

IMPLEMENTASI SISTEM INFORMASI MANAJEMEN PERSEDIAAN RESTORAN BERBASIS WEBSITE DENGAN METODE FIFO DAN REORDER POINT DI D'LISA BEACH

Alfa Rizki^{1*}, Fitri Yanti²

^{1,2}Universitas Pamulang; alamat; Jl. Raya Puspitek No. 46, Tangerang Selatan telp/Fax (021) 7412566

Keywords:

Sistem Informasi
Manajemen; Manajemen
Persediaan; FIFO, *Reorder Point*; Restoran

Correspondent Email:

alfarizki17004@gmail.com

Abstrak. Manajemen persediaan merupakan aspek krusial dalam operasional restoran. Restoran D'Lisa Beach menghadapi kendala pengelolaan pencatatan manual dan tidak ada pemantauan secara *real-time* yang memicu pemborosan persediaan kadaluwarsa serta kekosongan persediaan pada jam sibuk. Berdasarkan kondisi tersebut, Restoran D'Lisa Beach membutuhkan sistem informasi manajemen yang terkomputerisasi. Penelitian ini bertujuan merancang sistem informasi manajemen persediaan website dengan menerapkan metode *First In, First Out* (FIFO) guna rotasi persediaan dan *Reorder Point* (ROP) guna penentuan batas minimum pemesanan ulang otomatis. Pengembangan sistem menggunakan metode *Waterfall* dengan pemodelan UML dan ERD serta dibangun dengan PHP Native dan XAMPP untuk MySQL yang dievaluasi dengan pengujian *black-box* dan *white-box*. Hasil pengujian menunjukkan kalkulasi ROP tervalidasi secara akurat melalui komparasi perhitungan manual, dan implementasi FIFO terbukti efektif memprioritaskan persediaan berdasarkan *Batch ID*. Sistem ini diharapkan dapat meningkatkan efisiensi operasional dan menekan risiko kerugian finansial restoran.



Copyright © 2026 The Author(s),
Published by Jurnal Informatika
dan Teknik Elektro Terapan. This
is an open-access article
distributed under the terms of the
Creative Commons Attribution-
NonCommercial 4.0 International
License (CC BY-NC 4.0).

Abstract. Inventory management is a crucial aspect of restaurant operations. D'Lisa Beach Restaurant faces the constraints of manual record-keeping and lack of real-time monitoring, which leads to wastage of expired inventory and shortages during peak hours. Based on these conditions, D'Lisa Beach Restaurant requires a computerized management information system. This study aims to design a website inventory management information system by implementing the *First In, First Out* (FIFO) method for inventory rotation and the *Reorder Point* (ROP) method for automatic minimum reorder point determination. The system was developed using the *Waterfall* method with UML and ERD modeling and built with PHP Native and XAMPP for MySQL, which was evaluated through *black-box* and *white-box* testing. The test results show that the ROP calculation is accurately validated through a comparison of manual calculations, and the FIFO implementation is proven to be effective in prioritizing inventory based on *Batch ID*. This system is expected to improve operational efficiency and reduce the risk of restaurant financial losses.

1. PENDAHULUAN

Di tengah tingginya persaingan bisnis restoran, manajemen persediaan menjadi senjata utama bagi restoran untuk menjaga

efisiensi operasional dan menghindari kerugian akibat ketidakseimbangan persediaan. Namun tantangan terbesar muncul pada restoran yang proses pengelolaan persediaannya masih

bersifat manual. Penelitian terdahulu menunjukkan bahwa, penggunaan sistem manual sering kali menimbulkan kendala seperti keterlambatan pencatatan dan kesulitan akses informasi secara cepat [1]. Oleh karena itu sistem informasi manajemen yang terkomputerisasi menjadi sebuah kebutuhan untuk meringankan beban kerja dalam pengumpulan dan pengolahan data, sehingga informasi yang dihasilkan lebih akurat dan berguna bagi pengambilan keputusan [2].

Penerapan sistem informasi tersebut saat ini sangat dibutuhkan oleh D'Lisa Beach Restaurant. Kondisi pengelolaan persediaan di restoran ini masih murni manual, mengakibatkan kurangnya efisiensi dan akurasi. Terlebih lagi, operasional pengelolaan persediaan dilakukan langsung oleh staf tanpa adanya admin gudang khusus. Akibatnya, staf sering kali kesulitan dalam menentukan prioritas penggunaan persediaan serta acuan waktu untuk melakukan pemesanan ulang.

Sebagai contoh nyata, restoran sering kali kehabisan persediaan dan harus membuang persediaan yang sudah kadaluwarsa karena persediaan lama tidak terpantau dengan baik karena tidak adanya informasi persediaan secara *real-time* sehingga menimbulkan penurunan kualitas restoran dan kerugian finansial. Berdasarkan kondisi tersebut, sistem informasi manajemen persediaan restoran berbasis website dirancang sebagai solusi.

Agar sistem informasi manajemen dapat berfungsi sesuai dengan kebutuhan operasional, diterapkan integrasi metode FIFO (*First In First Out*) dan *Reorder Point* (ROP). Metode FIFO memastikan penggunaan persediaan secara berurutan berdasarkan waktu kedatangan, sementara ROP menghitung batas minimal persediaan dengan mempertimbangkan penggunaan rata-rata, waktu tunggu, dan stok pengaman, sehingga sistem dapat memberikan notifikasi otomatis untuk restok [3]. Pendekatan serupa dengan tambahan *safety stock* juga berhasil mengurangi pengeluaran berlebih dan lonjakan pesanan serta memastikan ketersediaan barang di perusahaan kuliner [4]. Dengan kombinasi metode FIFO dan ROP tersebut, sistem informasi manajemen persediaan restoran ini diharapkan dapat meningkatkan akurasi, efisiensi, dan kontrol kualitas persediaan, sehingga membantu

restoran dalam mengelola persediaan dengan lebih efisien.

2. TINJAUAN PUSTAKA

2.1 Sistem Informasi Manajemen (SIM)

Sistem informasi manajemen merupakan sistem perencanaan dan pengendalian internal dari suatu bisnis. Sistem ini beroperasi dengan pemanfaatan manusia, dokumen, teknologi, dan prosedur oleh akuntansi manajemen dengan tujuan mendukung akuntansi manajemen dalam menyelesaikan berbagai permasalahan bisnis, seperti penentuan harga pokok produk atau layanan hingga perumusan strategi perusahaan [2].

Sistem Informasi Manajemen menjadi landasan strategis untuk mencapai pengendalian internal yang optimal dalam sebuah entitas bisnis [5]. Dalam konteks operasional, khususnya dalam mengelola inventaris, bahkan di bidang industri jasa seperti restoran, kehadiran SIM berfungsi untuk mempermudah proses kerja. Dengan mengintegrasikan basis data teknologi (misalnya, menggunakan MySQL), beban kerja manual yang rentan kesalahan dapat dialihkan, sehingga karyawan dapat lebih fokus pada tugas yang lainnya.

2.2 Manajemen Persediaan

Manajemen persediaan merupakan sistem-sistem untuk mengelola persediaan, seperti bagaimana barang-barang persediaan dapat diklasifikasikan dan seberapa akurat catatan persediaan dapat dijaga. Sistem ini menjadi titik krusial, terutama dalam sektor pelayanan. Perusahaan seringkali dihadapkan dua kondisi, pengurangan persediaan dapat menghemat pengeluaran, tetapi jika ketersediaan barang tidak terjaga, hal tersebut berpotensi menghentikan proses operasional dan menimbulkan ketidakpuasan pelanggan [6].

2.3 Website

Website adalah sekumpulan halaman web terintegrasi yang dapat diakses secara global menggunakan *web browser*. Dalam pemrograman, platform ini dibangun sebagai solusi digital untuk menampilkan informasi dan memproses tugas-tugas spesifik guna memfasilitasi operasional pengguna. Struktur utama suatu halaman website ditulis

menggunakan standar bahasa *markup* yang secara otomatis diterjemahkan oleh komputer menjadi antarmuka visual yang informatif [7].

2.4 First In, First Out (FIFO)

FIFO (*First In, First Out*) merupakan metode pengelolaan persediaan dimana barang yang pertama kali masuk harus menjadi prioritas untuk dikeluarkan atau digunakan terlebih dahulu [8]. Pada penelitian ini Metode FIFO digunakan sebagai metode pengelolaan persediaan yang mengatur prioritas penggunaan persediaan berdasarkan Batch ID, yang dimana barang yang pertama masuk akan menjadi barang pertama yang keluar. Metode ini efektif dalam mengurangi risiko kadaluarsa atau kerusakan persediaan akibat penyimpanan jangka panjang.

2.5 Reorder Pont (ROP)

Reorder point (ROP) merupakan metode untuk menghitung batas minimal persediaan dengan menggunakan *average rate of usage*, *lead time*, dan *safety stock* yang akan digunakan sebagai acuan untuk menentukan waktu pemesanan ulang persediaan yang akurat. Hal ini menekankan bahwa ROP mampu mengatur proses pemesanan persediaan secara sistematis, memastikan ketersediaan produk secara waktu nyata serta mengurangi potensi kekurangan stok [9].

a. Perhitungan *Safety Stock* dan ROP

Safety stock berfungsi sebagai peredam kepastian permintaan. *Safety stock* dihitung menggunakan pendekatan distribusi normal dengan parameter yang dibutuhkan berupa nilai kemungkinan terjadinya kekurangan (α), nilai (Z) atau *service level* pada distribusi normal, standar deviasi (σ), dan waktu tunggu (L) [10]. Penelitian ini mengadopsi *service level* sebesar 95% ($Z = 1,65$) berdasarkan rujukan optimal dari penelitian sebelumnya [11]. Standar ini digunakan karena terbukti efektif diterapkan pada industri kuliner dalam menekan risiko kekosongan persediaan kritis hingga batas 5%, sekaligus menyeimbangkan efisiensi biaya ketersediaan barang agar tidak terjadi penumpukan persediaan yang berlebih. Berikut rumus *safety stock* yang digunakan :

$$SS = Z \times \sigma \times \sqrt{L} \quad (1)$$

Berdasarkan parameter yang sudah dimiliki, maka dapat di formulasikan ke dalam rumus ROP sebagai berikut :

$$ROP = (D \times L) + SS \quad (2)$$

Keterangan :

SS = *Safety stock*

Z = Nilai tingkat pelayanan (*Service Level*)

σ = Standar deviasi permintaan

L = *Lead time*

D = Permintaan harian rata-rata (unit/hari)

2.6 Pengujian Sistem

Pengujian sistem merupakan tahapan esensial dalam siklus pengembangan perangkat lunak yang berfungsi untuk mengevaluasi kualitas sistem, mengidentifikasi galat (*error*), dan memastikan aplikasi beroperasi sesuai dengan spesifikasi persyaratan yang telah ditetapkan sebelum dirilis kepada pengguna [12]. Dalam praktiknya, evaluasi fungsionalitas sistem umumnya mengadopsi metode *Black-box Testing*. Metode ini memperlakukan sistem sebagai "kotak hitam" yang murni berfokus pada elemen input dan output untuk memverifikasi fungsionalitas, keamanan, serta ketahanan sistem dari sudut pandang pengguna, tanpa perlu meninjau struktur implementasi kode program di dalamnya [12].

Untuk melengkapi pengujian fungsional, analisis mendalam pada sisi kode sumber (*source code*) dilakukan melalui metode *White-box Testing*. Berbeda dengan *black-box*, prosedur ini didasarkan pada pemahaman menyeluruh terhadap alur kontrol dan struktur internal aplikasi guna mengidentifikasi kesalahan implementasi dan logika [12]. Salah satu teknik spesifik yang diterapkan dalam *white-box* adalah pemetaan Basis *Path*. Teknik ini mengevaluasi kompleksitas logika prosedural dengan menyusun *flowgraph*, di mana pernyataan-pernyataan sekuensial dalam kode program dipetakan ke dalam bentuk titik (*node*) dan garis (*edge*) [13].

Tingkat kompleksitas logika di dalam program tersebut selanjutnya diukur secara kuantitatif menggunakan metrik *Cyclomatic Complexity* [14]. Penghitungan matematis ini diformulasikan dengan persamaan :

$$V(G) = N - E + 2 \quad (3)$$

Keterangan :

E = Jumlah *edges* pada *flowgraph*

N = Jumlah *Nodes*/Busur pada *flowgraph*

Hasil dari persamaan tersebut akan menentukan jumlah *Independent Path* atau jalur basis independen dari sebuah algoritma [14]. Pemetaan jalur ini sangat krusial untuk memastikan bahwa setiap pernyataan kondisional di dalam sistem telah tereksekusi dan divalidasi minimal satu kali selama fase pengujian, sehingga alur program dapat dijamin validitasnya [12].

2.7 Waterfall

Waterfall merupakan salah satu metodologi pengembangan perangkat lunak klasik yang diperkenalkan pada tahun 1970-an untuk memberikan struktur yang lebih terorganisir dalam proses perancangan sistem. Pendekatan ini mendasarkan pengembangan perangkat lunak pada serangkaian tahapan yang terurut dan sekuensial, di mana setiap fase harus diselesaikan secara tuntas sebelum melangkah ke tahapan berikutnya. Siklus hidup model ini secara utuh terdiri atas analisis kebutuhan, desain sistem, implementasi, pengujian, serta pemeliharaan [15].

Dalam naskah penelitian ini, penerapan metode *Waterfall* dibatasi hingga tahap pengujian saja, tidak mencakup fase pemeliharaan. Fokus penerapan metode ini mencakup fase analisis kebutuhan untuk mengidentifikasi spesifikasi sistem, dilanjutkan dengan tahap desain sistem, fase implementasi melalui penulisan kode program, serta diakhiri dengan tahap pengujian untuk memastikan kualitas dan fungsionalitas sistem telah sesuai dengan kebutuhan. Pembatasan tahapan ini dilakukan agar pengembangan sistem lebih terarah pada pencapaian tujuan fungsional yang telah ditetapkan dalam ruang lingkup penelitian.

3. METODE PENELITIAN

Penelitian ini menggunakan jenis penelitian terapan dengan menggunakan pendekatan model *waterfall*. Rancangan metodologi ini dirancang untuk membangun sistem informasi manajemen persediaan berbasis website yang mampu mengotomatisasi pengelolaan persediaan restoran.

3.1 Metode Pengumpulan Data

Proses pengumpulan data dilakukan melalui tiga metode guna memastikan rancangan sistem sesuai dengan kebutuhan. Pertama, observasi lapangan di Restoran D’Lisa Beach untuk melihat langsung alur pengelolaan persediaan serta mengidentifikasi kelemahan pada sistem saat ini. Kedua, wawancara dilakukan bersama staf terkait untuk mengetahui lebih jelas mengenai masalah keterlambatan pemesanan persediaan dan sulitnya memantau persediaan secara *real-time*. Ketiga, studi pustakan dilakukan untuk mengumpulkan landasan teori yang relevan mengenai perancangan sistem serta penerapan metode FIFO dan ROP.

3.2 Tahapan Pengembangan Sistem

Pengembangan perangkat lunak dieksekusi secara terstruktur berdasarkan model *waterfall*, yang mencakup tahapan analisa kebutuhan, perancangan sistem, implementasi, dan pengujian.

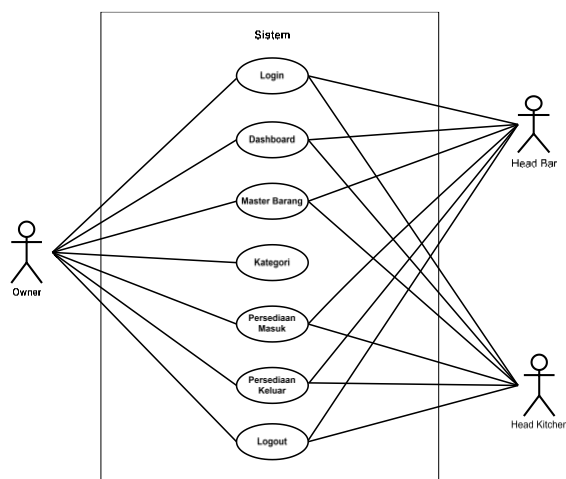
3.2.1 Analisa Kebutuhan Sistem

Berdasarkan hasil observasi dan wawancara, sistem dikembangkan sesuai dengan kebutuhan restoran untuk mengatasi kelemahan operasional seperti informasi *real-time* persediaan, master data, pencatatan persediaan masuk dan persediaan keluar. Tentunya pada pencatatan persediaan harus terintegrasi *Batch ID* untuk penggunaan metode FIFO pada persediaan keluar serta sistem harus memiliki parameter dinamis untuk memicu notifikasi otomatis ketika persediaan sudah mencapai batas minimum (ROP).

3.2.2 Perancangan Sistem

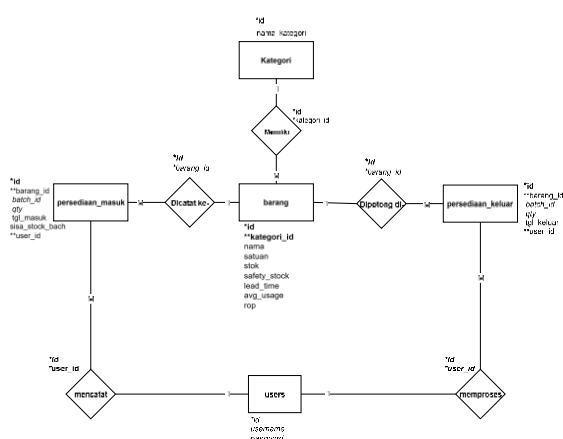
Proses perancangan sistem dilakukan dengan pendekatan menggunakan *Unified Modeling Language* (UML) dan rancangan basis data. Pemodelan fungsionalitas sistem dirancang menggunakan *Use Case Diagram* yang mendefinisikan batasan interaksi antara antarmuka sistem dengan tiga aktor penggunaannya, yaitu *Owner*, *Head Bar*, dan *Head Kitchen*, sebagaimana dengan pendekatan menggunakan *Unified Modeling Language* (UML) dan rancangan basis data. Pemodelan fungsionalitas sistem dirancang menggunakan *Use Case Diagram* yang mendefinisikan batasan interaksi antara antarmuka sistem

dengan tiga aktor penggunaanya, yaitu *Owner*, *Head Bar*, dan *Head Kitchen*, sebagaimana divisualisasikan pada Gambar 1.



Gambar 1. Use Case Diagram

Sementara itu, struktur arsitektur basis data dimodelkan menggunakan *Entity Relationship Diagram* (ERD). Pemodelan ini berfungsi menggambarkan konseptualisasi dan relasi kardinalitas antar entitas – entitas di dalam sistem. Dengan relasi yang terstruktur erat, fungsionalitas algoritma dinamis seperti pembuatan *Batch ID* untuk alur FIFO, serta kalkulasi penyesuaian nilai ROP saat terjadi pergerakan persediaan, dapat tereksekusi secara akurat. Relasi entitas tersebut secara utuh digambarkan pada Gambar 2 berikut.



Gambar 2. Entity Relationship Diagram (ERD)

3.2.3 Implementasi

Rancangan algoritma kemudian ditransformasikan ke dalam baris intruksi program (*source-code*). Pengembangan logika

pemrosesan data (*back-end*) menggunakan bahasa PHP *Native*. Pendekatan ini dipilih guna memberikan fleksibilitas penuh dalam mengintegrasikan metode FIFO dan ROP secara presisi. Tampilan antarmuka dibentuk menggunakan bantuan Bootstrap, sedangkan manajemen basis data dikelola menggunakan *MySQL* dan *XAMPP*.

3.2.4 Pengujian Sistem

Tahapan evaluasi dilakukan melalui skema pengujian berlapis guna memastikan sistem berfungsi sebagaimana mestinya. Pengujian *Black-Box* digunakan untuk meninjau fungsi navigasi antarmuka, pembatasan hak akses (*role*), dan keberhasilan peringatan *error*. Sementara itu, pengujian *White-Box* dengan pendekatan Basis *Path* digunakan untuk menguji struktur kontrol kode internal agar sistem terhindar dari cacat logika (*logic error*). Uji validasi tambahan juga dilakukan dengan membandingkan kalkulasi ROP otomatis di dalam sistem terhadap perhitungan matematis manual.

4. HASIL DAN PEMBAHASAN

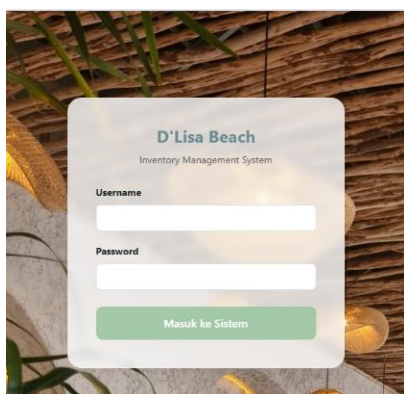
Bagian ini memaparkan hasil implementasi sistem yang telah dianalisa dan dirancang ke dalam bahasa pemrograman agar menjadi sistem utuh yang dapat dioperasikan. Pembahasan difokuskan pada realisasi antarmuka sistem, serta evaluasi fungsionalitas dan validasi akurasi algoritma pemecahan masalah (FIFO dan ROP) pada estoran D’Lisa Beach.

4.1 Implementasi Antarmuka Sistem

Tahap ini merupakan hasil penerjemahan rancangan desain sistem ke dalam bentuk antarmuka website. Berikut merupakan uraian antarmuka sistem yang dikembangkan :

4.1.1 Halaman Login

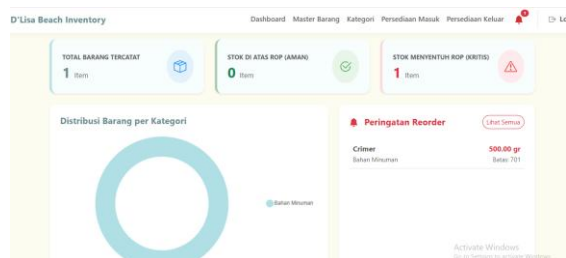
Halaman ini digunakan untuk melakukan validasi kredensial pengguna serta penyesuaian hak akses (*role*) dalam sistem.



Gambar 4. Halaman Login

4.1.2 Halaman Dashboard

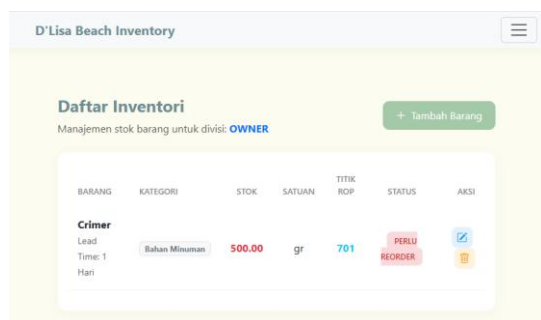
Halaman ini berisi halaman utama dengan ringkasan informasi *real-time* secara instan terkait persediaan, termasuk visualisasi data persediaan dan notifikasi peringatan jika terdapat persediaan yang menyentuh batas kritis (ROP).



Gambar 5. Halaman Dashboard

4.1.3 Halaman Master Data

Halaman ini berisi data persediaan, di mana pengguna dapat mengelola (menambah, edit, hapus) data item berdasarkan hak akses yang dimiliki serta proses inisialisasi parameter persediaan ROP dan *generate Batch ID* persediaan awal.



Gambar 6. Halaman Master Data

4.1.4 Halaman Kategori

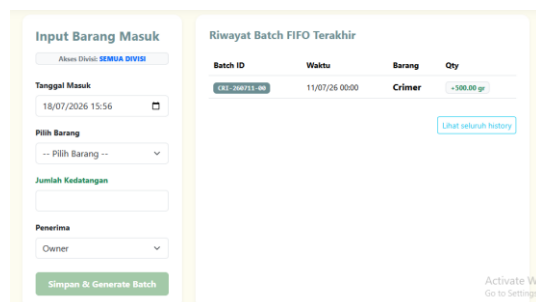
Halaman ini digunakan untuk mengklasifikasi item persediaan ke dalam kategori tertentu guna mempermudah manajemen persediaan.



Gambar 7. Halaman Kategori

4.1.5 Halaman Persediaan Masuk

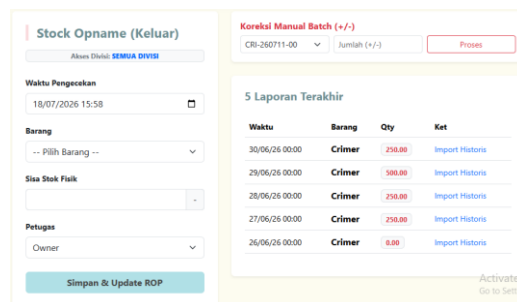
Halaman ini digunakan untuk mengelola data persediaan sesuai dengan hak akses (*role*). Dapat menambahkan item, mengedit item dan menghapus item serta proses inisialisasi parameter persediaan ROP dan *generate Batch ID* persediaan awal.



Gambar 8. Halaman Persediaan Masuk

4.1.6 Halaman Persediaan Keluar

Halaman ini merupakan antarmuka untuk mencatat penggunaan persediaan harian, di mana sistem secara otomatis menerapkan logika pemotongan FIFO, melakukan kalkulasi ulang ROP, serta memfasilitasi koreksi pencatatan jika diperlukan.



Gambar 9. Halaman Persediaan Keluar

4.2 Hasil Pengujian Sistem

Tahap ini dilakukan untuk mengevaluasi fungsionalitas sistem serta memastikan seluruh fitur dan algoritma berjalan sesuai dengan spesifikasi yang telah ditetapkan.

4.2.1 Pengujian Black-Box

Pengujian *Black-box* difokuskan pada pengujian spesifikasi fungsional perangkat lunak untuk memastikan setiap modul dapat menerima input dan menghasilkan output yang diharapkan.

Hasil pengujian pada modul autentikasi dan navigasi menunjukkan bahwa sistem berhasil menerapkan pembatasan hak akses (*role-based access control*) dengan baik. Pengguna dengan peran *Head Bar* dan *Head Kitchen* hanya dapat mengakses form transaksi serta melihat grafik Dashboard yang relevan dengan divisinya (makanan atau minuman), sementara fitur pengelolaan Master Kategori terkunci, kecuali untuk hak akses (*role*) *Owner*.

Pada aspek validasi data, sistem terbukti sangat responsif dalam menangani potensi kesalahan (*human error*). Sistem secara otomatis menolak dan memunculkan peringatan (*error alert*) ketika pengguna mencoba melakukan memasukan data yang tidak wajar, seperti menyimpan nama persediaan yang terduplikasi, menginput kuantitas bernilai nol atau negatif pada form Persediaan Masuk, hingga mencoba mengeluarkan sisa persediaan fisik yang jumlahnya melebihi stok yang tercatat di dalam sistem.

pengujian fungsionalitas ini membuktikan bahwa sistem berhasil mengeksekusi otomatisasi algoritma persediaan dengan baik. Saat transaksi persediaan masuk, sistem sukses menggenerasikan *Batch ID* secara terurut berdasarkan waktu. Pada transaksi persediaan keluar, sistem terbukti mampu melakukan perulangan (*looping*) pemotongan persediaan berbasis *First-In, First-Out* (FIFO) dengan menghabiskan *Batch* terlama terlebih dahulu sebelum memotong *Batch* yang lebih baru. Pemotongan persediaan ini juga secara instan memicu penyesuaian nilai *Reorder Point* (ROP), membuktikan bahwa seluruh fungsionalitas sistem telah berjalan dengan baik.

4.2.2 Pengujian White-Box

Pengujian *White-box* bertujuan untuk meneliti logika kode internal (*source code*), terutama pada algoritma yang paling krusial, yaitu logika pemotongan FIFO serta kalkulasi ROP saat input persediaan keluar. Pengujian dilakukan dengan memetakan *flow graph* dan menghitung *Cyclomatic Complexity V* (G) untuk memastikan seluruh jalur logika tereksekusi dengan benar. Pada Tabel 1 merupakan hasil pengujian *white-box* yang telah dilakukan :

Table 1. Pengujian *White-Box* Sistem

Modul	Jumlah Node (N)	Jumlah Busur (E)	Cyclomatic Complex V (G)	Jalur Independen
Login	8	9	3	3
Dashboard	10	12	4	4
Master Barang	14	15	3	3
Kategori	6	7	3	3
Persediaan masuk	11	12	3	3
Persediaan Keluar	13	15	4	4

4.2.3 Pengujian Akurasi Perhitungan ROP

Pengujian ini bertujuan untuk menguji tingkat akurasi perhitungan ROP pada sistem dengan membandingkan hasil komputasi sistem terhadap perhitungan matematis manual menggunakan sampel data riil historis penggunaan persediaan “Crimer” selama 30 hari operasional di Restoran D’Lisa Beach.

Guna efisiensi pelaporan, data penggunaan harian dikonsolidasikan ke dalam nilai akumulasi dan parameter statistik utama yang menjadi komponen dasar perhitungan ROP, sebagaimana disajikan pada Tabel 2 berikut :

Table 2. Parameter Perhitungan ROP

Parameter Komputasi	Nilai
Total Penggunaan	8.125 gram
Rata-rata Penggunaan Harian (<i>d</i>)	270,83 gram/hari
Standar Deviasi (σ)	260,375
Lead Time (<i>L</i>)	1 hari

Berdasarkan parameter pada Tabel 2 dan ketentuan nilai batas aman (*Z-score* = 1.65), perhitungan stok pengaman (*Safety-Stock/SS*) secara manual didapatkan sebagai berikut :

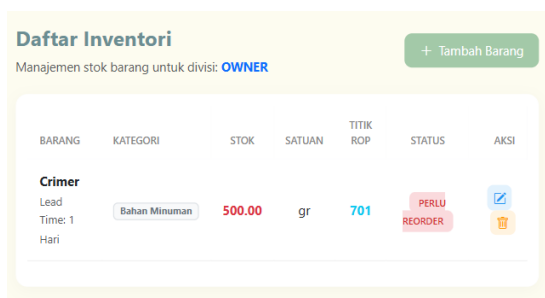
$$SS = 1.65 \times 260,375 \times \sqrt{1} = 429,618$$

Dengan didapatkan nilai *Safety-stock* tersebut, maka penentuan batas minimum *Reorder Point* (ROP) adalah :

$$\begin{aligned} \text{ROP} &= (270.83 \times 1) + 429.5775 \\ &= 700.44875 \end{aligned}$$

Berdasarkan aturan logika pada kode program (*source code*), sistem menggunakan fungsi *ceiling* untuk membulatkan angka desimal ke atas guna mencegah risiko kekurangan persediaan. Maka, nilai ROP ideal secara matematis untuk persediaan Crimer adalah 701 gram.

Sebagai bentuk validasi, perhitungan teoritis tersebut dikomparasikan dengan hasil keluaran komputasi otomatis oleh sistem pada halaman Master Barang, sebagaimana diperlihatkan pada Gambar 10 berikut :



BARANG	KATEGORI	STOK	SATUAN	TITIK ROP	STATUS	AKSI
Crimer	Bahan Minuman	500.00	gr	701	PERLU REORDER	[Icon]

Gambar 10. Hasil ROP dan Notifikasi Sistem

4.2.4 Analisis dan Kesimpulan Pengujian

Berdasarkan perhitungan manual di atas, nilai ROP ideal untuk persediaan crimer adalah 701gr dengan persediaan *pengaman* (*safety stock*) sebesar 429.618. Saat ini, persediaan terbaru crimer yang tercatat di dalam sistem adalah 500gr. Karena sistem diatur agar notifikasi batas stok baru akan menyala jika (Stok Aktual < ROP), maka dapat disimpulkan bahwa persediaan crimer saat ini dalam status Stok Kritis (karena 500gr < 701gr). Hasil perhitungan manual ini telah terbukti presisi dan selaras dengan output perhitungan yang dihasilkan oleh sistem. Hal ini membuktikan bahwa modul perhitungan persediaan ROP pada aplikasi telah lulus uji validasi logika dengan sangat baik.

5. KESIMPULAN

Berdasarkan hasil perancangan sistem, implementasi, dan pengujian yang telah

dipaparkan sebelumnya, dapat ditarik kesimpulan sebagai berikut :

1. Sistem Informasi Manajemen Persediaan berbasis website telah berhasil dibangun menggunakan kerangka kerja *Waterfall* dengan bahasa pemrograman PHP *Native* dan basis data MySQL. Kehadiran sistem ini terbukti sukses mengonversi kendala pengelolaan manual menjadi proses yang terkomputerisasi sehingga pemantauan ketersediaan barang dapat dilakukan secara *real-time* dan efisiensi operasional Restoran D'Lisa Beach dapat meningkat secara signifikan.
2. Penerapan metode *First-In, First-Out* (FIFO) di dalam logika sistem terbukti efektif untuk memprioritaskan rotasi persediaan secara otomatis. Kelebihan sistem dalam mengeksekusi pemotongan persediaan secara berurutan berdasarkan *Batch ID* dengan waktu kedatangan terlama sangat berdampak positif dalam menekan pemborosan finansial akibat risiko persediaan kadaluwarsa.
3. Integrasi metode *Reorder Point* (ROP) berhasil diimplementasikan untuk memberikan peringatan batas minimum pemesanan ulang secara otomatis tanpa memerlukan pengecekan fisik secara manual. Akurasi kalkulasi algoritma ROP di dalam sistem ini juga telah tervalidasi secara akurat melalui uji komparasi dengan perhitungan matematis, sehingga restoran dapat mencegah kekosongan persediaan (*stock-out*) terutama pada saat jam sibuk operasional.
4. Evaluasi sistem dilakukan menggunakan skema pengujian *black-box* dan *white-box* mengonfirmasi bahwa seluruh fungsionalitas antarmuka telah berjalan sesuai dengan spesifikasi yang dirancang. Pengujian alur kontrol logika internal perangkat lunak juga membuktikan bahwa algoritma sistem telah beroperasi dengan sangat baik.
5. Meskipun sistem telah beroperasi secara valid, kelemahan sistem saat ini terletak pada notifikasi peringatan pemesanan ulang yang hanya dapat dipantau apabila pengguna secara aktif mengakses website. Oleh karena itu, kemungkinan pengembangan selanjutnya disarankan agar berfokus pada integrasi sistem dengan layanan pesan pihak

ketiga seperti *WhatsApp* atau *Email* agar notifikasi dapat diterima langsung di perangkat seluler, serta penambahan algoritma pendukung keputusan untuk memprakirakan kebutuhan persediaan di masa mendatang berdasarkan data historis.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] Y. Saputra and D. Mardiaty, "Implementasi sistem informasi manajemen klinik menggunakan metode black box testing," vol. 13, no. 1, 2025, doi: <http://dx.doi.org/10.23960/jitet.v13i1.6015> IMPLEMENTASI.
- [2] G. Afia, "Makalah sistem informasi manajemen," 2020.
- [3] I. T. Sitio, Mugiarto, and R. Sari, "Implementasi Metode FIFO dan Reorder Point dalam Sistem Informasi Penjualan Produk pada Toko Makanan," vol. 1, no. 1, pp. 1–11, 2025.
- [4] Brahmantyo, "Rancang Bangun Sistem Informasi Manajemen Menggunakan Metode Safety Stock Dan Reorder Point Pada PT. Indonesia Berkah Mandiri," 2022.
- [5] J. Heizer, B. Render, and C. Munson, *Operations Management*. 2020.
- [6] N. Amahoru, "Makalah manajemen persediaan," no. June, 2023.
- [7] S. Tonni Limbong, *Pemrograman Web Dasar*. Yayasan Kita Menulis, 2021.
- [8] A. Mareta, A. N. Anissa, N. T. Sunggono, and R. N. Royadi, "Penerapan Metode FIFO Untuk Mengelola Persediaan Bahan Baku Pada PT MPM," pp. 793–804, 2025, doi: <http://dx.doi.org/10.35889/progresif.v21i2.2924>.
- [9] F. Fitra Rafifan, R. U. Maghfiroh, U. Islam, N. Sunan, and A. Surabaya, "Analisis Penerapan Metode First In First Out (FIFO) terhadap Efisiensi Manajemen Operasional pada Toko Kelontong Reni Fakultas Ekonomi dan Bisnis , Universitas Islam Negeri Sunan Ampel Surabaya , Indonesia (FIFO), menjadi sangat penting untuk meningkat," vol. 3, no. September, 2025, doi: <https://doi.org/10.59603/ppiman.v3i4.1087>.
- [10] P. K. S. C. W. E. M. Hudori, "Formulasi Model Safety Stock dan Reorder Point untuk Berbagai Kondisi Persediaan Material," vol. X, no. 3, pp. 217–224, 2018.
- [11] I. Gina, S. C. R. Ramadhan, and V. Hartati, "Optimasi Safety Stock dan Reorder Point untuk Mengurangi Stockout Produk Jadi di PT XYZ," vol. 04, no. 01, pp. 12–19, 2025, doi: <https://doi.org/10.33197/jlscc.v4i1.3003>.
- [12] M. Nur Muhammad, M. A. Nurdin, and H. Permatasari, "Pengujian Aplikasi Pengelolaan Laundry Menggunakan Metode Blackbox Testing dan Whitebox Testing," pp. 523–528, 2023.
- [13] D. Madhavi, "A White Box Testing Technique in Software Testing : Basis Path Testing," vol. 02, no. 04, pp. 12–17, 2016.
- [14] E. Sita Eriana, "PENGUJIAN SISTEM INFORMASI APLIKASI PERPUSTAKAAN BERBASIS WEB DENGAN WHITE BOX TESTING Emi," vol. XV, no. 10, pp. 28–33, 2020.
- [15] I. Bakti and M. Firdaus, *WATERFALL : METODE PERANCANGAN SOFTWARE UNTUK PEMULA*. 2024.