

Sistem Informasi Kasir Toko Kembar menggunakan Framework Webix dan MongoDB

Randy Petra Kamaz ^{1*}, Yos Richard Beeh ²

^{1*,2} Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Kota Salatiga, Provinsi Jawa Tengah, Indonesia.

Corresponding Email: randykamaz@gmail.com ^{1*}

Histori Artikel:

Dikirim 8 Oktober 2024; *Diterima dalam bentuk revisi* 11 November 2024; *Diterima* 20 November 2024; *Diterbitkan* 10 Januari 2025. Semua hak dilindungi oleh Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) STMIK Indonesia Banda Aceh.

Abstrak

Toko Kembar adalah toko yang menyediakan berbagai barang kebutuhan sehari-hari. Proses transaksi penjualan di Toko Kembar masih dilakukan secara manual meskipun jumlah transaksinya cukup besar. Petugas toko harus memeriksa harga barang di catatan, menghitung total, dan menulis struk pembelian secara manual. Aktivitas ini memerlukan waktu lama, sering kali menyebabkan antrean panjang pembeli. Pencatatan stok barang dan riwayat transaksi juga masih dilakukan secara manual di buku, sehingga rentan terhadap kehilangan atau kerusakan data. Penelitian ini bertujuan untuk merancang dan mengembangkan aplikasi kasir berbasis website untuk mengatasi permasalahan tersebut. Metode penelitian yang digunakan adalah Research and Development (R&D). Penelitian ini menghasilkan aplikasi kasir dengan fitur menu transaksi, menu riwayat transaksi, dan menu data barang. Hasil pengujian menunjukkan bahwa aplikasi kasir ini dapat mempermudah proses transaksi serta pengelolaan data di Toko Kembar.

Kata Kunci: Aplikasi Kasir; Website; Webix; Flask; MongoDB.

Abstract

Toko Kembar is a store providing various daily necessities. The sales transactions at Toko Kembar are still conducted manually, despite the high number of transactions. Store staff must manually check item prices on records, calculate totals, and write purchase receipts. This process is time-consuming, often leading to long queues of buyers. Stock and transaction history records are also manually documented in books, making them susceptible to loss or damage. This research aims to design and develop a web-based cashier application to address these issues. The research method employed is Research and Development (R&D). This study produced a cashier application with transaction menu, transaction history menu, and item data menu features. Testing results indicate that the application effectively simplifies the transaction process and data management at Toko Kembar.

Keyword: Cashier Application; Website; Webix; Flask; MongoDB.

1. Pendahuluan

Perkembangan teknologi informasi yang pesat memberikan dampak signifikan pada berbagai sektor kehidupan, terutama pada sektor bisnis *retail*. Bisnis *retail* merupakan jenis usaha yang menjual produk secara langsung kepada konsumen akhir untuk memenuhi kebutuhan mereka (Shantilawati *et al.*, 2024). Dalam sektor ini, layanan kepada pelanggan menjadi aspek yang sangat krusial. Layanan merupakan aktivitas yang terjadi melalui interaksi langsung antara individu atau perangkat fisik dengan tujuan untuk menjamin kepuasan pelanggan (Barata, 2003). Kecepatan layanan dalam proses transaksi dapat meningkatkan kenyamanan dan kepuasan pelanggan (Yessayabella & Adys, 2021). Kemajuan teknologi saat ini memungkinkan terciptanya layanan yang lebih mudah dan cepat (Cahyono & Jayanti, 2022). Oleh karena itu, peran teknologi menjadi sangat penting bagi toko *retail* seperti Toko Kembar yang masih menjalankan operasionalnya secara manual.

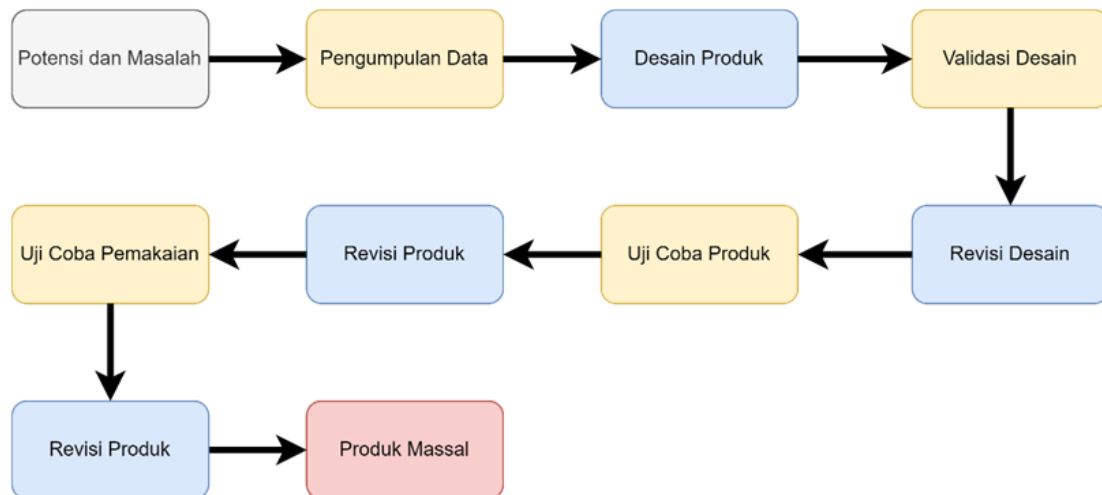
Toko Kembar adalah penyedia berbagai barang kebutuhan sehari-hari dengan lebih dari 12 jenis produk yang dijual, seperti makanan, minuman, perlengkapan mandi, pulsa, kuota internet, pembayaran listrik, PDAM, dan lain-lain. Jumlah rata-rata penjualan mencapai 200 transaksi setiap hari, dengan potensi peningkatan di masa depan. Namun, proses transaksi penjualan di Toko Kembar masih bersifat manual dan kurang efisien. Pemilik atau petugas toko harus memeriksa harga barang pada buku catatan atau rak display, menghitung total pembelian, dan menulis nota secara manual. Proses ini membutuhkan waktu yang lama, terutama karena hanya ada dua petugas toko. Situasi ini sering menyebabkan antrean panjang, terutama saat toko ramai, dan berisiko menyebabkan kesalahan dalam pencatatan serta penghitungan nota, yang dapat merugikan baik pihak toko maupun pelanggan. Pencatatan stok barang dan riwayat transaksi juga dilakukan secara manual di buku, yang rentan terhadap kehilangan atau kerusakan data. Hal ini berpotensi menimbulkan dampak serius pada manajemen keuangan toko, bahkan dapat mengarah pada risiko kebangkrutan. Di sisi pelanggan, masalah ini dapat menyebabkan ketidaknyamanan, kerugian material, dan menurunnya kepercayaan terhadap toko.

Penelitian sebelumnya membahas pengembangan sistem informasi kasir berbasis web sebagai solusi untuk menciptakan catatan penjualan yang sistematis. Sistem ini memungkinkan pengguna mengakses aplikasi dari berbagai perangkat elektronik. Hasil penelitian menunjukkan bahwa sistem informasi kasir tersebut dapat digunakan untuk melihat stok barang, menyimpan data barang, menambah data penjualan, dan menghasilkan laporan penjualan secara sistematis (Kristianto *et al.*, 2023). Penelitian lain juga membahas pengembangan sistem informasi kasir berbasis web untuk mengatasi pencatatan manual. Sistem ini dibuat menggunakan bahasa pemrograman PHP dengan MySQL sebagai server basis data. Hasil penelitian menunjukkan bahwa sistem ini mempermudah transaksi, mempercepat pembuatan laporan, dan meningkatkan layanan (Anwari *et al.*, 2020). Penelitian lainnya mengembangkan aplikasi kasir untuk mencatat transaksi penjualan dan pembelian, memantau stok barang, serta mencatat pengeluaran operasional. Sistem ini dibangun menggunakan PHP, Bootstrap, dan MySQL, yang memungkinkan pemilik toko memahami kondisi laba atau rugi toko dengan lebih baik (Ongsa & Ciang, 2021).

Berdasarkan permasalahan yang dihadapi Toko Kembar dan kajian dari penelitian terdahulu, diperlukan sebuah aplikasi kasir berbasis web untuk mempermudah proses transaksi dan pengelolaan data barang. Sistem ini diharapkan dapat mengurangi kesalahan perhitungan dan risiko kehilangan catatan, sehingga meningkatkan efisiensi dan layanan toko. Penelitian ini bertujuan untuk merancang dan membangun aplikasi kasir berbasis web sebagai solusi untuk Toko Kembar. Inovasi utama dari penelitian ini terletak pada teknologi yang digunakan dalam pengembangan sistem. Aplikasi ini diharapkan mampu meningkatkan kepercayaan pelanggan, mengurangi risiko kebangkrutan, serta memperkuat citra dan daya tarik toko.

2. Metode Penelitian

Metode penelitian yang digunakan pada penelitian ini *Research and Development (R&D)*. *Research and Development* adalah sebuah metode penelitian yang bertujuan untuk menciptakan produk dan menguji tingkat keefektifannya melalui serangkaian tahapan penelitian (Sugiyono, 2009). Adapun tahapan penelitian yang dilakukan seperti pada gambar.



Gambar 1. Tahapan Penelitian R&D

Sumber: Sugiyono, 2009

Tahap pertama adalah menemukan potensi dan masalah yang ada. Pada tahap ini dilakukan observasi dan wawancara secara langsung dengan pemilik toko untuk mengetahui masalah yang sedang dialami. Pengumpulan data dilakukan dengan penelusuran teori melalui buku, jurnal, internet atau wawancara dengan pihak terkait dalam merancang aplikasi kasir (Sugiyono, 2009). Data yang dikumpulkan berupa data barang, data transaksi penjualan, dan data pendukung lainnya yang dapat diimplementasikan ke dalam aplikasi kasir. Aplikasi kasir yang dikembangkan untuk menyelesaikan masalah ini memanfaatkan beberapa teknologi antara lain *framework* Webix, Flask, dan MongoDB. Webix sebagai sebuah *library User Interface (UI)* yang mendukung pembuatan tampilan *website* dengan berbagai komponen, *widget*, dan kontrol UI lintas *platform browser* (Webix, n.d.). Penggunaan Webix memudahkan pengembang membuat tampilan aplikasi *web* kompleks dengan penggunaan kode yang lebih sedikit (Nugroho & Susetyo, 2022). Flask sebagai *microframework* berbasis Python yang mempermudah pembuatan dan pengelolaan fungsionalitas *website* sesuai kebutuhan, serta memungkinkan terciptanya struktur aplikasi yang lebih teratur (Irsyad, 2018; Yulius & Susetyo, 2023). MongoDB merupakan salah satu *database* NoSQL (Not Only SQL) bersifat dokumen (Yulius & Susetyo, 2023). MongoDB menyimpan data dalam dokumen JSON, sehingga lebih ringan digunakan (Suhartini & Purbasari, 2023). MongoDB menggunakan konsep *key-value*, sehingga membuat MongoDB menjadi sistem basis data yang sangat cepat dibandingkan dengan konsep *non key-value* seperti *relational database* (Kurniawan & Kurniati, 2022; Yulius & Susetyo, 2023).

Tahap desain produk dilakukan dengan menggunakan *Unified Modelling Language (UML)* untuk memodelkan kebutuhan dalam sistem yang dapat memudahkan proses pengembangan sistem (Hermawan & Fauzi, 2021). UML pada penelitian ini diwujudkan melalui *use case diagram*, *activity diagram*, dan *class diagram*. Desain yang dibuat akan dikoordinasikan atau divalidasi oleh pihak Toko Kembar. Revisi desain dilakukan jika terdapat kekurangan berdasarkan hasil validasi desain.

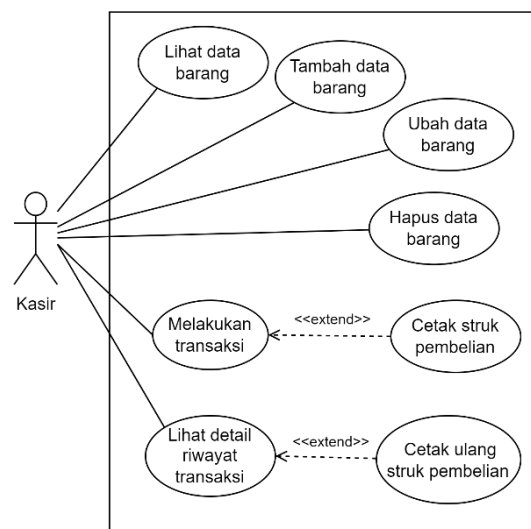
Produk atau aplikasi yang dibuat akan dilakukan pengujian awal oleh *developer* untuk mendeteksi *bug/error*. Revisi produk akan dilakukan untuk menyelesaikan *error* yang sudah ditemukan. Aplikasi yang sudah sesuai akan dilakukan uji coba pemakaian oleh pengguna yaitu pihak Toko Kembar. Wawancara dengan pihak Toko Kembar juga akan dilakukan. Wawancara ini bertujuan untuk mendapatkan umpan balik mengenai kesesuaian aplikasi dengan kebutuhan yang dihadapi. Revisi akan dilakukan berdasarkan umpan balik pengguna jika fitur di dalam aplikasi kurang sesuai.

3. Hasil dan Pembahasan

3.1 Hasil

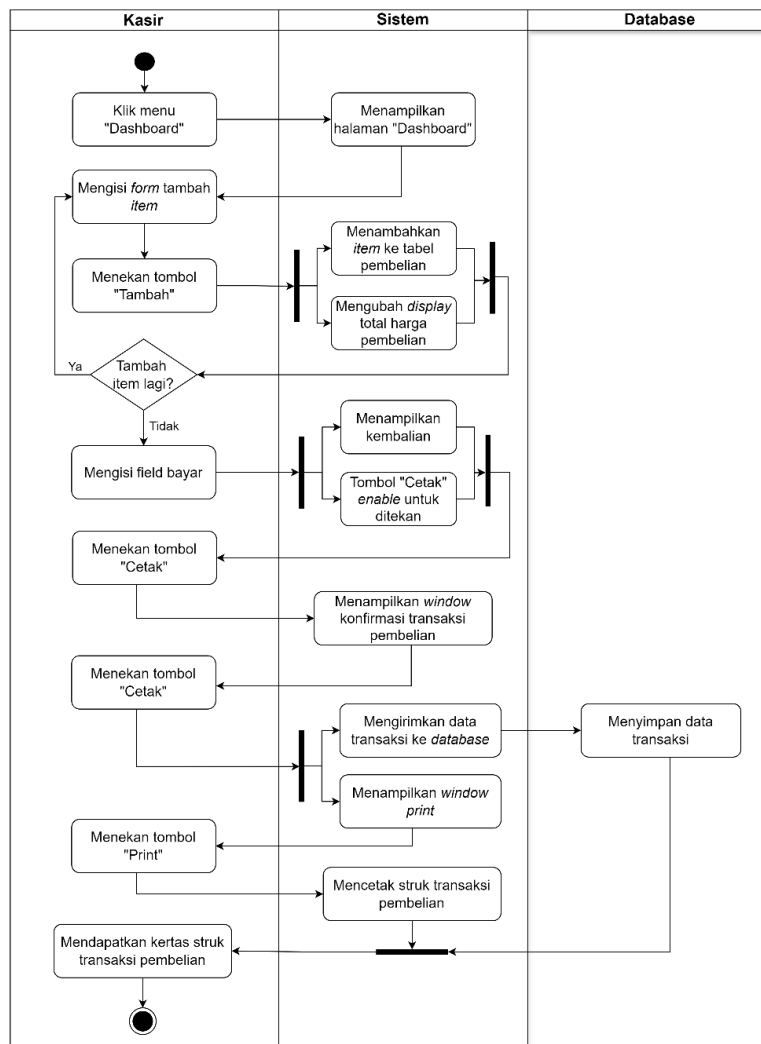
Hasil penelitian meliputi analisis kebutuhan sistem, implementasi teknis dari aplikasi yang dikembangkan, serta evaluasi melalui pengujian untuk menilai kesesuaian sistem dengan kebutuhan pengguna.

3.1.1 Analisis Kebutuhan



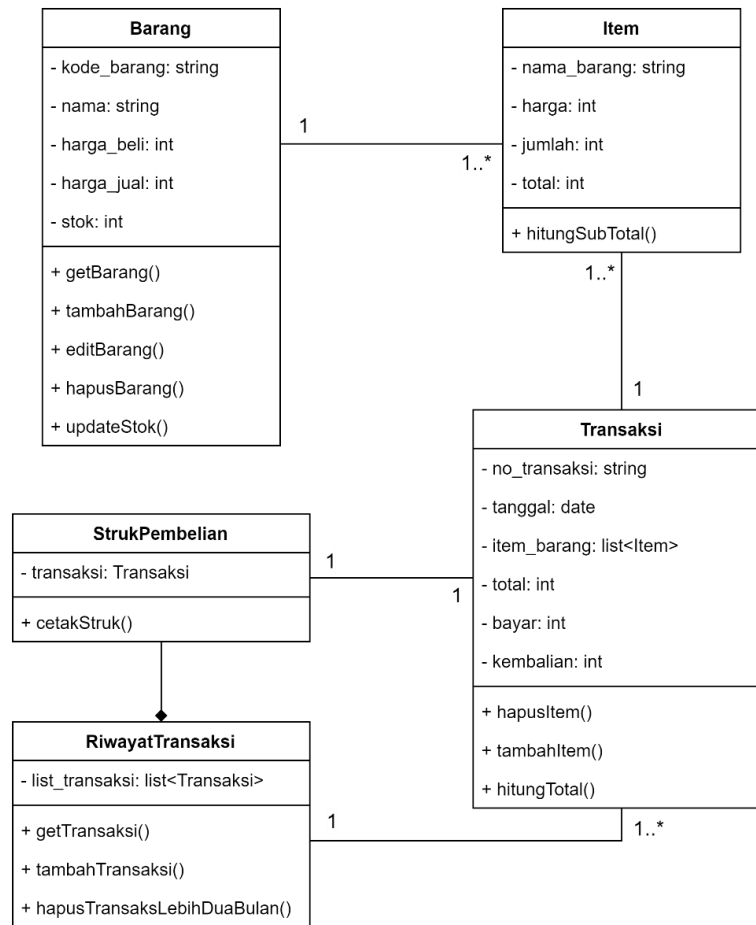
Gambar 2. Use Case Diagram

Gambar 2 merupakan diagram yang menunjukkan interaksi antara aktor dan fungsionalitas sistem. Aktor yang terlibat dalam sistem adalah kasir. Kasir memiliki hak untuk melakukan atau membuat transaksi penjualan, mencetak struk pembelian, melihat riwayat transaksi, mencetak ulang struk pembelian, dan mengelola data barang seperti melihat/memantau, menambah, mengubah, dan menghapus data.



Gambar 3. Activity Diagram Proses Transaksi

Gambar 3 merupakan diagram yang menunjukkan aktivitas kasir dalam melakukan proses transaksi pada *website* kasir. Aktivitas kasir dimulai dengan menekan tombol menu *Dashboard*, lalu sistem akan mengarahkan ke halaman *Dashboard*. Kasir mengisi *form* untuk menambahkan barang (*item*) yang akan dibeli ke dalam tabel pembelian. Tampilan total harga pembelian akan berubah seiring dengan penambahan atau pengurangan barang pada tabel pembelian. Kasir mengisi *field* "Bayar" sesuai total harga pembelian setelah selesai menambahkan semua barang yang dibeli ke dalam tabel pembelian. Jumlah kembalian akan muncul dan tombol "Cetak" menjadi *enable* (bisa ditekan) jika *input* bayar sesuai. Kasir menekan tombol "Cetak" jika sudah selesai melakukan transaksi. Sistem akan mencetak struk pembelian dan data transaksi akan disimpan ke dalam *database*. Pada tahap akhir, kasir mendapatkan hasil cetak kertas struk pembelian yang akan diberikan kepada pembeli.

Gambar 4. *Class Diagram*

Gambar 4 merupakan diagram yang menunjukkan hubungan dari kelas-kelas yang ada di dalam aplikasi kasir. Pada *class diagram* tersebut, terdapat lima kelas dalam aplikasi kasir yaitu **Barang**, **Item**, **Transaksi**, **RiwayatTransaksi**, dan **StrukPembelian**. Hubungan antar kelas dalam diagram menggambarkan alur sistematis dari pengelolaan barang hingga pencatatan transaksi dan pembuatan struk pembelian.

3.1.2 Implementasi Sistem

Implementasi sistem mencakup pengembangan fitur pada menu Dashboard, menu Riwayat, dan menu Data Barang. Penjelasan disertai dengan kode program dan diagram untuk memberikan gambaran rinci tentang cara kerja setiap fitur serta integrasi dengan teknologi seperti Webix, Flask, dan MongoDB. Setiap fitur dirancang untuk meningkatkan efisiensi dan akurasi dalam proses transaksi serta pengelolaan data barang.

Kode Program 1. Pembuatan Tabel Pembelian Pada Menu *Dashboard*

```

01 {
02   responsive: "transaction_section",
03   id: "transaction table",
04   view: "datatable",
05   css: "webix_header_border webix_data_border",
06   columns: [
07     {id: "no", header: "No", width: 40},
08     {id: "kode_barang", header: "Kode Barang", fillspace: 1},
09     {id: "nama", header: "Nama Barang", fillspace: 4},
10     {
11       id: "harga_jual", header: "Harga", fillspace: 2,
12       css: "text_right", template: currencyHargaJual
13     },
14     {
15       id: "total", header: "Total", fillspace: 2,
16       math: "[$r, harga_jual] * [$r, jumlah_barang]",
17       css: "text_right", template: currencyTotal
18     },
19     {
20       id: "aksi", header: "",
21       template: "{common.trashIcon()}", css: "text_center"
22     }
23   ],
24   math: true,
25   onClick: {"wxi-trash": deleteDatatable},
26   on: {
27     onAfterRender: updateTotal,
28   }
29 }
30 }

```

Kode Program 1 merupakan kode untuk pembuatan tabel pembelian yang ada di halaman *Dashboard*. Sintaks “view” digunakan untuk mendefinisikan komponen Webix apa yang dibuat, misalnya *datatable*, *list*, atau *button*. Baris 6-27 merupakan kode untuk inisialisasi dan konfigurasi kolom di dalam tabel. Sintaks “math” bernilai *true* jika pada *datatable* terdapat penggunaan rumus matematika seperti baris 20 yang memiliki rumus menghitung sub total harga tiap barang. *Datatable* ini juga memiliki *event listener* yang dijalankan pada kondisi tertentu seperti yang ditunjukkan pada baris 33 dan 35.

Gambar 5. Tampilan Halaman *Dashboard*

Gambar 5 merupakan tampilan menu *Dashboard*. Kasir akan langsung masuk ke halaman *Dashboard* ketika mengakses *website* kasir. Pada halaman ini, kasir bisa membuat transaksi serta

mencetak struk bukti pembelian yang akan diserahkan kepada pembeli. Transaksi yang berhasil akan tersimpan di dalam *database*.

Kode Program 2. Fungsi *Print* Struk Pembelian

```
01 function printReceipt() {
02     let receiptTemplate = $$("receipt_transaction").$view.innerHTML;
03     let printWindow = window.open('', '', 'height=600, width=800');
04     printWindow.document.write(`
05         <html>
06             ${headerReceipt}
07             <body>
08                 ${headerReceipt}
09                 ${receiptTemplate}
10                 ${footerReceipt}
11             </body>
12         </html>
13     `);
14     printWindow.document.close();
15     printWindow.print();
16 }
```

Kode Program 2 merupakan kode untuk mencetak struk pembelian. Fungsi *receiptTemplate* digunakan untuk menampung konten HTML yang ada di elemen dengan ID bernama “*receipt_transaction*” (detail transaksi pada Gambar 6). Kode pada baris 3 berguna untuk membuka *window* baru yang kosong. *Window* tersebut lalu diisi dokumen HTML struk pembelian seperti pada baris 4 sampai 13. *Window* akan ditutup dan dokumen HTML struk pembelian tersebut akan dicetak.

MENU RIWAYAT TRANSAKSI				
Filter Bulan	Cari No.Struk			
11	May, 2024 14:34	Pendapatan	12.000	Laba 2.000
11	May, 2024 14:27	Pendapatan	36.000	Laba 6.000
24	Apr, 2024 15:17	Pendapatan	187.000	Laba 27.000
16	Apr, 2024 17:29	Pendapatan	99.000	Laba 15.000
	Apr, 2024 20:18	Pendapatan		
Total Pendapatan			454.000	Total Laba 69.000

Detail Transaksi

16-04-2024 TK240416172931

17:29

Ultra Milk 1L 40.000

2 x 20.000

Ice Cream Vanilla 24.000

2 x 12.000

Roma Kelapa Vanilla 24.000

2 x 12.000

Indomilk 250ml 11.000

2 x 5.500

Total 99.000

Bayar 100.000

Kembalian 1.000

Gambar 6. Tampilan Halaman Riwayat

Gambar 6 merupakan tampilan dari menu Riwayat. Pada menu ini kasir bisa melihat *list* transaksi apa saja yang sudah dilakukan beserta detailnya. *List* transaksi berisi tanggal transaksi, nomor transaksi, pendapatan, dan laba setiap transaksi. Bagian bawah *list* transaksi terdapat total pendapatan dan total laba sesuai data yang muncul di *list* transaksi. Riwayat transaksi yang sudah lebih dari dua bulan akan otomatis terhapus dari *database*. Kasir bisa melakukan *filter* dan cari data transaksi. *Filter* dilakukan dengan memilih bulan dan tahun, sedangkan pencarian data dilakukan dengan mengisi nomor transaksi yang ingin dicari. Pada bagian detail transaksi terdapat tampilan struk pembelian pada transaksi yang dipilih dan dua tombol. Tombol tersebut adalah tombol mencetak ulang struk pembelian dan tombol menutup tampilan detail transaksi.

Kode Program 3. Fungsi Tambah Data Barang

```

01 @app.route('/barang/tambah-barang', methods=['POST'])
02 def tambah_barang():
03     try:
04         kode_barang = request.form['kode_barang']
05         existing_data = collection_barang.find_one({
06             'kode_barang': kode_barang
07         })
08
09         if existing_data:
10             return jsonify({'status': 'error',
11                             'message': 'Kode barang sudah digunakan'})
12
13         new_data = {
14             'kode_barang': request.form['kode_barang'],
15             'nama': request.form['nama'],
16             'harga_beli': int(request.form['harga_beli']),
17             'harga_jual': int(request.form['harga_jual']),
18             'stok': int(request.form['stok'])
19         }
20
21         collection_barang.insert_one(new_data)
22         return jsonify({'status': 'success',
23                         'message': 'Data barang berhasil ditambahkan'})
24     except Exception as e:
25         print({'status': 'error', 'message': str(e)})
26         return jsonify({'status': 'error',
27                         'message': 'Gagal menambahkan data barang'})

```

Kode Program 3 merupakan kode untuk menambah data barang. Pada baris 1 terdapat kode untuk inisialisasi *route* dan *method*-nya. Ketika *route* dipanggil, maka akan menjalankan fungsi “tambah_barang” pada baris 2. Fungsi “tambah_barang” berisi kode untuk menyimpan data barang ke dalam *database*. Baris 21 merupakan *command* MongoDB untuk menambahkan data ke *database*. Pada kode ini juga terdapat *error handling* yang berfungsi ketika terjadi *error* di dalam blok kode “try”, sistem tetap berjalan dengan mengeksekusi kode yang berada dalam blok “except”.

Kode Program 4. Fungsi Ubah Data Barang

```

01 @app.route('/barang/ubah-barang', methods=['PUT'])
02 def ubah_barang():
03     try:
04         kode_barang = request.form['kode_barang']
05         updated_data = {
06             'nama': request.form['nama'],
07             'harga_beli': int(request.form['harga_beli']),
08             'harga_jual': int(request.form['harga_jual']),
09             'stok': int(request.form['stok'])
10         }
11
12         result = collection_barang.update_one(
13             {'kode_barang': kode_barang}, {'$set': updated_data})
.....
21     except Exception as e:
.....

```

Kode Program 4 merupakan kode untuk mengubah data barang. Baris 12 dan 13 merupakan *command* MongoDB untuk mengubah data barang yang sudah disimpan di dalam *database*. Pada baris 13, MongoDB akan mengubah data barang berdasarkan kode barang yang sesuai. “Kode barang” menjadi *primary key* dan harus bersifat *unique*, artinya setiap data barang harus memiliki kode barang yang berbeda atau tidak boleh ada yang sama.

Kode Program 5. Fungsi Hapus Data Barang

```

01 @app.route('/barang/hapus-barang', methods=['DELETE'])
02 def hapus_barang():
03     try:
04         kode_barang = request.form['kode_barang']
05         collection_barang.delete_one({'kode_barang': kode_barang})
06         return jsonify({'status': 'success',
07             'message': 'Data barang berhasil dihapus'})
08     except Exception as e:
09         .....

```

Kode Program 5 merupakan kode untuk menghapus data barang yang ada di *database*. Baris 5 merupakan *command* MongoDB untuk menghapus data barang di dalam *database* berdasarkan kode barang yang sesuai.

No	Kode	Nama Barang	Harga Beli	Harga Jual	Stok
1	B255	Roma Kelapa Vanilla	10.000	12.000	1
2	V255	Ice Cream Vanilla	9.000	12.000	6
3	L900	Aqua Botol 250ml	6.000	7.000	4
4	P800	Oreo	8.000	10.000	8
5	W300	Indomie Goreng	5.000	5.500	4
6	Z300	Indomilk 250ml	5.000	5.500	4
7	C100	You C-100	6.000	7.000	1
8	A123	Ultra Milk 1L	18.000	20.000	3
9	D600	Tisu	5.000	6.000	1

Gambar 7. Tampilan Halaman Data Barang

Gambar 7 merupakan tampilan dari menu Data Barang. Pada menu ini kasir bisa mencari data barang dan *filter* stok barang. Kasir juga bisa melihat, menambahkan, mengubah, dan menghapus data barang. Pada tabel barang, kasir bisa mengurutkan data berdasarkan kolom yang ingin dijadikan acuan dalam pengurutan. Data bisa diurutkan secara *ascending* ataupun *descending*.

3.1.3 Pengujian Sistem

Pengujian dilakukan dengan melakukan wawancara kepada pihak Toko Kembar. Wawancara tersebut dilakukan untuk mengetahui apakah aplikasi kasir yang dibuat sudah sesuai dengan kebutuhan dan mampu menyelesaikan masalah yang terjadi. Hasil yang diperoleh dari wawancara adalah pengguna merasa dengan adanya aplikasi kasir dapat memudahkan proses transaksi penjualan. Pihak Toko Kembar merasa lebih cepat dalam melakukan proses transaksi penjualan karena total belanja pembeli dapat dihitung secara otomatis oleh sistem dan sistem juga dapat mencetak struk pembelian. Pihak Toko Kembar bisa mengetahui pendapatan dan laba tiap bulan karena adanya menu riwayat transaksi dalam aplikasi kasir. Pihak Toko Kembar juga merasa bahwa aplikasi kasir memudahkan dalam mengelola data barang. Pihak Toko Kembar bisa melihat stok barang yang hampir habis dengan adanya fitur *filter* stok barang. Pihak Toko Kembar juga bisa menambahkan data barang, menghapus data barang, dan mengubah data barang seperti stok barang, harga beli, dan harga jual.

3.2 Pembahasan

Penelitian menghasilkan aplikasi kasir berbasis web yang dirancang untuk meningkatkan efisiensi operasional Toko Kembar dalam transaksi penjualan dan pengelolaan data barang. Sistem manual sebelumnya mengakibatkan masalah seperti antrean panjang, kesalahan pencatatan, dan pengelolaan

stok yang tidak optimal. Permasalahan tersebut sejalan dengan penelitian Anwari *et al.* (2020), yang mengidentifikasi inefisiensi dan risiko kehilangan data pada sistem manual. Selain itu, efisiensi layanan terbukti menjadi faktor penting dalam meningkatkan pengalaman pelanggan sebagaimana dijelaskan oleh Barata (2003). Aplikasi ini memanfaatkan teknologi modern, yaitu Webix, Flask, dan MongoDB, untuk menciptakan solusi yang efisien. Webix digunakan untuk membangun antarmuka pengguna yang responsif dan terstruktur (Nugroho & Susetyo, 2022), Flask sebagai *microframework* Python memberikan fleksibilitas dalam pengelolaan fungsi aplikasi (Irsyad, 2018), dan MongoDB menawarkan kecepatan tinggi dalam pengolahan data dengan basis data berbasis dokumen (Suhartini & Purbasari, 2023).

Fitur utama yang diterapkan mencakup *Dashboard*, Riwayat Transaksi, dan Manajemen Data Barang. *Dashboard* memungkinkan kasir melakukan transaksi, mencetak struk, dan menyimpan data transaksi secara otomatis, yang mendukung efisiensi operasional sebagaimana ditunjukkan oleh Cahyono & Jayanti (2022). Fitur Riwayat Transaksi memfasilitasi pemantauan data transaksi, termasuk pendapatan dan laba per transaksi, yang relevan dengan pandangan Ongsa & Ciang (2021) tentang pentingnya laporan transaksi berbasis data. Fitur Manajemen Data Barang memungkinkan penambahan, pengubahan, dan penghapusan data barang dengan filter stok yang mempermudah pengelolaan inventaris secara terorganisasi, mendukung pendekatan pengelolaan inventaris yang terarah seperti yang dijelaskan oleh Pramudita & Abidin (2024).

Pengujian sistem menunjukkan bahwa aplikasi yang dikembangkan mampu memenuhi kebutuhan pengguna. Kasir dapat melakukan transaksi dengan cepat berkat perhitungan otomatis dan pencetakan struk langsung melalui sistem. Pencatatan data barang memberikan fleksibilitas dalam memantau stok dan memperbarui informasi barang, yang konsisten dengan temuan Kristianto *et al.* (2023) bahwa aplikasi berbasis web dapat mengurangi beban kerja manual dan meningkatkan akurasi data. Dibandingkan dengan penelitian sebelumnya yang menggunakan PHP dan MySQL (Anwari *et al.*, 2020; Ongsa & Ciang, 2021), aplikasi ini lebih unggul dengan memanfaatkan MongoDB untuk pengelolaan data yang cepat dan fleksibel, serta Webix untuk antarmuka pengguna yang intuitif (Nugroho & Susetyo, 2022). Meskipun aplikasi telah berhasil memenuhi kebutuhan utama, terdapat ruang untuk pengembangan lebih lanjut. Integrasi dengan perangkat keras seperti *barcode scanner* dan sistem pelaporan yang lebih rinci dapat memberikan nilai tambah. Pendekatan ini relevan dengan pandangan Hermawan & Fauzi (2021), yang menekankan pentingnya fleksibilitas sistem untuk mendukung pertumbuhan bisnis. Aplikasi kasir berbasis web ini membuktikan efektivitasnya dalam menyelesaikan masalah operasional di Toko Kembar dan menawarkan potensi untuk diterapkan pada bisnis *retail* lainnya dengan kebutuhan serupa.

4. Kesimpulan

Pembangunan aplikasi kasir dimulai dengan mendesain produk/sistem sesuai dengan kebutuhan. Implementasi desain dilakukan setelah desain yang dibuat sudah sesuai. Implementasi dilakukan dengan membuat tampilan aplikasi, membuat fungsi-fungsi, dan mengintegrasikan aplikasi dengan *database*. Pengujian dilakukan setelah proses implementasi selesai dilakukan. Hasil pengujian akan menentukan apakah aplikasi kasir sudah sesuai kebutuhan. Berdasarkan hasil pengujian yang dilakukan, maka dapat disimpulkan bahwa aplikasi kasir mampu menyelesaikan masalah di Toko Kembar. Aplikasi kasir yang dibuat mampu mempermudah transaksi penjualan. Pengelolaan data barang juga mudah dilakukan dengan adanya fitur tambah, ubah, dan hapus data barang. Pengguna bisa memantau stok barang dengan mudah dengan adanya *filter* stok barang. Aplikasi kasir juga mampu mencatat riwayat transaksi dan mencetak struk pembelian.

5. Daftar Pustaka

- Anwari, V. B., Ferdiansyah, F., & Samsinar, S. (2020, August). Implementasi Sistem Informasi Kasir Pada Rakab Mercon Berbasis Web. In *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)* (Vol. 4, No. 3, pp. 001-008). <https://doi.org/10.29407/inotek.v4i3.24>.
- Barata, A. A. (2003). *Dasar-dasar pelayanan prima*. Elex Media Komputindo.
- Cahyono, D. E., & Jayanti, A. (2022). Implementasi Aplikasi Kasir Berbasis Web Pada Toko Ghafya Fruits Shop. *Jurnal Ekonomi Dan Teknik Informatika*, 10(1), 32-40.
- Hermawan, R., & Fauzi, A. (2021). Perancangan Sistem Informasi Kasir Penjualan Barang Berbasis Website Metode Spiral Toko Warna. *Jurnal SIFO Mikroskil*, 22(2), 101-114.
- Irsyad, R., 2018. Penggunaan Python Web Framework Flask Untuk Pemula. <https://doi.org/10.31219/osf.io/t7u5r>.
- Kristianto, A., Sena, I. G. W., & Julianto, R. (2023). Perancangan dan Pembuatan Sistem Informasi Kasir Berbasis Web pada Toko XYZ. *KONSTELASI: Konvergensi Teknologi dan Sistem Informasi*, 3(1), 184-196. <https://doi.org/10.24002/konstelasi.v3i1.7131>
- Kurniawan, D., & Kurniati. (2022). Penerapan MongoDB pada sistem informasi management academic SDN 13 Banyuasin. *Seminar Hasil Penelitian Vokasi (SEMHLAVOK)*.
- Nugroho, C., & Susetyo, Y. A. (2022). Implementation of Webix dynamic scripting for web-based form app development as an alternative to data-collecting applications. *Jurnal Teknik Informatika (JUTIF)*, 3(2), 387–394. <https://doi.org/10.20884/1.jutif.2022.3.2.212>
- Ongsa, R., & Ciang, K. (2021). Implementasi Sistem Kasir untuk Toke POK. *Computer Technology and Information Systems*, 5(2), 39-44.
- Pramudita, R. A. (2024). Cashier App Revolutionizes Internal Controls in Global Enterprises. *Indonesian Journal of Law and Economics Review*, 19(2), 6-17. <https://doi.org/10.21070/ijler.v19i2.1097>
- Shantilawati, I., Zebua, S., & Tarmizi, R. (2024). Penggunaan Digital Marketing Dalam Meningkatkan Penjualan Bisnis Retail. *Jurnal Manajemen Retail Indonesia*, 5(1), 30-37.
- Sugiyono, D. (2013). Metode penelitian pendidikan pendekatan kuantitatif, kualitatif dan R&D.
- Purbasari, Y. (2023). Analisis Fitur Dan Keunggulan Database Mongo dalam Berbagai Bahasa Pemrograman. *JSK (Jurnal Sistem Informasi dan Komputerisasi Akuntansi)*, 7(1), 7-13. <https://doi.org/10.56291/jsk.v7i1.122>.
- Webix. (2024). Webix: JavaScript UI library; HTML5 JS framework, 102 UI web widgets. Retrieved from <https://webix.com/>
- Yessayabella, D., & Adys, Y. (2021). Implementasi sistem informasi manajemen kasir berbasis aplikasi moka pos (point of sales) pada kafe x tahun 2022. *Jamanta: Jurnal Mahasiswa Akuntansi Unita*, 1(2), 54-71.

Yulius, & Susetyo, Y. A. (2023). Analisis dan Penerapan Database MongoDB pada Aplikasi Manajemen Dokumen di PT. XYZ. *Jurnal JTIK (Jurnal Teknologi Informasi Dan Komunikasi)*, 7(4), 570–578. <https://doi.org/10.35870/jtik.v7i4.1047>.