

APLIKASI KASIR BERBASIS WEB TERINTEGRASI DENGAN LAPORAN PENJUALAN MENGGUNAKAN ALGORITMA *FLUXSORT* (STUDI KASUS: TOKO JOEL, KELURAHAN PUNGGOLAKA, KOTA KENDARI)

Laode Ikhwanul Uzlah^{*1}, Statiswaty¹, Jumadil Nangi¹

¹Jurusan Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari, Indonesia

Penulis korespondensi: Laode Ikhwanul Uzlah
(laodeikhwanuluzlah_e1e122017@student.uho.ac.id)

Pencatatan transaksi manual pada sebuah usaha memicu kesalahan perhitungan, kesulitan pencarian data, serta keterlambatan pelaporan. Penelitian ini bertujuan mengembangkan aplikasi kasir berbasis web dengan mengimplementasikan algoritma Fluxsort sebagai metode pengurutan data di lapisan aplikasi. Sistem dibangun menggunakan Python/Flask dan basis data SQLite, serta beroperasi secara luring (offline) pada jaringan lokal. Algoritma Fluxsort diintegrasikan untuk mengurutkan daftar produk pada modul Kasir serta laporan agregasi penjualan pada modul Admin. Pendekatan ini efektif menghindari latensi query berulang pada server basis data serta menangani agregasi multi-tabel. Hasil pengujian fungsional mencapai keberhasilan 100% dari 11 skenario uji. Pengujian kinerja membuktikan bahwa algoritma Fluxsort mampu menyelesaikan pengurutan 1.000 entri data acak dalam waktu 16,508 milidetik. Black Box Testing memvalidasi 20 skenario antarmuka secara 100% valid. User Acceptance Testing (UAT) terhadap 5 responden menghasilkan indeks 91% (Sangat Baik). Dengan demikian, sistem ini berstatus Sangat Layak sebagai validasi awal solusi digitalisasi operasional yang spesifik pada lingkungan Toko Joel.

Kata Kunci - Aplikasi Kasir, Fluxsort, Laporan Penjualan, Sistem Informasi, Web.

I. PENDAHULUAN

Perkembangan teknologi informasi telah membawa perubahan signifikan dalam berbagai sektor, salah satunya pada sektor perdagangan ritel [1]. Digitalisasi tidak lagi sekadar menjadi tren, melainkan telah menjadi kebutuhan mendasar dalam menunjang efektivitas operasional, efisiensi waktu, serta ketepatan pengelolaan data [2]. Data Kementerian Koperasi dan UKM mencatat bahwa jumlah

UMKM di Indonesia pada tahun 2022 mencapai lebih dari 65 juta unit, namun hanya sekitar 26,5% yang telah terhubung dengan ekosistem digital. Keterbatasan ini berdampak langsung pada efisiensi operasional dan kemampuan analisis penjualan [3].

Toko Joel, sebuah toko sembako yang berlokasi di Kelurahan Punggolaka, Kota Kendari, merupakan salah satu UMKM yang masih mengandalkan pencatatan transaksi secara manual, yakni dengan menggunakan buku tulis. Metode ini menimbulkan berbagai kendala, seperti risiko kehilangan data, kesulitan rekapitulasi penjualan, keterbatasan pencarian transaksi, serta tidak tersedianya informasi stok dan produk terlaris secara akurat. Proses perhitungan pembayaran juga masih dilakukan dengan menggunakan kalkulator secara manual sebelum dicatat ke dalam buku transaksi, yang berpotensi menyebabkan kesalahan dan memperlambat pelayanan. Kondisi serupa juga dihadapi oleh berbagai UMKM di Indonesia yang belum terhubung dengan ekosistem digital [4] [5].

Untuk menjawab permasalahan tersebut, maka diperlukan sebuah sistem aplikasi kasir berbasis web yang tidak hanya mampu mencatat transaksi secara otomatis, tetapi sekaligus menyajikan laporan penjualan yang informatif, terstruktur, dan dapat diurutkan sesuai kebutuhan analisis. Pada sistem ini, proses pengurutan data menjadi salah satu kebutuhan fungsional yang penting. Penelitian ini menerapkan pengurutan di lapisan aplikasi dengan dua alasan: (1) pada modul kasir, data produk dimuat sekali ke memori untuk menghindari latensi berulang; (2) pada modul admin, nilai yang diurutkan merupakan hasil agregasi multi-tabel yang dihitung di lapisan aplikasi, sehingga lebih efisien dibanding mengirimkan *query* baru ke basis data secara berulang [6].

Algoritma *Fluxsort* dipilih sebagai mekanisme pengurutan. *Fluxsort* adalah algoritma *hybrid* yang

menggabungkan *branchless Quicksort* dan *Quadsort*, bersifat stabil, adaptif terhadap pola data sebagian terurut, dan efisien dalam menangani duplikat [7]. Meskipun keunggulan algoritma *hybrid* telah terbukti secara teoritis [8] [9], implementasi *Fluxsort* secara langsung pada sistem kasir berbasis web untuk UMKM belum pernah dikaji sebelumnya. Sehingga penelitian ini mengisi celah tersebut dengan membuktikan efisiensi dan validitas *Fluxsort* sebagai solusi pengurutan data di lapisan aplikasi.

II. METODE PENELITIAN

A. Metode Pengumpulan Data

Pengumpulan data dilakukan melalui tiga teknik:

1. Observasi langsung di Toko Joel untuk memahami alur transaksi penjualan dan kendala operasional yang dihadapi.
2. Wawancara terstruktur dengan pemilik toko dan kasir untuk memperoleh informasi mengenai kebutuhan pengguna.
3. Studi literatur terhadap berbagai jurnal dan artikel ilmiah, buku, dokumentasi resmi *Fluxsort*, serta referensi terkait sistem berbasis web.

B. Metode Pengembangan Sistem

Pengembangan sistem menggunakan metode *Rational Unified Process* (RUP) yang bersifat iteratif dan terstruktur [10]. Tahapan meliputi:

1. *Inception*: berfokus pada identifikasi permasalahan, kebutuhan pengguna, serta ruang lingkup sistem yang akan dikembangkan.
2. *Elaboration*: pemodelan alur sistem melalui diagram alir (*flowchart*), perancangan struktur basis data menggunakan *Entity Relationship Diagram* (ERD) dengan Notasi Chen, serta pemodelan sistem menggunakan *Unified Modeling Language* (UML) yang mencakup *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*.
3. *Construction*: implementasi sistem dengan menggunakan bahasa pemrograman Python dan *framework* Flask, juga integrasi algoritma *Fluxsort* untuk pengurutan data pada lapisan aplikasi, serta penggunaan SQLite sebagai media penyimpanan data
4. *Transition*: berfokus pada pengujian, evaluasi, dan penyerahan sistem, dengan menggunakan tiga metode utama, yaitu pengujian efektivitas algoritma *Fluxsort* untuk memastikan akurasi pengurutan, *Black Box Testing* untuk memverifikasi fungsionalitas fitur, serta *User Acceptance Testing* (UAT) untuk menilai tingkat penerimaan pengguna akhir.

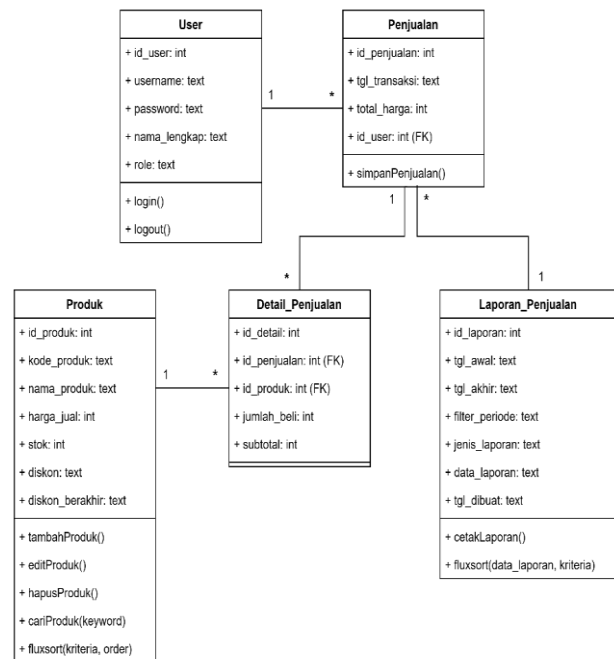
C. Arsitektur Sistem

Sistem beroperasi secara luring (*offline*) pada jaringan lokal (WLAN) toko. Arsitektur terdiri dari dua modul terintegrasi, yakni Modul Kasir untuk transaksi *real-time*, dan Modul Admin untuk pengelolaan data dan laporan. Pemisahan tugas diterapkan tegas antara mesin basis data (SQLite untuk agregasi SQL) dan lapisan aplikasi (Python/Flask untuk pengurutan dengan menggunakan algoritma *Fluxsort*). Pendekatan ini serupa dengan pola yang digunakan pada sistem kasir berbasis web lainnya

[11], namun dengan tambahan algoritma pengurutan di lapisan aplikasi [7].

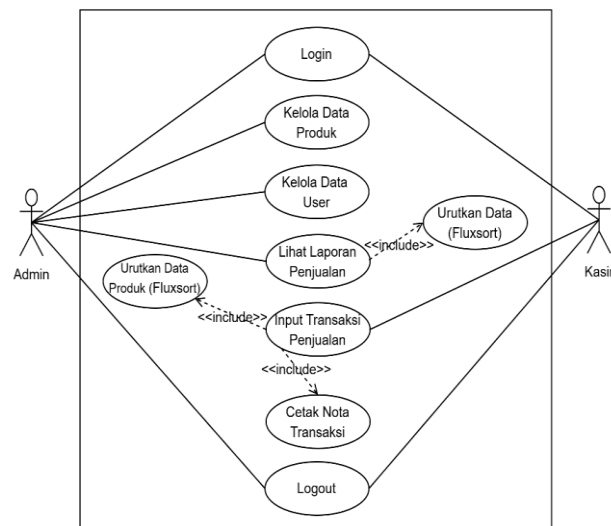
D. Implementasi Algoritma Fluxsort

Fluxsort [7] diimplementasikan dalam Python dengan empat komponen utama, yaitu fungsi rekursi utama *_fluxsort_recursive*, sub-algoritma *Quadsort* untuk data ≤ 96 elemen, *Analyzer* untuk mendeteksi pola keterurutan sebelum partisi, partisi adaptif menggunakan pivot *quasimedian-of-9* [12] [13], dan *stable merge* untuk menjaga stabilitas pada data duplikat [14]. Implementasi ini diterapkan pada dua modul sistem yang dimodelkan melalui struktur kelas Python/Flask sebagaimana ditunjukkan pada Gambar 1.



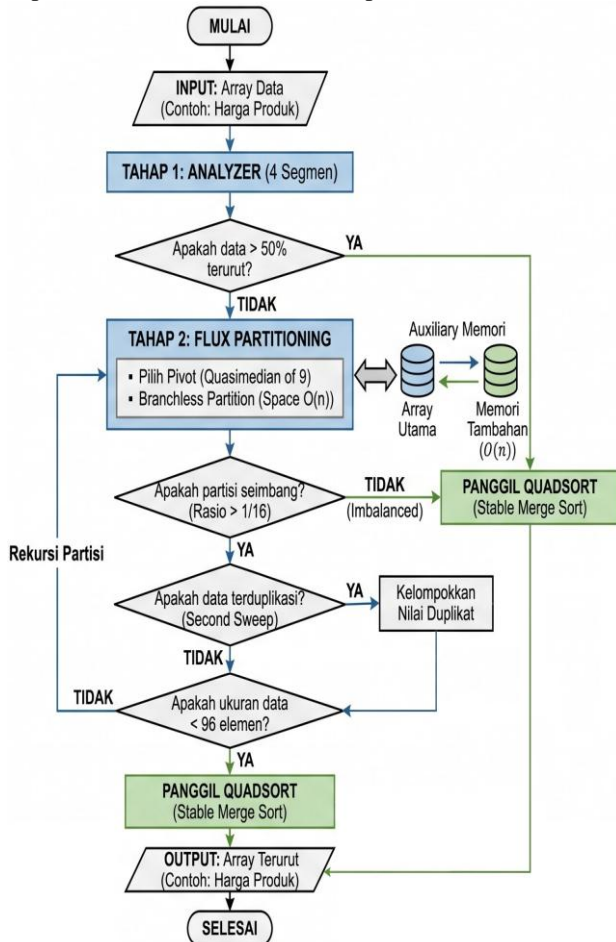
Gambar 1. Class Diagram Sistem Aplikasi Kasir

Kelas Produk menjadi objek pengurutan pada Modul Kasir, sedangkan kelas Laporan mengolah data agregasi yang diurutkan pada Modul Admin setelah proses SQL selesai dieksekusi. Kedua modul ini melayani dua aktor dengan hak akses berbeda sebagaimana dimodelkan pada Gambar 2.



Gambar 2. Use Case Diagram Sistem Aplikasi Kasir

Aktor Admin mengelola data produk, data *user*, dan laporan penjualan dengan fitur pengurutan menggunakan algoritma *Fluxsort* pada tiga kategori: Produk Terlaris, Transaksi Terbesar, dan Tren Pendapatan. Sementara untuk Aktor Kasir menangani transaksi *real-time* dengan pengurutan produk dinamis berdasarkan nama, harga, atau stok. Pada kedua modul, *Fluxsort* dipanggil secara dinamis menggunakan fungsi *lambda* sebagai *sort key* sesuai kriteria yang dipilih pengguna. Alur kerja lengkap implementasi ini divisualisasikan pada Gambar 3.



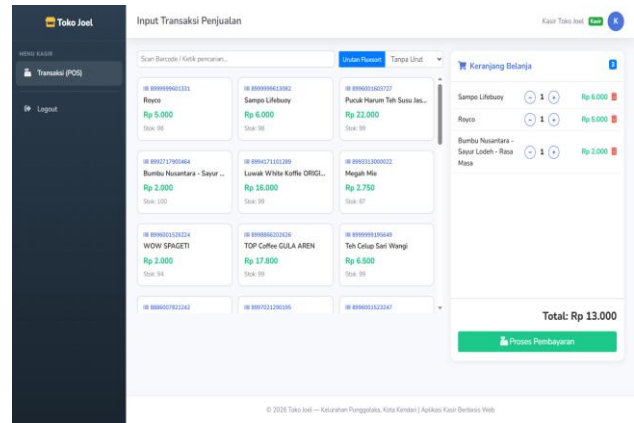
Gambar 3. Mekanisme Hybrid Algoritma Fluxsort

Berdasarkan alur kerja tersebut, algoritma *Fluxsort* hanya melakukan komputasi penuh ketika data benar-benar acak. Pada kondisi data sudah terurut sebagian, *Analyzer* memotong proses partisi dan mengalihkan langsung ke *Quadsort*, sehingga beban komputasi berkurang secara signifikan. Pendekatan ini menjadikan *Fluxsort* lebih efisien dibanding penggunaan *SQL ORDER BY* secara berulang, khususnya untuk kebutuhan pengurutan dinamis multi-kriteria pada sistem kasir UMKM.

III. HASIL DAN PEMBAHASAN

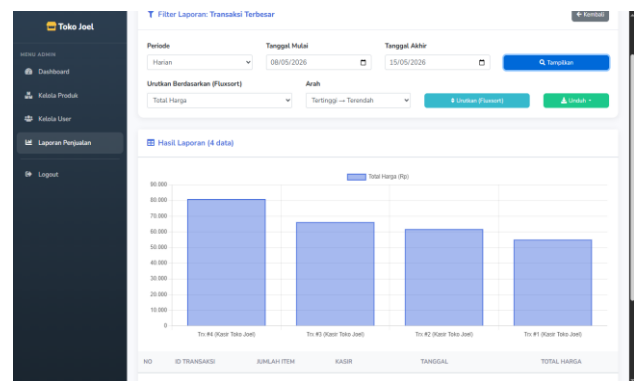
A. Implementasi Antarmuka Sistem

Antarmuka sistem diimplementasikan menggunakan HTML, CSS, JavaScript dengan *template* SBAdmin dan berjalan pada *localhost*. Gambar 4 menampilkan antarmuka Modul Kasir yang mengintegrasikan fitur pencarian produk via *barcode*, pengurutan *Fluxsort* secara dinamis, dan pengelolaan keranjang belanja.



Gambar 4. Antarmuka Halaman Input Transaksi Penjualan

Gambar 5 menampilkan antarmuka laporan penjualan pada Modul Admin. Setelah Admin memilih kategori laporan dan rentang waktu, algoritma *Fluxsort* akan mengeksekusi pengurutan hasil agregasi secara instan. Laporan yang telah terurut dapat diunduh dalam format Excel atau CSV.



Gambar 5. Antarmuka Halaman Laporan Penjualan

B. Hasil Pengujian Fungsional Fluxsort

Pengujian fungsional dilakukan untuk memverifikasi kebenaran *output* algoritma *Fluxsort* pada seluruh skenario pengurutan yang dirancang, dengan mencakup data teks dan numerik pada modul kasir maupun admin.

Tabel 1. Hasil Pengujian Fungsional Fluxsort

Skenario	Data Awal	Kriteria Urut	Hasil	Status
Nama produk acak (Kasir)	100 produk acak	A-Z	Terurut benar	Valid
Nama produk acak (Kasir)	100 produk acak	Z-A	Terurut benar	Valid
Harga produk (Kasir)	100 produk acak	Termurah-Termahal	Terurut benar	Valid
Harga produk (Kasir)	100 produk acak	Termahal-Termurah	Terurut benar	Valid
Jumlah stok (Kasir)	100 produk acak	Terbesar-Terkecil	Terurut benar	Valid
Jumlah stok (Kasir)	100 produk acak	Terkecil-Terbesar	Terurut benar	Valid
Produk terlaris (Admin)	100 produk agregasi penjualan	Tertinggi-Terendah	Terurut benar	Valid

Skenario	Data Awal	Kriteria Urut	Hasil	Status
Transaksi terbesar (Admin)	100 produk agregasi penjualan 49 hari	Terbesar-Terkecil	Terurut benar	Valid
Tren pendapatan (Admin)	data pendapatan	Tertinggi-Terendah	Terurut benar	Valid
Data sudah terurut - uji adaptivitas	100 produk terurut A-Z	A-Z	Tanpa urut ulang	Valid
Data terbalik penuh - uji adaptivitas	100 produk terurut Z-A	A-Z	Optimasi Reversal	Valid

Berdasarkan Tabel 1, algoritma *Fluxsort* berhasil mengeksekusi seluruh 11 skenario dengan tingkat keberhasilan 100%. Dua skenario terakhir secara khusus membuktikan kemampuan adaptivitas algoritma, yakni pada data yang sudah terurut sempurna, *Fluxsort* tidak melakukan pengurutan ulang, sedangkan pada data yang terbalik penuh, algoritma memicu optimasi *reversal* secara instan tanpa melewati proses partisi penuh. Hasil ini mengonfirmasi bahwa algoritma *Fluxsort* tidak hanya benar secara fungsional, tetapi juga adaptif terhadap kondisi data nyata yang bervariasi pada lingkungan transaksi toko.

C. Hasil Pengujian Kinerja dan Adaptivitas

Pengujian kinerja dilakukan di lapisan aplikasi tanpa latensi basis data, mengukur operasi perbandingan dan waktu eksekusi (ms).

Tabel 2. Hasil Pengujian Kinerja *Fluxsort*

Distribusi Data	Jumlah Elemen	Operasi Perbandingan	Waktu (ms)	Kategori
Acak penuh (Kasir)	10	33	0,058	Skala sangat kecil
Sudah terurut (Kasir)	10	9	0,017	Kasus terbaik skala kecil
Acak penuh (Kasir)	50	276	0,277	Skala harian toko
Sudah terurut (Kasir)	50	49	0,065	Kasus terbaik - uji adaptivitas
Semi terurut (Kasir)	50	220	0,203	Kondisi nyata transaksi Uji
Banyak duplikat (Kasir)	50	295	0,259	kelemahan <i>equal keys</i>
Acak penuh (Kasir)	100	665	4,392	Skala menengah
Semi terurut (Kasir)	100	610	0,638	Kondisi nyata skala besar

Distribusi Data	Jumlah Elemen	Operasi Perbandingan	Waktu (ms)	Kategori
Banyak duplikat (Kasir)	100	663	1,100	Uji kelemahan skala besar
Acak penuh (Admin)	50 transaksi	270	0,193	Laporan skala kecil
Semi terurut (Admin)	100 transaksi	583	0,441	Kondisi nyata laporan
Acak penuh (Admin)	500 transaksi	5606	3,609	Laporan skala besar
Semi terurut (Admin)	500 transaksi	4085	3,748	Uji adaptivitas laporan
Acak penuh (Admin)	1000 transaksi	11093	16,508	Uji batas atas sistem

Berdasarkan Tabel 2, efisiensi paling signifikan terlihat pada perbandingan data acak dan data terurut di skala yang sama. Pada 50 elemen, kondisi sudah terurut hanya membutuhkan 49 operasi perbandingan dibanding 276 pada kondisi acak, dengan penghematan sebesar 82%. Pola serupa konsisten pada skala 100 elemen hingga 500 transaksi, membuktikan bahwa mekanisme *Analyzer* bekerja efektif dalam memotong komputasi yang tidak diperlukan. Pada uji batas atas sistem dengan 1000 entri acak, *Fluxsort* menyelesaikan pengurutan hanya dalam 16,508 milidetik (setara dengan 0,016508 detik), yang mengonfirmasi bahwa pendekatan pengurutan di lapisan aplikasi lebih optimal dibanding pemanggilan *query ORDER BY* berulang ke basis data.

D. Hasil Black Box Testing

Black Box Testing dilakukan untuk memverifikasi bahwa setiap fitur pada aplikasi kasir berfungsi sesuai dengan hasil yang diharapkan berdasarkan rancangan sistem. Pendekatan pengujian ini berfokus penuh pada fungsionalitas antarmuka dan interaksi pengguna, dengan setiap modul dievaluasi melalui pemberian masukan (*input*) tertentu untuk kemudian diamati kesesuaian keluarannya (*output*) tanpa meninjau struktur kode internal logika program.

Tabel 3. Hasil *Black Box Testing*

Fitur Sistem	Skenario Uji	Hasil yang Diharapkan	Hasil Pengujian	Ket.
<i>Login</i>	<i>Input</i> akun admin	Masuk ke <i>dashboard</i> Admin	Berhasil masuk	Valid
<i>Login</i>	<i>Input</i> akun kasir	Masuk ke halaman POS	Berhasil masuk	Valid
<i>Login</i>	<i>Input</i> akun salah	Pesan <i>error</i> tampil	<i>Error</i> tampil	Valid
<i>Logout</i>	Klik <i>Logout</i>	Kembali ke <i>login</i>	Kembali ke <i>login</i>	Valid
Tambah Produk	<i>Input</i> <i>form</i> , Simpan	Produk tersimpan	Tersimpan	Valid
Edit Produk	Ubah data,	Data diperbarui	Diperbarui	Valid

Fitur Sistem	Skenario Uji	Hasil yang Diharapkan	Hasil Pengujian	Ket.
	Simpan			
Hapus Produk	Klik hapus	Produk terhapus	Terhapus	Valid
Tambah User	Isi form, Simpan	Produk tersimpan	Tersimpan	Valid
Edit User	Ubah data, Simpan	Data diperbarui	Diperbarui	Valid
Hapus User	Klik hapus	User terhapus	Terhapus	Valid
Laporan Penjualan	Generate laporan	Laporan tampil	Tampil sesuai	Valid
Fluxsort Laporan	Pilih kriteria urut	Data terurut	Terurut benar	Valid
Unduh Laporan	Klik Download CSV	File terunduh	CSV terunduh	Valid
Pencarian Produk	Ketik di pencarian	Produk terfilter	Terfilter	Valid
Scan Barcode	Scan via scanner/HP	Produk masuk keranjang	Masuk keranjang	Valid
Tambah Qty	Klik tombol (+)	Qty bertambah	Bertambah	Valid
Kurangi/Hapus Qty	Klik tombol (-)	Qty berkurang/hapus	Berkurang/g/Hapus	Valid
Proses Transaksi	Bayar dan proses	Transaksi tersimpan	Tersimpan	Valid
Fluxsort POS	Pilih urutan Fluxsort	Produk terurut	Terurut benar	Valid
Cetak Struk	Klik Cetak Struk	Struk tercetak	Tercetak	Valid

E. Hasil User Acceptance Testing (UAT)

UAT melibatkan 5 responden dengan Skala *Likert* lima tingkat (STS=1 hingga SS=5) pada 8 pernyataan yang mencakup kesesuaian fungsional, efisiensi kinerja, dan kemudahan penggunaan. Kelima responden tersebut merupakan pengguna akhir (*end-user*) yang berinteraksi langsung dengan sistem, terdiri dari 1 orang pemilik toko dan 4 orang staf kasir Toko Joel.

Mengingat ukuran sampel responden yang terbatas pada lingkungan operasional tunggal, hasil kelayakan ini direpresentasikan sebagai evaluasi dan validasi awal pada studi kasus Toko Joel, bukan sebagai generalisasi kelayakan untuk ekosistem UMKM secara luas.

Tabel 4. Hasil User Acceptance Testing

No	Pernyataan	STS	TS	N	S	SS	Subtotal
Aspek Kesesuaian Fungsional							
1.	Sistem memproses transaksi penjualan dengan benar	0	0	0	1	4	24
2.	Data produk dapat disimpan dan ditampilkan	0	0	0	1	4	24

3.	dengan akurat Laporan penjualan sesuai dengan data transaksi	0	0	0	1	4	24
Aspek Efisiensi Kinerja							
4.	Sistem merespons perintah dengan cepat	0	0	0	2	3	23
5.	Sistem berjalan stabil tanpa gangguan saat digunakan	0	0	0	3	2	22
Aspek Kemudahan Penggunaan							
6.	Antarmuka sistem mudah dipahami oleh pengguna	0	0	1	3	1	20
7.	Navigasi sistem sederhana dan nyaman digunakan	0	0	0	3	2	22
8.	Secara keseluruhan sistem mudah dioperasikan	0	0	0	2	3	23

Berdasarkan Tabel 4, diperoleh total skor aktual dari seluruh pernyataan kuesioner sebesar 182 dari skor maksimum 200, sehingga:

$$Persentase = (182 / 200) \times 100\% = 91\% \quad (1)$$

Hasil 91% berada pada kategori Sangat Baik, yang membuktikan bahwa sistem memenuhi kebutuhan operasional pengguna dan layak diterapkan sebagai solusi digitalisasi di Toko Joel.

IV. KESIMPULAN DAN SARAN

Aplikasi kasir berbasis web berhasil dirancang sebagai solusi digitalisasi bagi Toko Joel, yang beroperasi secara luring pada jaringan lokal dengan dua modul terintegrasi. Algoritma *Fluxsort* berhasil diimplementasikan pada lapisan aplikasi untuk pengurutan data produk pada Modul Kasir dan pengurutan data agregasi laporan penjualan pada Modul Admin. Pengujian fungsional membuktikan keberhasilan 100% pada 11 skenario uji, termasuk uji adaptivitas. Pengujian kinerja mengonfirmasi penyelesaian pengurutan 1.000 entri acak dalam waktu 16,508 milidetik, dengan efisiensi operasi perbandingan hingga 82% pada data terurut. Selanjutnya, *Black Box Testing* memvalidasi seluruh 20 skenario alur kerja sistem dengan hasil 100% valid, dan *User Acceptance Testing* (UAT) menghasilkan indeks kelayakan 91% (Sangat Baik) dari 5 responden sebagai evaluasi awal kelayakan. Berdasarkan batasan data dan skenario yang diuji, serta ketiadaan pengujian perbandingan (*benchmarking*) dengan algoritma lain, *Fluxsort* secara spesifik terbukti efisien, adaptif, dan sangat layak diterapkan pada lingkungan operasional Toko Joel. Namun demikian, kesimpulan kelayakan ini bersifat terbatas pada studi kasus terkait dan belum dapat digeneralisasi sebagai solusi sistem informasi untuk ekosistem UMKM secara luas.

Saran pengembangan meliputi: (1) penambahan fitur laporan laba rugi dan manajemen stok otomatis; (2) mengkaji implementasi algoritma *Fluxsort* pada bahasa pemrograman dengan optimasi kompilasi tingkat tinggi; (3) migrasi ke arsitektur *cloud* untuk mendukung operasional multi-cabang; (4) melakukan perbandingan kinerja (*benchmarking*) secara empiris antara *Fluxsort* dengan algoritma hibrida modern lainnya, seperti *OptiFlexSort* dan *TimSort*, pada skala data yang lebih besar.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Universitas Halu Oleo beserta seluruh civitas akademika atas fasilitas dan dukungan akademik yang diberikan selama proses penelitian berlangsung. Terima kasih kepada dosen pembimbing atas arahan, masukan, dan bimbingan yang diberikan hingga penelitian ini selesai. Penghargaan yang tulus pula penulis sampaikan kepada kedua orang tua atas dukungan moral dan materil yang tak ternilai.

Penulis juga mengucapkan terima kasih kepada pemilik Toko Joel, Kelurahan Punggolaka, Kota Kendari, yang telah bersedia menjadi objek penelitian dan memberikan akses data operasional. Penulis pula mengucapkan terima kasih kepada Dewan Redaksi Jurnal SIMTEK (Jurnal Sistem Informasi dan Teknik Komputer) atas kesempatan publikasi, serta kepada seluruh pihak yang terlibat dalam proses penyuntingan dan publikasi artikel ini sehingga dapat terselesaikan dengan baik.

DAFTAR PUSTAKA

- [1] S. R. Pratamansyah, "Transformasi Digital dan Pertumbuhan UMKM: Analisis Dampak Teknologi pada Kinerja Usaha Kecil dan Menengah di Indonesia," *J. Akuntansi, Manajemen, dan Perenc. Kebijakan.*, vol. 2, no. 2, pp. 1–17, 2024, doi: 10.47134/jampk.v2i2.475.
- [2] L. Mendrofa, B. Zendrato, and I. Zai, "Pengaruh Digitalisasi Pada Peningkatan Efisiensi Operasional Usaha Mikro, Kecil, dan Menengah (UMKM) di Indonesia Tahun 2023," *IDENTIK J. Ilmu Ekon. Pendidik. dan Tek.*, vol. 02, no. 01, pp. 100–108, 2025, [Online]. Available: <https://sihojournal.com/index.php/identik/article/view/251>
- [3] L. Firdaus, R. Amelia, and L. Hakim, "Strategi Peningkatan Usaha Mikro, Kecil, Dan Menengah Di Era Digital," *J. Ilmu Ekon.*, vol. 5, no. 1, pp. 44–49, 2024, [Online]. Available: <http://jurnal.unsil.ac.id/index.php/welfare>
- [4] E. A. R. Kasim, Stasiswaty, N. Ransi, and Isnawaty, "Sistem Rekomendasi Produk UMKM Menggunakan Algoritma User-Based Collaborative Filtering Berbasis Website," *J. SISFOTENIKA*, vol. 14, no. 2, pp. 152–162, 2024.
- [5] L. A. M. Ramadhan, Sutardi, and J. Nangi, "PEMBUATAN WEB E-COMMERCE PADA TOKO KENIME STORE MENGGUNAKAN SISTEM REKOMENDASI BERBASIS METODE COLLABORATIVE FILTERING DENGAN ALGORITMA ADJUSTED COSINE SIMILARITY," *semantik*, vol. 3, no. 2, pp. 227–236, 2017.
- [6] M. Burhanudin and D. Hermanto, "Pengembangan Aplikasi Kasir Berbasis Web Dalam Pengelolaan Transaksi Keuangan," *Jatekom (Jurnal Apl. Teknol. dan Komputasi)*, vol. 1, no. 1, pp. 1–12, 2025.
- [7] I. van den Hoven, "Fluxsort: A fast branchless stable quicksort / mergesort hybrid that is highly adaptive," 2022. [Online]. Available: <https://github.com/scandum/fluxsort>
- [8] N. S. Abuba, E. Y. Baagyere, C. I. Nakpih, and J. K. Wiredu, "OptiFlexSort : A Hybrid Sorting Algorithm for Efficient Large-Scale Data Processing," *J. Adv. Math. Comput. Sci.*, vol. 40, no. 2, pp. 67–81, 2025.
- [9] S. Akobre, J. K. Wiredu, I. Aabaah, and U. A. Wumpini, "A Unified Framework for Theoretical and Experimental Evaluation of Classical and Modern Sorting Algorithms in Real-Time Systems," *J. Inf. Syst. Informatics*, vol. 7, no. 4, pp. 3143–3185, 2025, doi: 10.63158/journalisi.v7i4.1287.
- [10] Y. I. Chandra, D. R. Irawati, and M. Riastuti, "Penerapan Model Rational Unified Process (RUP) Dalam Membangun Aplikasi Promosi dan Penjualan Berbasis Web (Studi Kasus: Toko Mie Ayam 'Neng Kene Kiyeh')," *IKRA-ITH Inform. J. Komput. dan Inform.*, vol. 8, no. 3, pp. 107–120, 2024, doi: 10.37817/ikraith-informatika.v8i3.4369.
- [11] M. R. Hermawan, F. S. F. Kusumah, and H. Fajri, "The Development of a Web-Based Point of Sale (PoS) and Stock Management System for Warung IT Mart at the Informatics Engineering Laboratory of Ibnu Khaldun University Bogor," *e-JURNAL Penyelid. DAN Inov.*, vol. 12, no. 5, pp. 166–190, 2025, doi: 10.53840/ejpi.v12i5.317.
- [12] S. Edelkamp and A. Weiß, "BlockQuicksort: How Branch Mispredictions don't affect Quicksort," *arXiv*, pp. 1–23.
- [13] L. M. Wegner, "Quicksort for Equal Keys," *IEEE Trans. Comput.*, vol. C-34, no. 4, pp. 362–367, 1985.
- [14] D. J. Mankowitz *et al.*, "Faster sorting algorithms discovered using deep reinforcement learning," *Nature*, vol. 618, pp. 1–18, 2023, doi: 10.1038/s41586-023-06004-9.