebut menjadi dasar dalam menciptakan proses pengembangan sistem yang fleksibel dan responsif terhadap perubahan kebutuhan pengguna.



Gambar 1. Values are crucial to the agile approach

2.3. Aktivitas dalam Agile Development

Dalam pendekatan Agile terdapat empat aktivitas utama yang dilakukan dalam proses pengembangan sistem, yaitu:

2.3.1. Coding

Coding merupakan proses penulisan kode program untuk membangun sistem. Aktivitas ini menjadi inti dari pengembangan perangkat lunak karena melalui kode program ide dan rancangan sistem dapat diwujudkan.

2.3.2. Testing

Testing merupakan proses pengujian sistem untuk memastikan bahwa sistem berjalan sesuai dengan fungsi yang diharapkan. Pengujian dilakukan untuk mengidentifikasi kesalahan atau bug dalam sistem.

2.3.3. Listening

Listening merupakan proses mendengarkan kebutuhan dan masukan dari pengguna. Dalam Agile, komunikasi dengan pengguna sangat penting agar sistem yang dikembangkan benar-benar sesuai dengan kebutuhan pengguna.

2.3.4. Designing

Designing merupakan proses perancangan struktur sistem yang meliputi desain logika sistem, antarmuka pengguna (user interface), serta arsitektur sistem secara keseluruhan.

2.4. Variabel Sumber Daya dalam Agile

Dalam proses pengembangan sistem menggunakan Agile terdapat empat variabel sumber daya yang harus dikelola dengan baik, yaitu:

1. Time: waktu yang diperlukan untuk menyelesaikan proyek.
2. Cost: biaya yang digunakan dalam proses pengembangan sistem.
3. Quality: kualitas sistem yang dihasilkan.
4. Scope: ruang lingkup fitur atau fungsi yang akan dikembangkan.

Keempat variabel tersebut saling berkaitan dan harus dikelola dengan baik agar proyek pengembangan sistem dapat diselesaikan secara efektif.

2.5. Praktik Inti Agile (Core Practices)

Pendekatan Agile memiliki beberapa praktik inti yang membedakannya dengan metode pengembangan sistem lainnya, yaitu:

1. Short Releases

Sistem dikembangkan dan di rilis dalam waktu yang lebih singkat sehingga pengguna dapat segera menggunakan versi awal sistem.

2. 40-Hour Workweek

Tim pengembang bekerja secara intensif dalam waktu kerja yang normal untuk menjaga produktivitas dan kualitas kerja.

3. On-site Customer

Pengguna atau perwakilan pengguna terlibat langsung dalam proses pengembangan sistem sehingga kebutuhan sistem dapat dipahami dengan lebih baik.

4. Pair Programming

Dua programmer bekerja bersama dalam proses penulisan kode untuk meningkatkan kualitas program serta mengurangi kesalahan dalam pengembangan sistem.

2.6. Proses Pengembangan Agile

Secara umum, proses pengembangan sistem menggunakan metode Agile dilakukan melalui beberapa langkah berikut:

1. Mengidentifikasi user stories dari pengguna untuk memahami kebutuhan sistem.
2. Membuat model alur kerja logis untuk menggambarkan proses bisnis yang ada.
3. Mengembangkan user stories tambahan berdasarkan model yang telah dibuat.

4. Membuat prototype tampilan sistem untuk menunjukkan antarmuka yang akan digunakan pengguna.
5. Menggunakan umpan balik dari pengguna untuk menyempurnakan sistem hingga menghasilkan model sistem yang lengkap.

Proses ini bersifat iteratif sehingga sistem dapat terus diperbaiki dan dikembangkan sesuai dengan kebutuhan pengguna.

2.7. Product Backlog

Product backlog merupakan daftar seluruh fitur atau kebutuhan sistem yang akan dikembangkan dalam proyek perangkat lunak. Pada pengembangan sistem informasi Toko Tingwe Tembakau, product backlog disusun berdasarkan kebutuhan pengguna yang diperoleh melalui proses wawancara dengan pemilik toko. Setiap fitur yang terdapat dalam backlog kemudian diberikan tingkat prioritas untuk menentukan urutan pengembangan sistem. Berikut merupakan daftar product backlog pada sistem yang akan dikembangkan.

Tabel 1. Fitur Sistem

No	Fitur Sistem	Deskripsi	Prioritas
1	Login Sistem	Fitur untuk melakukan autentikasi pengguna sebelum mengakses sistem	Tinggi
2	Dashboard	Menampilkan ringkasan informasi seperti total transaksi, stok hampir habis, dan aktivitas toko	Tinggi
3	Transaksi Penjualan	Fitur untuk mencatat transaksi penjualan secara digital	Tinggi
4	Perhitungan Total Otomatis	Sistem menghitung total pembayaran secara otomatis saat transaksi	Tinggi
5	Update Stok Otomatis	Sistem mengurangi jumlah stok barang setiap terjadi transaksi	Tinggi
6	Manajemen Data Produk	Fitur untuk menambah, mengubah, dan menghapus data produk	Tinggi
7	Manajemen Stok Barang	Fitur untuk melihat jumlah stok barang yang tersedia	Tinggi
8	Notifikasi Stok Minimum	Sistem memberikan pemberitahuan ketika stok barang hampir habis	Tinggi
9	Pembelian dari Supplier	Fitur untuk mencatat pembelian barang dari pemasok	Sedang
10	Laporan Penjualan Harian	Menampilkan laporan transaksi penjualan harian	Tinggi
11	Laporan Penjualan Bulanan	Menampilkan laporan transaksi penjualan bulanan	Tinggi
12	Laporan Keuntungan	Sistem menghitung keuntungan berdasarkan data penjualan dan pembelian	Tinggi
13	Perhitungan Zakat Mal	Sistem menghitung zakat mal dari keuntungan usaha	Sedang
14	Riwayat Transaksi	Menampilkan riwayat transaksi penjualan yang telah dilakukan	Sedang
15	Manajemen Supplier	Fitur untuk mengelola data supplier	Sedang

2.8. Pengujian Sistem (Testing)

2.8.1. Konsep Dasar Testing

Testing merupakan proses pengujian sistem untuk memastikan bahwa sistem berjalan sesuai dengan fungsi yang diharapkan. Pengujian software sangat penting dilakukan dengan tujuan untuk mengetahui kesalahan yang ada saat dilakukannya pembuatan software [5]. Dalam pengembangan perangkat lunak, pengujian menjadi tahapan krusial untuk memastikan kualitas sebelum sistem diimplementasikan secara penuh kepada pengguna.

2.8.2. White Box Testing

White box testing disebut sebagai pengujian structural, yang mana perangkat lunak yang diuji merupakan hal transparan kepada penguji[5]. Pengujian white box dapat mengungkap kesalahan implementasi seperti manajemen kunci yang buruk dengan menganalisis cara kerja internal dan struktur dari sebuah perangkat lunak[6]. Berbeda dengan pengujian yang berfokus pada perilaku eksternal perangkat lunak, metode ini membutuhkan pengetahuan mengenai kode sumber, logika program, dan arsitektur sistem yang diuji. Pengujian ini bertujuan untuk memastikan bahwa jalur logika dan struktur internal aplikasi telah diperiksa secara memadai.

Salah satu bentuk penerapan white box testing adalah *Static Code Analysis* atau analisis kode statis. Analisis kode statis merupakan pemeriksaan terhadap kode sumber tanpa mengeksekusi program untuk mengidentifikasi potensi cacat, pelanggaran aturan pemrograman, masalah keamanan, kompleksitas, dan karakteristik kode lainnya[17]. Kajian sistematis terhadap alat

analisis kode statis menunjukkan bahwa alat-alat tersebut digunakan untuk mendeteksi berbagai jenis masalah perangkat lunak melalui pemeriksaan otomatis terhadap kode sumber[18]. SonarQube merupakan salah satu alat analisis kode statis yang dapat digunakan untuk mengidentifikasi metrik kualitas kode dan indikasi utang teknis, termasuk potensi bug, masalah keamanan, kompleksitas, *code smell*, dan duplikasi kode[19].

Dalam SonarQube, *Quality Gate* digunakan sebagai sekumpulan kondisi atau ambang batas untuk menilai apakah kualitas kode suatu proyek telah memenuhi kriteria yang ditetapkan. Konsep ini berkaitan dengan mekanisme pengambilan keputusan berbasis metrik sebelum perangkat lunak dinyatakan memenuhi standar kualitas tertentu[20]. tatus *Passed* menunjukkan bahwa kondisi yang ditetapkan telah terpenuhi, sedangkan status *Failed* menunjukkan bahwa terdapat satu atau lebih kondisi yang belum terpenuhi[19].

Metrik *Security* digunakan untuk menilai adanya potensi kerentanan keamanan (*vulnerabilities*) pada kode sumber[21]. Metrik *Reliability* berkaitan dengan *bugs* atau cacat kode yang berpotensi menyebabkan kegagalan dan perilaku sistem yang tidak sesuai[22]. Metrik *Maintainability* berkaitan dengan *code smells*, yaitu indikasi kelemahan pada struktur atau praktik pengodean yang dapat meningkatkan usaha untuk memahami, mengubah, dan memelihara perangkat lunak[20]. ementara itu, metrik *Duplications* digunakan untuk mengidentifikasi bagian kode yang berulang atau memiliki kemiripan, karena duplikasi dapat meningkatkan beban pemeliharaan dan menyulitkan perubahan kode secara konsisten[23].

2.8.3. Black Box Testing

Pengujian black-box merupakan teknik pengujian perangkat lunak di mana penguji tidak mengetahui cara kerja internal aplikasi dan lebih berfokus pada input serta output yang diharapkan untuk memverifikasi fungsionalitas[7]. Dalam pendekatan ini, penguji memperlakukan sistem sebagai kotak hitam yang hanya dapat diamati dari sisi input yang diberikan dan output yang dihasilkan. Implementasi metode black box testing menjadi tahapan vital dalam pengembangan perangkat lunak untuk memvalidasi bahwa fungsionalitas sistem telah selaras dengan ekspektasi dan kebutuhan pengguna[8].

2.8.4. Usability Testing

Usability testing merupakan sebuah tolok ukur untuk menilai seberapa mudah suatu sistem atau produk dipelajari dan dioperasikan oleh pengguna demi mencapai tujuan tertentu, sekaligus memberikan kepuasan dalam penggunaannya [9]. Salah satu instrumen standar yang umum digunakan untuk mengukur tingkat kemudahan ini adalah System Usability Scale (SUS) yang dikembangkan oleh John Brooke pada tahun 1986[25]. SUS merupakan standar industri yang reliabel untuk mengukur tingkat usability atau kegunaan suatu sistem atau produk berdasarkan persepsi pengguna[10].

3. METODE

3.1. Pendekatan dan Fokus Penelitian

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak dengan metode *Agile Modeling*. Metode ini dipilih karena berpusat pada pengguna (*user-centered approach*) dan mengutamakan fleksibilitas, kolaborasi, serta iterasi pengembangan melalui siklus *sprint*. Fokus operasional pada penelitian ini adalah perancangan dan implementasi Sistem Informasi Penjualan dan Manajemen Stok berbasis web (konsep *Point of Sale*) yang mengintegrasikan pencatatan transaksi digital, pembaruan stok otomatis, notifikasi inventaris, hingga pelaporan keuangan dan perhitungan Zakat Mal. Rekayasa sistem ini dilakukan melalui pembagian urutan prioritas fitur (*product backlog*) yang dieksekusi ke dalam empat siklus *Sprint Planning* berkala (durasi dua minggu per-sprint), dengan rincian operasional sebagai berikut:

1. Sprint 1 (Pengembangan Sistem Dasar): Berfokus pada perancangan struktur basis data, pembuatan halaman autentikasi/login, dan penyusunan halaman dashboard utama.
2. Sprint 2 (Modul Transaksi Penjualan): Berfokus pada pengembangan fitur utama *Point of Sale* (POS), meliputi pemilihan produk, kalkulasi total pembayaran otomatis, penyimpanan transaksi, dan pengurangan stok otomatis.

3. Sprint 3 (Manajemen Produk dan Stok): Berfokus pada fitur manipulasi data produk (CRUD), sistem notifikasi batas stok minimum, dan pencatatan restock barang dari supplier.
4. Sprint 4 (Modul Laporan dan Zakat): Berfokus pada pembuatan laporan penjualan harian/bulanan, perhitungan keuntungan otomatis, serta integrasi fitur perhitungan Zakat Mal harian (nisab 2,5% dari akumulasi aset lancar dan laba bersih) beserta pencatatan shadaqoh.

3.2. Ruang Lingkup dan Tempat Penelitian

Ruang lingkup objek penelitian mencakup analisis proses bisnis manual yang sedang berjalan, perancangan antarmuka (*UI/UX*), pengembangan basis data, hingga pengujian kelayakan sistem terintegrasi. Penelitian ini dilaksanakan dan diimplementasikan pada Toko Tingwe Tembakau, sebuah usaha retail UMKM yang berfokus pada penjualan produk tembakau dan perlengkapan linting.

3.3. Populasi dan Sampel

Pemilihan informan untuk tahap analisis kebutuhan sistem dilakukan secara *purposive*, yaitu melibatkan pemilik toko (*owner*) dan kasir sebagai pengguna utama (*end-user*). Sementara itu, untuk tahap pengujian kemudahan penggunaan (*Usability Testing*), sampel yang digunakan berjumlah 30 responden yang bersedia mencoba sistem, yang terdiri dari 29 mahasiswa (96,7%) dan 1 wirausaha/pemilik toko (3,3%).

3.4. Bahan dan Alat Utama

Pengembangan sistem didukung oleh beberapa alat utama yang mencakup perangkat lunak (*software*) dan kerangka kerja (*framework*). Desain antarmuka dirancang menggunakan aplikasi Figma. Sistem dibangun menggunakan *framework backend* Laravel 12 dengan bahasa pemrograman PHP 8.2, *framework frontend* Tailwind CSS v4 dan Alpine.js, serta sistem manajemen basis data MySQL (melalui XAMPP). Untuk tahap pengujian sistem, alat bantu yang digunakan adalah SonarQube Community Edition v26.4 untuk analisis kode statis dan Postman API Client untuk pengujian respons *endpoint*.

3.5. Teknik Pengumpulan Data

Data dalam penelitian ini dikumpulkan melalui tiga teknik utama:

1. Observasi Langsung: Mengamati secara langsung proses operasional, transaksi tatap muka, dan manajemen persediaan fisik yang masih menggunakan pencatatan buku tulis di Toko Tingwe.
2. Wawancara: Melakukan tanya jawab mendalam bersama pemilik toko untuk merumuskan spesifikasi masalah, *user stories*, dan penyusunan *product backlog*.
3. Kuesioner: Menyebarkan kuesioner berbasis metrik ukur *System Usability Scale* (SUS) melalui *Google Form* kepada 30 responden setelah mereka melakukan simulasi penggunaan sistem.

3.6. Teknik Analisis Data

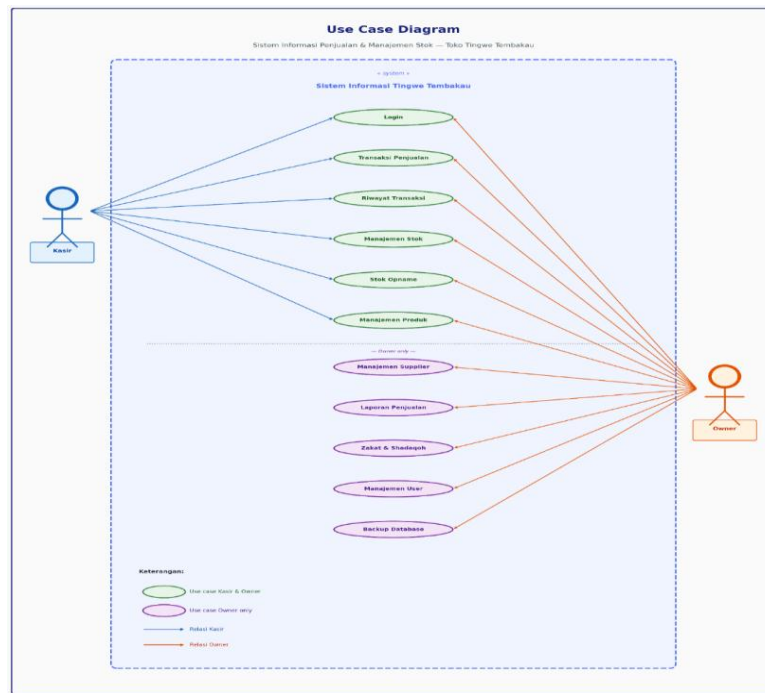
Analisis data dilakukan secara bertahap meliputi analisis proses bisnis menggunakan pemodelan *Logical Workflow* dan *Use Case Diagram*, serta analisis evaluasi sistem yang dibagi menjadi tiga metode pengujian:

1. *White Box Testing*: Menganalisis struktur kode internal secara statis menggunakan metrik evaluasi dari SonarQube untuk melihat tingkat *Security*, *Reliability*, *Maintainability*, dan duplikasi kode.
2. *Black Box Testing*: Menguji fungsionalitas fitur dan validasi *input/output* sistem menggunakan metode pengujian skenario berbasis peran (*owner* dan kasir) tanpa melihat logika struktur kodenya.
3. *Usability Testing*: Menganalisis tingkat kepuasan dan kemudahan pengguna dengan menghitung total skor kontribusi kuesioner SUS dari skala 1-5 (Sangat Tidak Setuju hingga Sangat Setuju). Skor kemudian dikonversi dengan mengalikan bobot 2,5 untuk menghasilkan persentase kelayakan berskala 0 hingga 100.

4. HASIL DAN PEMBAHASAN

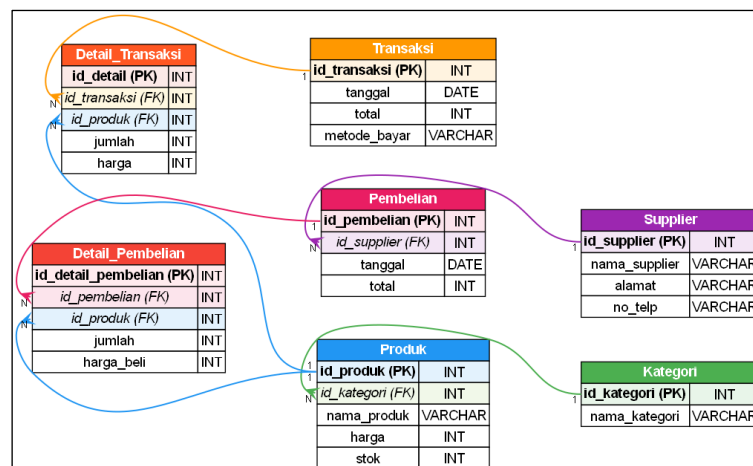
4.1. Hasil Perancangan Sistem

Perancangan Sistem Informasi Penjualan dan Manajemen Stok pada Toko Tingwe Tembakau difokuskan pada penyederhanaan proses operasional manual menjadi sistem Point of Sale (POS) digital yang terintegrasi. Analisis kebutuhan sistem menghasilkan desain logical workflow yang memisahkan hak akses ke dalam dua aktor utama, yaitu Pemilik Toko (Owner) dan Kasir. Kasir memiliki akses terbatas pada fungsi operasional harian seperti transaksi, sedangkan Pemilik Toko memiliki otoritas penuh terhadap manajemen produk, laporan keuangan, dan fitur khusus seperti perhitungan zakat.



Gambar 2. Use Case Diagram

Pada aspek penyimpanan data, basis data dirancang secara relasional menggunakan MySQL untuk memastikan integritas data antara aktivitas penjualan, pembelian ke pemasok (supplier), dan penyusutan stok barang.



Gambar 3. Entity Relationship Diagram Basis Data

Tabel 2. Spesifikasi Lingkungan Implementasi

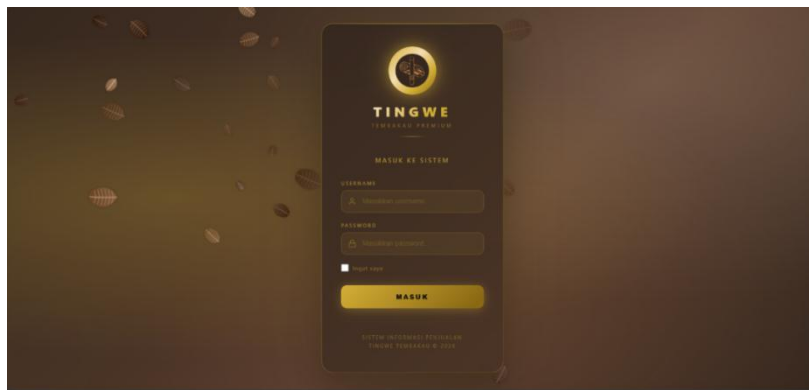
Komponen	Spesifikasi
Framework Backend	Laravel 12
Bahasa Pemrograman	PHP 8.2
Framework CSS	Tailwind CSS v4
JavaScript Framework	Alpine.js
Database	MySQL (via XAMPP)
Web Server	Apache (XAMPP) + Laravel Artisan Server
Port Server	8000 (localhost)
Asset Bundler	Vite
Development Tools	Visual Studio Code
Database Management	phpMyAdmin
Nama Database	tingwe_pos
Path Proyek	C:\xampp\htdocs\tingwe-pos

4.2. Hasil Implementasi Sistem

Sistem diimplementasikan dalam bentuk aplikasi berbasis web menggunakan framework backend Laravel 12 (PHP 8.2), serta Tailwind CSS v4 dan Alpine.js untuk membangun antarmuka yang dinamis. Hasil implementasi menghasilkan beberapa modul utama yang yaitu:

4.2.1. Halaman Login

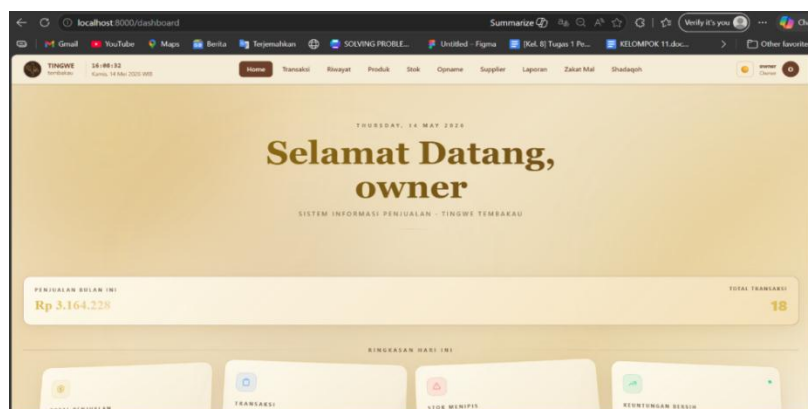
Halaman login berhasil diimplementasikan sesuai dengan rancangan UI Design. Sistem autentikasi menggunakan fitur Auth bawaan Laravel yang dilengkapi dengan middleware untuk memproteksi seluruh halaman dari akses tanpa autentikasi.



Gambar 4. Halaman Login

4.2.2. Halaman Dashboard

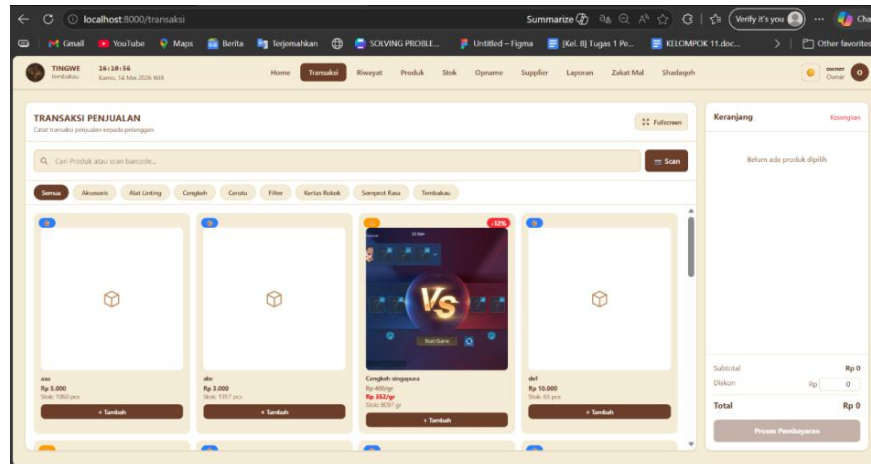
Halaman dashboard berhasil diimplementasikan dengan menampilkan ringkasan aktivitas toko secara real-time. Dashboard menampilkan total penjualan harian, jumlah transaksi, produk hampir habis, keuntungan, grafik penjualan bulanan, tabel produk terlaris, jadwal restock, dan notifikasi stok hampir habis.



Gambar 5. Halaman Dashboard

4.2.3. Halaman Transaksi Penjualan

Halaman transaksi penjualan diimplementasikan dengan tampilan Point of Sale (POS) yang memungkinkan kasir memilih produk, memasukkan ke keranjang, dan memproses pembayaran secara digital. Sistem secara otomatis mengurangi stok produk setiap kali transaksi berhasil dilakukan dan mencetak struk transaksi.



Gambar 6. Halaman Transaksi Penjualan



Gambar 7. Struk Transaksi

4.2.4. Halaman Manajemen Produk

Halaman manajemen produk diimplementasikan dengan fitur CRUD (Create, Read, Update, Delete) yang memungkinkan owner mengelola seluruh data produk. Sistem dilengkapi dengan validasi input agar data yang disimpan selalu valid.

Produk	Kategori	Nama	Barcode	Harga Jual	Harga Beli	Diskon	Stok	Status	Aksi
aaa	Sempurna Rasa	4 Pcs	77702875013	Rp 5.000/pcs	Rp 5.000/pcs	-	1060 pcs	Aman	Tambah
abx	Sempurna Rasa	4 Pcs	77702875017	Rp 3.000/pcs	Rp 3.000/pcs	-	1247 pcs	Aman	Tambah
Cengkeh singapura	Cengkeh	5 Gram	77702875014	Rp 400/gr	Rp 300/gr	12%	8079 gr	Aman	Tambah
def	Cendua	4 Pcs	77702875015	Rp 10.000/pcs	Rp 7.000/pcs	-	65 pcs	Aman	Tambah
egrltpe	Cendua	4 Pcs	-	Rp 113/pcs	Rp 122/pcs	-	123 pcs	Aman	Tambah
egrltpe	Cendua	4 Pcs	-	Rp 113/pcs	Rp 122/pcs	-	123 pcs	Aman	Tambah
gayah super	Tembakau	5 Gram	77702875016	Rp 400/gr	Rp 300/gr	-	328 gr	Aman	Tambah
lqjz	Cengkeh	4 Pcs	77702875018	Rp 5.555/pcs	Rp 4.444/pcs	-	888 pcs	Aman	Tambah

Gambar 8. Halaman Manajemen Produk

4.2.5. Halaman Stok Produk

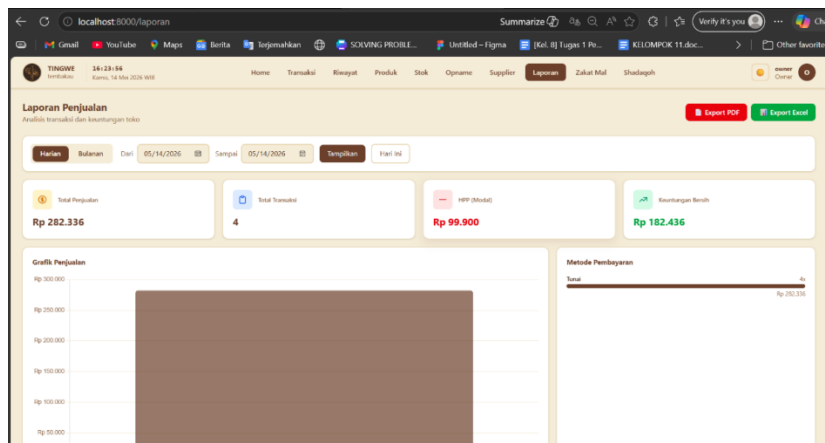
Halaman stok barang berhasil diimplementasikan dengan tampilan yang memperlihatkan status stok setiap produk secara real-time. Sistem memberikan label status berwarna berbeda untuk kondisi Aman, Stok Menipis, dan Stok Habis, serta menyediakan fitur restock yang terintegrasi dengan modul supplier.

Produk	Kategori	Stok Saat Ini	Stok Minimum	Status	Aksi
def	Cendua	65 pcs	5 pcs	Aman	Restock
Nizar Signature	Tembakau	100 pcs	5 pcs	Aman	Restock
egrltpe	Cendua	123 pcs	10 pcs	Aman	Restock
egrltpe	Cendua	123 pcs	10 pcs	Aman	Restock
gayah super	Tembakau	328 gr	1 gr	Aman	Restock
lqjz	Cengkeh	888 pcs	8 pcs	Aman	Restock
Kortas Buluk	Kortas Buluk	996 pcs	5 pcs	Aman	Restock

Gambar 9. Halaman Stok Produk

4.2.6. Halaman Laporan Penjualan

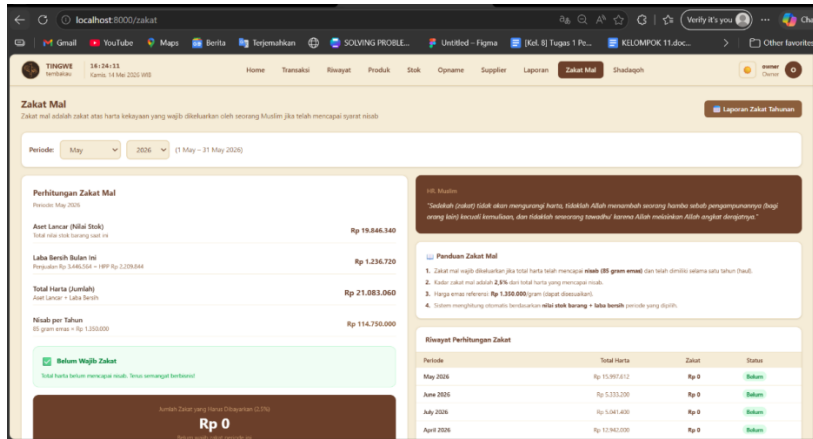
Halaman laporan penjualan diimplementasikan dengan fitur filter berdasarkan tanggal dan produk. Sistem menyediakan tampilan grafik penjualan bulanan serta tabel rincian transaksi. Fitur ekspor laporan dalam format PDF dan Excel juga berhasil diimplementasikan.



Gambar 10. Halaman Laporan Penjualan

4.2.7. Halaman Zakat Mal

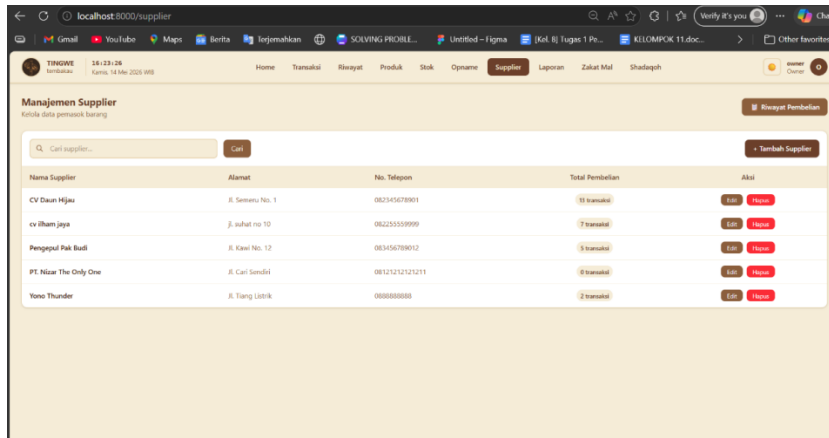
Halaman zakat mal diimplementasikan sebagai fitur unik sistem yang membantu pemilik toko memenuhi kewajiban zakat. Sistem secara otomatis menghitung nisab, total harta, dan jumlah zakat yang harus dibayarkan berdasarkan data aset dan laba usaha yang tersimpan dalam database.



Gambar 11. Halaman Zakat Mal

4.2.8. Halaman Supplier dan Pembelian

Halaman supplier dan pembelian berhasil diimplementasikan dengan fitur manajemen data supplier dan pencatatan riwayat pembelian barang. Setiap transaksi pembelian yang dicatat akan secara otomatis memperbarui jumlah stok produk yang bersangkutan.



Gambar 12. Halaman Supplier dan Pembelian

4.3. Hasil Pengujian Sistem

Untuk memastikan kelayakan, fungsionalitas, dan tingkat penerimaan pengguna, sistem dievaluasi menggunakan tiga metode pengujian yang komprehensif

4.3.1. White Box Testing (Analisis Statis)

Pengujian struktur internal kode dilakukan menggunakan SonarQube dengan hasil akhir Quality Gate Passed. Kode sistem memperoleh peringkat "A" pada aspek Keamanan (Security) yang mengindikasikan ketiadaan celah kerentanan (zero vulnerabilities), serta peringkat "A" pada Maintainability dengan tingkat duplikasi kode yang sangat efisien, yakni hanya 4,3%.

Tabel 3. Ringkasan Hasil Analisis SonarQube

Metrik	Nilai	Peringkat	Keterangan
Quality Gate	Passed	✓	Sistem memenuhi standar kualitas minimum
Total Lines of Code	17.000	—	Baris kode yang dianalisis
Total Issues	506	—	Estimasi perbaikan: 5 hari 4 jam

Security	0 issues	A	Tidak ada celah keamanan
Reliability	160 issues	C	Terdapat potensi bug aksesibilitas
Maintainability	348 issues	A	Code smell terdeteksi, mayoritas ringan
Duplications	4,3%	✓	Duplikasi kode sangat rendah
Security Hotspots	23	E	Perlu ditinjau manual (dari Docker config)
Coverage	0,0%	—	Belum ada automated unit test

4.3.2. Black Box Testing (Fungsionalitas)

Pengujian fungsionalitas difokuskan pada validasi input, middleware autentikasi, dan perlindungan route berdasarkan peran pengguna. Dari total 86 test case yang diuji menggunakan Postman API dan browser, sistem memperoleh Pass Rate 100% tanpa adanya error logika bisnis maupun kebocoran akses.

Tabel 4. Rekap Hasil Keseluruhan Black Box Testing

Skenario	Total Test Case	Pass	Fail	Pass Rate
Akses tanpa login	12	12	0	100%
Akses role Owner	11	11	0	100%
Akses role Kasir	11	11	0	100%
Fungsionalitas	52	52	0	100%
Total Keseluruhan	86	86	0	100%

4.3.3. Usability Testing (Kemudahan Penggunaan)

Pengujian tingkat penerimaan dan kepuasan pengguna akhir diukur menggunakan instrumen System Usability Scale (SUS) yang melibatkan 30 responden.

Tabel 5. Distribusi Predikat Skor SUS

Predikat	Rentang Skor	Jumlah Responden	Persentase
Excellent	85 – 100	8	26,7%
Good	71 – 84	8	26,7%
Above Average	68 – 70	3	10,0%
Below Average	51 – 67	9	30,0%
Poor	0 – 50	2	6,7%
Total	—	30	100%

Berdasarkan hasil kuesioner SUS, sistem memperoleh skor rata-rata sebesar 72,67. Merujuk pada skala penilaian SUS, skor tersebut masuk ke dalam predikat GOOD (rentang 71–84) dan berada pada kategori Acceptable. Hal ini membuktikan bahwa antarmuka sistem dinilai intuitif, fungsi operasionalnya mudah dipelajari (learnability), dan fungsionalitasnya telah berhasil memenuhi ekspektasi serta kebutuhan digitalisasi Toko Tingwe Tembakau secara efektif.

5. KESIMPULAN

Perancangan dan implementasi Sistem Informasi Penjualan dan Manajemen Stok pada Toko Tingwe Tembakau menggunakan metode *Agile Modeling* telah berhasil dilakukan. Sistem ini secara efektif menjawab kebutuhan tujuan penelitian dengan mengatasi permasalahan operasional manual melalui digitalisasi pencatatan transaksi, pembaruan stok secara terotomatisasi, sistem notifikasi inventaris minimum, serta pembuatan laporan penjualan dan perhitungan Zakat Mal secara instan. Evaluasi sistem menunjukkan hasil yang sangat memuaskan; pengujian *Black Box* mencapai tingkat keberhasilan 100% pada seluruh fungsionalitas, pengujian *White Box* membuktikan kualitas dan keamanan kode yang sangat baik (Predikat A pada aspek *Security* dan *Maintainability*), dan pengujian *Usability* menggunakan metrik *System Usability Scale* (SUS) memperoleh rata-rata skor 72,67 (Kategori *Good*), yang mengindikasikan bahwa antarmuka sistem intuitif dan mudah dioperasikan oleh pengguna.

Penelitian ini memberikan kontribusi praktis terhadap pengembangan sistem *Point of Sale* (POS) bagi Usaha Mikro, Kecil, dan Menengah (UMKM) retail, sekaligus memberikan kontribusi teoretis berupa inovasi penggabungan proses bisnis komersial dengan aspek sosial-religius melalui

fitur perhitungan Zakat Mal dan Shadaqoh secara terintegrasi. Untuk pengembangan aplikasi lebih lanjut, disarankan untuk melakukan pengoptimalan antarmuka agar lebih responsif pada perangkat seluler (*mobile*), peningkatan kontras warna teks untuk visibilitas yang lebih baik, serta penambahan validasi otomatis pada beberapa form *input*.

DAFTAR PUSTAKA

- [1] K. Fajri, A. Saputra, Z. Umar, and I. Albana, "Analisis pendekatan metode agile dalam manajemen proyek pada sistem informasi," *Jurnal Mahasiswa Sistem Informasi (JMSI)*, 2024.
- [2] A. Frisdayanti, "Peranan brainware dalam sistem informasi manajemen," *Jurnal Ekonomi Manajemen Sistem Informasi (JEMSI)*, vol. 1, no. 1, 2019.
- [3] K. E. Kendall and J. E. Kendall, *Systems analysis and design*, 10th ed. Pearson, 2019.
- [4] S. Setiawansyah, Q. J. Adrian, and R. N. Devija, "Penerapan sistem informasi administrasi perpustakaan menggunakan model desain user experience," *Jurnal Manajemen Informatika (JAMIKA)*, vol. 11, no. 1, pp. 24–36, 2021. doi: 10.34010/jamika.v11i1.3710.
- [5] A. C. Praniffa, A. Syahri, F. Sandes, U. Fariha, and Q. A. Giansyah, "Pengujian black box dan white box sistem informasi parkir berbasis web," *Jurnal Testing dan Implementasi Sistem Informasi*, vol. 1, no. 1, pp. 1-16, 2023.
- [6] M. E. Khan and F. Khan, "A comparative study of white box, black box and grey box testing techniques," (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 6, pp. 12-15, 2012.
- [7] Y. F. Putri, A. B. P. Irianto, and S. Sharma, "Comparison of automatic and manual regression testing on mobile application health technology with black box testing method," *Indonesian Journal of Information Systems (IJIS)*, vol. 7, no. 2, pp. 218–230, 2025.
- [8] M. P. A. Ginting and A. S. Lubis, "Pengujian aplikasi berbasis web data ska menggunakan metode black box testing," *COSMIC Jurnal Teknik*, vol. 2, no. 1, pp. 41-48, 2024. doi: 10.55537/cosmic.
- [9] D. Kurniawan and F. Yuamita, "Usability testing penggunaan menu kartu hasil studi di website sistem informasi akademik Universitas Teknologi Yogyakarta," *Jurnal Teknologi dan Manajemen Industri Terapan*, vol. 2, no. 1, pp. 41–52, 2023. doi: 10.55826/tmit.v2i1.57.
- [10] N. Kholis, M. Akbar, and A. Bardadi, "Pengukuran tingkat kemudahan pegawai terhadap penggunaan layanan aplikasi SIMRS menggunakan metode system usability scale (SUS) pada Rumah Sakit Umum Daerah Sekayu," *Jurnal Fasilkom*, vol. 12, no. 2, pp. 108-113, 2022.
- [11] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Information and Software Technology*, vol. 50, no. 9–10, pp. 833–859, Aug. 2008, doi: 10.1016/j.infsof.2008.01.006.
- [12] T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe, "A decade of agile methodologies: Towards explaining agile software development," *Journal of Systems and Software*, vol. 85, no. 6, pp. 1213–1221, Jun. 2012, doi: 10.1016/j.jss.2012.02.033.
- [13] R. Vallon, B. J. da Silva Estácio, R. Prikladnicki, and T. Grechenig, "Systematic literature review on agile practices in global software development," *Information and Software Technology*, vol. 96, pp. 161–180, Apr. 2018, doi: 10.1016/j.infsof.2017.12.004.
- [14] J. Recker, R. Holten, M. Hummel, and C. Rosenkranz, "How agile practices impact customer responsiveness and development success: A field study," *Project Management Journal*, vol. 48, no. 2, pp. 99–121, Apr. 2017, doi: 10.1177/875697281704800208.
- [15] S. W. Ambler, *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process*. New York, NY, USA: John Wiley & Sons, 2002.
- [16] J. Erickson, K. Lyytinen, and K. Siau, "Agile modeling, agile software development, and extreme programming: The state of research," *Journal of Database Management*, vol. 16, no. 4, pp. 88–100, Oct. 2005, doi: 10.4018/jdm.2005100105.

- [17] D. Stefanović, D. Nikolić, D. Dakić, I. Spasojević, and S. Ristić, "Static code analysis tools: A systematic literature review," in *Proc. 31st DAAAM Int. Symp. Intell. Manuf. Autom.*, 2020, pp. 565–573, doi: 10.2507/31st.daaam.proceedings.078.
- [18] S. Heckman and L. Williams, "A systematic literature review of actionable alert identification techniques for automated static code analysis," *Information and Software Technology*, vol. 53, no. 4, pp. 363–387, Apr. 2011, doi: 10.1016/j.infsof.2010.12.007.
- [19] D. Guaman, P. A.-Q. Sarmiento, L. Barba-Guamán, P. Cabrera, and L. Enciso, "SonarQube as a tool to identify software metrics and technical debt in the source code through static analysis," in *Proc. 7th Int. Workshop Comput. Sci. Eng. (WCSE)*, Beijing, China, 2017, pp. 171–175, doi: 10.18178/wcse.2017.06.030.
- [20] R. Almeida, N. R. O. Bastos, and M. T. T. Monteiro, *Fractional Differential Equations: An Approach via Fractional Derivatives with Respect to Another Function*. Cham, Switzerland: Springer, 2020, doi: 10.1007/978-3-030-47230-5.
- [21] J. P. da Silva, D. A. da Silva, and C. de O. Melo, "A systematic literature review on the use of static analysis tools to identify technical debt," in *Proc. 16th ACM/IEEE Int. Symp. Empirical Softw. Eng. Meas. (ESEM)*, Helsinki, Finland, 2022, pp. 248–257, doi: 10.1145/3549035.3561182.
- [22] T. Hall, S. Beecham, D. Bowes, D. Gray, and S. Counsell, "A systematic literature review on fault prediction performance in software engineering," *IEEE Transactions on Software Engineering*, vol. 38, no. 6, pp. 1276–1304, Nov./Dec. 2012, doi: 10.1109/TSE.2011.103.
- [23] S. Mishra, D. Mishra, and G. H. Santra, "Adaptive boosting of weak regressors for forecasting of crop production considering climatic variability: An empirical assessment," *Journal of King Saud University - Computer and Information Sciences*, vol. 32, no. 8, pp. 949–964, Oct. 2020, doi: 10.1016/j.jksuci.2017.12.004.
- [24] M. T. I. Rahmayani, N. Nur, A. Hurairah, I. Saputra, and P. Nadia, "Sistem zakat digital berbasis web dengan pendekatan agile untuk meningkatkan keakuratan perhitungan pada Kabupaten Bengkalis," *Djtechno: Jurnal Teknologi Informasi*, vol. 6, no. 3, pp. 863–877, 2025.
- [25] J. Brooke, "SUS: A "quick and dirty" usability scale," in *Usability Evaluation in Industry*, P. W. Jordan, B. Thomas, I. L. McClelland, and B. Weerdmeester, Eds. London, U.K.: Taylor & Francis, 1996, pp. 189–194.