

PENGEMBANGAN APLIKASI INVENTORY MENGGUNAKAN ALGORITMA DIFFERENTIAL EVOLUTION DAN C 4.5 BERBASIS WEB

Moch Miftakhul Rizal^{1*}, Mukh. Taofik Chulkamdi², Filda Febrinita³

^{1,2,3}Teknik Informatika, Fakultas Teknik dan Informatika, Universitas Islam Balitar

E-mail: mftkhlrizal@gmail.com^{1*}, chulkamdi@gmail.com², febrinitafilda80@gmail.com³

INFO ARTIKEL

Sejarah Artikel

Diterima : 14/10/2025

Direvisi : 01/11/2025

Diterbitkan : 01/12/2025

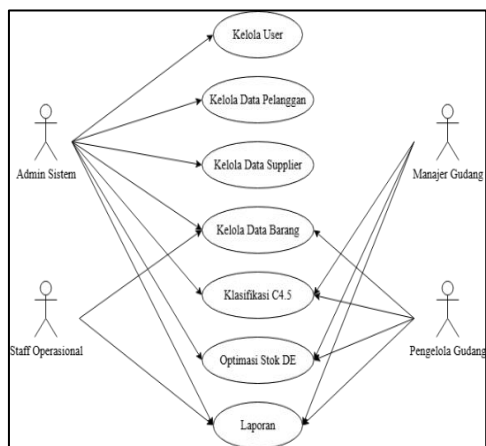
*Corresponding author

mftkhlrizal@gmail.com

DOI: 10.70247/jumistik.v4i2.219

<https://ojs.amiklps.ac.id>

GRAPHICAL ABSTRACT



ABSTRACT

The main issue in Kepinos Warehouse lies in inefficient stock management, characterized by frequent cases of overstock, understock, and inaccurate record-keeping that disrupt operational performance. This study aims to develop a web-based inventory system integrating the Differential Evolution (DE) algorithm for stock optimization and the C4.5 algorithm for item category classification based on sales frequency. The system was developed using the Rapid Application Development (RAD) methodology to ensure iterative and user-centered design. System testing was conducted using Black Box Testing to verify functionality and Closed Beta Testing to evaluate user feasibility. The results show that all features performed as expected, with user validation scores of 74% from IT experts, 86.4% from warehouse administrators, and 76% from operational staff. These findings indicate that the system is technically and functionally feasible, enhancing efficiency, accuracy, and responsiveness in warehouse inventory management.

Keywords: Inventory System, Differential Evolution, C4.5, Stock Optimation, RAD

ABSTRAK

Permasalahan utama dalam pengelolaan stok di Kepinos Warehouse adalah terjadinya *overstock*, *understock*, serta ketidaktepatan pencatatan yang berdampak pada menurunnya efisiensi operasional. Penelitian ini bertujuan untuk mengembangkan aplikasi sistem *inventory* berbasis web yang mengintegrasikan algoritma *Differential Evolution* (DE) untuk optimasi kebutuhan stok serta algoritma C4.5 untuk klasifikasi kategori barang berdasarkan tingkat kelarisan. Metode pengembangan yang digunakan adalah *Rapid Application Development* (RAD) yang menekankan kecepatan iterasi dan keterlibatan aktif pengguna. Pengujian sistem dilakukan melalui *Black Box Testing* untuk verifikasi fungsionalitas dan *Closed Beta Testing* guna menilai aspek kelayakan dari sisi pengguna. Hasil penelitian menunjukkan bahwa seluruh fitur berfungsi sesuai skenario yang ditetapkan, dengan hasil uji kelayakan sebesar 74% dari ahli IT, 86,4% dari admin gudang, dan 76% dari staf operasional. Temuan tersebut menunjukkan bahwa sistem dinilai layak digunakan dari aspek teknis, fungsionalitas, dan kemudahan penggunaan. Secara keseluruhan, penerapan algoritma DE dan C4.5 mampu meningkatkan efisiensi, akurasi, serta responsivitas dalam pengelolaan stok barang di lingkungan gudang.

Kata kunci: Sistem Inventory, Differential Evolution, C4.5, Optimasi Stok, RAD

PENDAHULUAN

Pengelolaan inventaris merupakan elemen penting dalam mendukung efisiensi operasional organisasi, baik di sektor publik maupun swasta. Sistem inventaris yang efektif tidak hanya berfungsi sebagai sarana pencatatan data barang, tetapi juga menjadi dasar bagi pengambilan keputusan strategis dalam hal perencanaan, distribusi, serta pengendalian aset. Meski demikian, banyak instansi masih menghadapi tantangan dalam menjaga akurasi dan efisiensi data inventaris akibat penggunaan metode pencatatan manual. Ketergantungan pada pencatatan konvensional sering kali menimbulkan kesalahan, redundansi data, serta keterlambatan proses administrasi.

Sebagai contoh, Badan Pusat Statistik (BPS) Kabupaten Barito Kuala masih mengelola inventaris secara manual, sehingga sering ditemukan ketidaksesuaian antara data administrasi dan kondisi barang di lapangan. Kondisi serupa terjadi pada BPS Kabupaten Sukoharjo, di mana sistem inventarisasi aset masih dilakukan menggunakan lembar kerja terpisah, sehingga pengelolaan data menjadi tidak efisien dan rawan kesalahan manusia. Permasalahan tersebut menunjukkan bahwa sistem manual tidak lagi memadai dalam mendukung kebutuhan operasional yang dinamis dan berbasis data *real-time*. Oleh karena itu, dibutuhkan pendekatan digital yang mampu mengintegrasikan seluruh proses inventarisasi dalam satu sistem terpadu.

Penerapan teknologi informasi berbasis web menjadi solusi potensial dalam mengatasi permasalahan tersebut. Sistem *web-based* memungkinkan data inventaris tersimpan secara terpusat, mudah diakses oleh berbagai pengguna, serta mendukung pembaruan data secara *real-time*. Menurut [1], penerapan sistem inventaris berbasis web mampu meningkatkan efisiensi, akurasi pencatatan, dan transparansi informasi dalam pengelolaan aset. Hal ini sejalan dengan temuan BPS Kabupaten Tegal [2] yang menegaskan pentingnya sistem inventaris terintegrasi untuk memantau jumlah, nilai, dan kondisi *Barang Milik Negara (BMN)* secara akurat. Dengan demikian, digitalisasi pengelolaan inventaris merupakan langkah strategis dalam mendukung tata kelola aset yang baik dan efisien.

Berangkat dari kondisi tersebut, penelitian ini berfokus pada pengembangan aplikasi *web-based inventory system* yang mengintegrasikan dua algoritma kecerdasan buatan, yaitu *Differential Evolution (DE)* untuk optimasi stok dan *C4.5* untuk klasifikasi barang berdasarkan tingkat kelarisan. Penelitian dilakukan pada Kepinos Warehouse, sebuah usaha yang menghadapi berbagai permasalahan seperti *overstock*, *understock*, dan ketidaktepatan pencatatan stok akibat sistem manual. Integrasi kedua algoritma tersebut diharapkan mampu meningkatkan efektivitas pengelolaan inventaris, meminimalisir kesalahan pencatatan, serta membantu manajemen dalam

mengambil keputusan berbasis data yang akurat dan prediktif.

Dalam sistem yang dikembangkan, algoritma *Differential Evolution* berperan mengoptimalkan jumlah stok agar sesuai dengan kebutuhan dan permintaan pelanggan. Algoritma ini mencari solusi terbaik melalui proses evolusi populasi kandidat menggunakan tahapan *mutation*, *crossover*, dan *selection* [3]. Sementara itu, algoritma *C4.5* digunakan untuk mengklasifikasikan data inventaris berdasarkan pola permintaan dan tingkat kelarisan barang. Menurut [4], kombinasi kedua algoritma tersebut mampu memperkuat kinerja sistem dalam mengolah data berukuran besar (*big data*) dan menghasilkan prediksi yang lebih akurat. Dengan demikian, integrasi *DE* dan *C4.5* memberikan solusi yang komprehensif terhadap permasalahan optimasi dan klasifikasi stok dalam sistem inventaris.

Temuan ini diperkuat oleh beberapa penelitian terdahulu. Penelitian oleh Amarullah, dkk. [5] menunjukkan bahwa penerapan algoritma *C4.5* pada sistem manajemen stok obat di Apotek Nafa Farma mampu meningkatkan efisiensi pencatatan dan prediksi kebutuhan dengan tingkat akurasi mencapai 80,67% serta *F1-score* rata-rata sebesar 80,52%. Temuan serupa juga dilaporkan oleh Alyshen dan Harman [6] pada PT Prima Niaga Indomas, di mana penggunaan algoritma *C4.5* dalam memprediksi penjualan barang terbukti meningkatkan ketepatan estimasi stok dan efisiensi inventaris secara signifikan. Berdasarkan hasil tersebut, integrasi antara algoritma *C4.5* dan *Differential Evolution* dalam penelitian ini diharapkan dapat memperkuat kemampuan sistem dalam melakukan prediksi stok sekaligus optimasi pemesanan secara adaptif dan akurat.

Dalam konteks objek penelitian, Kepinos Warehouse mengalami peningkatan jumlah pelanggan rutin dari delapan pada tahun 2022 menjadi tiga belas pada tahun 2024. Pertumbuhan pelanggan ini diikuti oleh peningkatan aktivitas keluar masuk barang yang signifikan. Proses pencatatan yang masih menggunakan buku dan *Microsoft Excel* sering kali menyebabkan kesalahan, kehilangan data, dan keterlambatan pemesanan ulang. Akibatnya, perusahaan kerap mengalami ketidakseimbangan stok, baik berupa *overstock* yang menyebabkan penumpukan barang di gudang maupun *understock* yang menunda pemenuhan permintaan pelanggan. Situasi ini berdampak pada penurunan kepuasan pelanggan dan potensi kerugian akibat kerusakan barang.

Masalah tersebut menunjukkan pentingnya penerapan sistem inventaris yang tidak hanya berfungsi mencatat data, tetapi juga mampu menganalisis dan memprediksi kebutuhan stok secara otomatis. Dengan menerapkan algoritma *Differential Evolution* dan *C4.5*, sistem diharapkan dapat melakukan pengambilan keputusan secara cerdas, seperti menentukan jumlah stok optimal serta mengidentifikasi barang dengan tingkat permintaan tinggi berdasarkan pola historis penjualan.

Proses pengembangan sistem dilakukan menggunakan metode *Rapid Application Development (RAD)* karena metode ini memungkinkan pengembangan yang cepat, iteratif, dan melibatkan pengguna secara aktif pada setiap tahapannya. Sitio dkk. [7] menjelaskan bahwa RAD efektif diterapkan pada proyek yang memerlukan waktu pembangunan singkat dengan tetap mempertahankan kualitas hasil. Dengan metode ini, sistem dapat dikembangkan secara fleksibel sesuai dengan kebutuhan pengguna di lapangan, khususnya dalam konteks operasional gudang yang dinamis seperti di Kepinos Warehouse.

Setelah sistem selesai dikembangkan, dilakukan pengujian dengan dua metode, yaitu *Black Box Testing* dan *Closed Beta Testing*. Metode *Black Box Testing* digunakan untuk menguji fungsionalitas sistem tanpa melihat struktur internal kode, memastikan bahwa setiap fitur berfungsi sesuai rancangan. Sementara itu, *Closed Beta Testing* dilakukan dengan melibatkan tiga kelompok pengguna, yaitu ahli IT, admin gudang, dan staf operasional, untuk menilai aspek teknis, fungsional, dan kemudahan penggunaan. Pendekatan ini sesuai dengan pandangan Sugiono (2022) bahwa uji teknis dan validasi pengguna perlu dilakukan secara paralel guna menjamin reliabilitas sistem sebelum implementasi penuh.

Dengan demikian, penelitian ini diharapkan mampu menghasilkan sistem inventaris berbasis web yang tidak hanya efisien secara teknis, tetapi juga efektif dalam mendukung pengambilan keputusan strategis. Melalui integrasi algoritma *Differential Evolution* dan C4.5, sistem diharapkan dapat meningkatkan akurasi prediksi stok, mengoptimalkan alur distribusi barang, serta meminimalkan potensi kesalahan manusia. Secara praktis, penerapan sistem ini dapat membantu Kepinos Warehouse mengelola inventaris secara lebih terstruktur, responsif, dan berkelanjutan, sekaligus menjadi kontribusi nyata dalam pengembangan teknologi informasi di bidang logistik dan manajemen persediaan.

TINJAUAN PUSTAKA

Inventory

Inventory atau persediaan merupakan bahan atau barang yang disimpan untuk tujuan tertentu, seperti proses produksi yang mengubah bahan mentah menjadi produk jadi. Jika barang yang disimpan berupa komponen, maka akan dijual kembali sebagai barang dagangan. Menurut Sudarto [8], *inventory* adalah stok barang atau sumber daya yang digunakan dalam organisasi perusahaan dan memiliki peran penting dalam meningkatkan fleksibilitas serta efisiensi operasi produksi. Fungsi utama *inventory* antara lain untuk mengantisipasi permintaan pelanggan, memisahkan komponen produksi, menstabilkan operasi dari fluktuasi permintaan, memperlancar kebutuhan musiman, serta memungkinkan perusahaan memperoleh potongan

harga karena pembelian dalam jumlah besar. Selain itu, *inventory* berfungsi sebagai penyangga dalam operasi produksi, mencegah kekurangan stok akibat keterlambatan pasokan, melindungi dari inflasi, dan membantu mengontrol biaya pembelian. Berdasarkan klasifikasinya, *inventory* dapat dibedakan menjadi lima kategori, yaitu persediaan bahan baku (*raw material*), barang dalam proses, *supplies inventory* sebagai penunjang operasi, barang dagangan, dan barang jadi hasil produksi [9].

Optimasi

Optimasi adalah cabang ilmu matematika yang berfokus pada pencarian nilai maksimum atau minimum suatu fungsi untuk mencapai hasil paling efisien dengan sumber daya terbatas. Konsep ini sejalan dengan prinsip ekonomi yang menekankan pengeluaran minimal untuk hasil maksimal [10]. Teknik *optimasi* terbagi menjadi *mathematical programming* seperti *support vector machines* dan *gradient descent*, serta *combinatorial optimization* seperti *graph theory* dan *genetic algorithm*. Penerapan *optimasi* meliputi bidang teknologi, ekonomi, hingga pertanian. Dalam praktiknya, *optimasi* dapat diterapkan misalnya untuk meningkatkan performa komputer dengan menonaktifkan program tidak diperlukan agar sistem bekerja lebih efisien.

Differential Evolution

Differential Evolution (DE) merupakan algoritma *optimasi* berbasis populasi yang dikembangkan dari konsep evolusi biologis. Menurut Wang dan Yu [11], algoritma ini bekerja melalui proses *mutation*, *crossover*, dan *selection* untuk menghasilkan solusi yang semakin mendekati nilai optimal pada setiap iterasi. Proses tersebut memungkinkan DE mengeksplorasi ruang pencarian secara luas sekaligus mempertahankan kemampuan eksploitasi terhadap solusi potensial.

Selain itu, Farda dan Thammano, 2023 [12] menyebutkan bahwa fleksibilitas DE membuatnya efektif diterapkan pada berbagai permasalahan *optimasi* numerik dan multidimensi. Pengembangan strategi mutasi adaptif dapat meningkatkan kecepatan konvergensi dan mengurangi kemungkinan terjebak pada *local optimum*. Dengan demikian, *Differential Evolution* dipandang sebagai metode *optimasi* yang efisien, stabil, dan relevan untuk mendukung sistem pengambilan keputusan berbasis komputasi [13].

Algoritma C4.5

Algoritma C4.5 merupakan pengembangan dari *ID3* yang populer untuk menyusun *decision tree* pada skenario *supervised learning*, di mana pemilihan atribut didasarkan pada ukuran informasi seperti *entropy* dan *information gain* sehingga atribut paling informatif dijadikan pemisah awal dalam pohon klasifikasi. Proses pembentukan pohon meliputi penentuan akar (atribut dengan *gain* tertinggi), pembentukan cabang untuk setiap nilai atribut,

pembagian ulang kasus pada setiap cabang, dan iterasi hingga setiap cabang homogen terhadap kelas target. Karena struktur pohon yang intuitif dan kemampuannya menangani atribut kontinu maupun diskrit, C4.5 banyak dipakai dalam aplikasi nyata mulai dari prediksi penjualan hingga analisis risiko, penelitian terbaru juga menyorot upaya mempercepat dan mendistribusikan perhitungan C4.5 untuk data berukuran besar [14]. Selain itu, modifikasi *modern* misalnya strategi pemilihan fitur yang adil atau mekanisme *pruning* dan mutasi parameter telah diteliti untuk meningkatkan akurasi, kestabilan, dan keadilan model pada berbagai *domain*, sehingga C4.5 tetap relevan sebagai metode klasifikasi yang mudah diinterpretasi dan dapat ditingkatkan untuk kebutuhan big data [15].

Rapid Application Development

Rapid Application Development (RAD) merupakan metode pengembangan perangkat lunak yang menekankan kecepatan dan fleksibilitas dalam proses pembuatan aplikasi. Pendekatan ini berfokus pada keterlibatan aktif pengguna melalui umpan balik berkelanjutan selama tahap pengembangan, sehingga sistem yang dihasilkan lebih sesuai dengan kebutuhan [16]. RAD terdiri atas beberapa fase utama, yaitu perencanaan kebutuhan, perancangan cepat (*rapid design*), pembangunan iteratif (*construction*), serta penerapan dan evaluasi sistem. Setiap tahap dilakukan secara berulang dengan prototipe sebagai alat utama untuk mempercepat validasi fungsionalitas. Dengan demikian, waktu pengembangan dapat dipersingkat tanpa mengorbankan kualitas hasil akhir. Metode ini sangat efektif diterapkan pada proyek yang memiliki batas waktu ketat dan kebutuhan yang dapat berubah secara dinamis. Selain itu, RAD juga mendorong kolaborasi intensif antara pengembang dan pengguna, menjadikannya pendekatan yang efisien dalam menghasilkan solusi perangkat lunak yang adaptif dan responsif [17].

Blac Box Testing

Black Box Testing merupakan metode pengujian perangkat lunak yang berfokus pada evaluasi fungsionalitas sistem tanpa memperhatikan struktur internal atau kode sumbernya. Pengujian ini bertujuan untuk memastikan bahwa perangkat lunak berjalan sesuai dengan spesifikasi yang telah ditetapkan, dengan cara memberikan *input* dan mengamati *output* yang dihasilkan. Menurut Praniffa dkk. [18], *Black Box Testing* dilakukan berdasarkan spesifikasi fungsional tanpa memerlukan pengetahuan tentang implementasi internal. Metode ini berperan penting dalam mendeteksi kesalahan pada fungsi, antarmuka, model data, serta akses terhadap sumber data eksternal. Beberapa teknik yang umum digunakan dalam *Black Box Testing* antara lain *Equivalence Partitioning*, yang membagi domain *input* ke dalam partisi ekuivalen untuk efisiensi pengujian; *Boundary Value Analysis*, yang menguji nilai batas karena

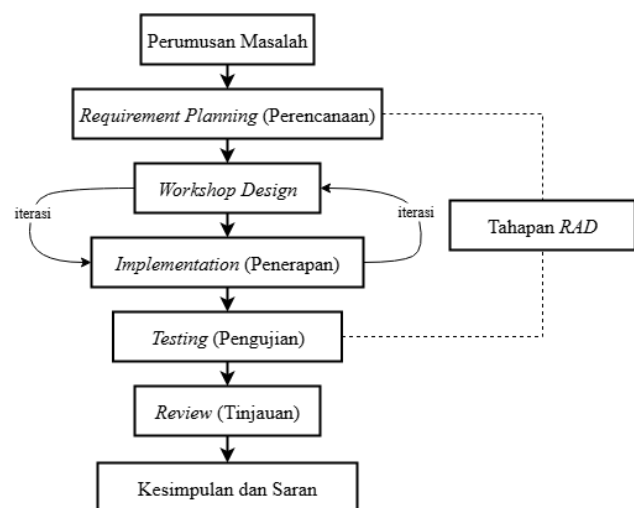
kesalahan sering terjadi di area tersebut; *Decision Table Testing*, yang memanfaatkan tabel keputusan untuk memverifikasi berbagai kombinasi *input*; serta *State Transition Testing*, yang menggunakan model *state machine* untuk memastikan transisi antarstatus perangkat lunak terjadi sesuai spesifikasi.

Closed Beta Testing

Closed Beta Testing merupakan tahap pengujian perangkat lunak yang dilaksanakan sebelum peluncuran resmi dengan melibatkan sejumlah pengguna terbatas guna menilai stabilitas sistem, menemukan kesalahan, serta memperoleh masukan pengguna. Menurut penelitian oleh Fata Nidaul Khasanah dkk. (2022) dalam artikel "*Beta Testing Techniques in Non-Functional Testing of Gamified Learning Applications*" [19], *Closed Beta Testing* melibatkan pengguna akhir untuk menguji aspek non-fungsional seperti kegunaan, kepuasan, dan kemudahan penggunaan sebelum peluncuran resmi. Pengujian ini dilakukan berdasarkan skenario penggunaan nyata dan umpan balik dari pengguna untuk memastikan aplikasi memenuhi harapan pengguna. Selanjutnya, studi oleh Hakim, dkk. (2024) [20] pada aplikasi TIJE menunjukkan bahwa *beta testing* menghasilkan identifikasi bug kritis dan peningkatan stabilitas setelah dilakukan *alpha testing*, sehingga versi beta menjamin aplikasi lebih siap digunakan publik. Dari dua studi tersebut, dapat disimpulkan bahwa *Closed Beta Testing* tidak hanya memungkinkan evaluasi sistem dalam kondisi nyata, tetapi juga memberi kesempatan untuk melakukan penyempurnaan berdasarkan umpan balik pengguna akhir guna memastikan kualitas produk sebelum dirilis secara formal.

METODOLOGI PENELITIAN

Tahapan Penelitian



Gambar 1. Tahapan Penelitian

Sumber : Hasil Olah Data

Gambar 1 menunjukkan tahapan metode *Rapid Application Development* (RAD) yang digunakan

dalam penelitian ini. Proses dimulai dari *Requirement Planning* untuk menentukan kebutuhan sistem, dilanjutkan dengan *Workshop Design* sebagai tahap perancangan bersama pengguna. Tahap berikutnya adalah *Implementation* dan *Testing* untuk memastikan sistem sesuai dengan kebutuhan yang telah dirumuskan. Jika ditemukan kekurangan, proses akan berulang ke tahap *Workshop Design* hingga sistem optimal. Metode RAD bersifat iteratif, fleksibel, cepat beradaptasi terhadap perubahan, serta melibatkan pengguna secara aktif dalam setiap siklus pengembangan. Adapun penjelasan rinci terkait proses RAD adalah sebagai berikut:

1. *Requirement Planning* (Perencanaan)
Tahap awal yang bertujuan untuk mengidentifikasi dan mendefinisikan kebutuhan sistem secara detail sebagai dasar dalam proses pengembangan berikutnya.
2. *Workshop Design* (Perancangan Bersama)
Pada tahap ini, pengembang dan pengguna berkolaborasi untuk merancang tampilan serta fungsionalitas sistem agar sesuai dengan kebutuhan yang telah ditentukan.
3. *Implementation* (Penerapan)
Merupakan proses penerjemahan hasil rancangan menjadi sistem nyata yang dapat dijalankan sesuai dengan desain dan kebutuhan pengguna.
4. *Testing* (Pengujian)
Dilakukan untuk memastikan seluruh fitur dan fungsi sistem berjalan sesuai spesifikasi serta mendeteksi kemungkinan kesalahan atau bug.
5. *Review* (Tinjauan dan Evaluasi)
Tahap akhir yang berfokus pada evaluasi menyeluruh terhadap hasil pengembangan. Jika ditemukan kekurangan, proses dapat kembali ke tahap *Workshop Design* untuk dilakukan perbaikan melalui iterasi.

Pengumpulan Data

Metode Pengumpulan Data

Metodologi pengumpulan data untuk mengumpulkan data dari sampel penelitian yang telah ditentukan, dilakukan dengan metode tertentu sesuai dengan tujuannya. Metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut:

- a. Observasi
Observasi merupakan suatu teknik yang dilakukan dengan cara mengadakan pengamatan secara teliti serta pencatatan secara sistematis. Dengan menggunakan metode ini dilakukan pengumpulan data dengan cara pengamatan secara langsung pelaksanaan kegiatan pada Kepinos Warehouse guna mengetahui secara langsung bagaimana sistem yang sedang berjalan.
- b. Wawancara
Wawancara merupakan pertemuan dua orang untuk bertukar informasi dan ide melalui tanya jawab, sehingga dapat dikonstruksikan makna dalam suatu topik tertentu. Dilakukan tanya jawab dengan pemilik Kepinos Warehouse untuk

memperoleh gambaran, keterangan, dan penjelasan untuk membantu dalam proses perancangan dan pembangunan sistem.

c. Studi Pustaka

Dilakukan studi pustaka sebagai bahan tambahan guna melengkapi sekurang-kurangnya data yang diperoleh dari wawancara maupun observasi.

Tahapan Pengembangan

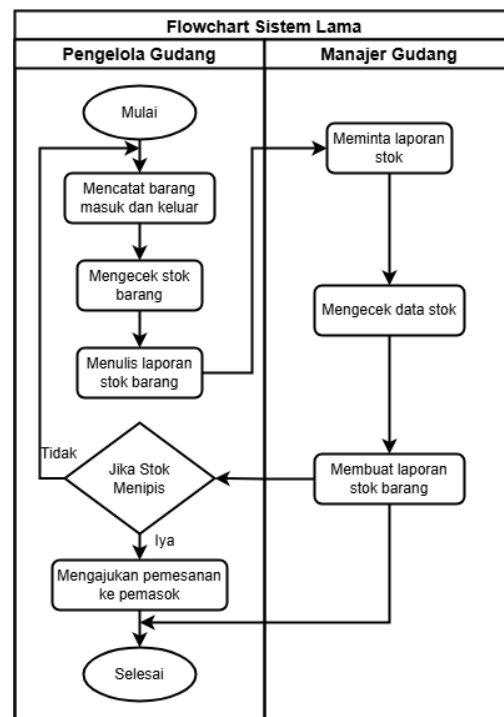
1. Requirement Planning (Perencanaan)

Perencanaan (*Requirement Planning*) dilakukan untuk mengidentifikasi dan merumuskan kebutuhan fungsional serta non-fungsional yang harus dipenuhi oleh aplikasi *inventory* yang dikembangkan. Proses analisis ini didasarkan pada hasil observasi langsung dan wawancara, yang mengungkap berbagai kendala dalam sistem pencatatan manual dan penggunaan spreadsheet. Proses ini menghasilkan kebutuhan *hardware* dan *software* sistem serta alur sistem lama dan baru sebagai berikut:

Tabel 1. Kebutuhan *hardware* dan *software*

No	Kebutuhan <i>hardware</i> dan <i>software</i>
1	Prosesor Intel i5 atau setara, RAM 8GB, Penyimpanan 256GB SSD
2	Laptop atau desktop dengan spesifikasi minimal RAM 4GB dan prosesor Intel i3
3	Operating System Linux atau Windows
4	Server Web Apache atau Nginx
5	Browser
6	Laravel dan ReactJS
7	Database MySQL atau PostgreSQL

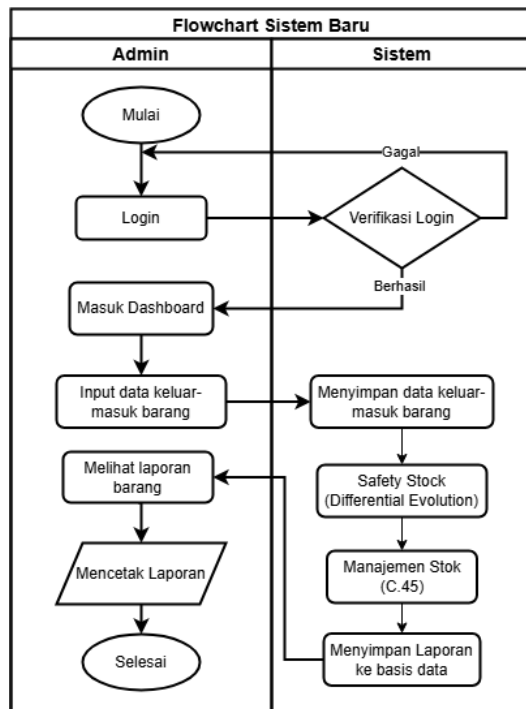
Sumber : Hasil Olah Data



Gambar 2. Flowchart Sistem Lama

Sumber : Hasil Olah Data

Setelah dirumuskan sistem yang sedang berjalan seperti gambar 2 maka kemudian dilakukan perumusan alur sistem yang baru. Alur tersebut nantinya digunakan sebagai acuan tujuan pengembangan sistem pada proses berikutnya.



Gambar 3. Flowchart Sistem Baru
Sumber : Hasil Olah Data

Gambar 3 menunjukkan bahwa sistem baru telah diimplementasikan dengan berbasis web, memungkinkan pengelolaan *inventory* secara otomatis dan lebih terstruktur.

2. Workshop Design (Perancangan Bersama)

Tahap *Workshop Design* menggambarkan proses iteratif dalam perancangan sistem aplikasi *inventory* berbasis web agar fungsional, efisien, dan mudah digunakan. Pada *iterasi pertama*, rancangan awal dievaluasi melalui masukan pengguna terkait alur *login*, struktur data, dan tata letak menu, sehingga dilakukan penyederhanaan proses serta perbaikan tampilan. *Iterasi kedua* melibatkan pengguna utama untuk menguji hasil revisi, menghasilkan perbaikan pada navigasi, penambahan fitur notifikasi stok menipis, dan peningkatan konsistensi desain. Sementara itu, *iterasi ketiga* difokuskan pada penyempurnaan aspek *UI/UX*, mencakup pengaturan ukuran huruf, warna, dan tata letak tombol agar lebih ergonomis. Hasil akhir tahap ini menghasilkan antarmuka yang konsisten, intuitif, serta sesuai dengan kebutuhan operasional pengguna.

3. Implementation (Penerapan)

Pada tahap ini rancangan yang telah dibangun akan diimplementasikan dengan melakukan pemrograman yang disesuaikan dengan rancangan

yang ada. Pada tahap ini juga dilakukan instalasi program yang dibutuhkan untuk mengakses sistem yang telah dibangun berbasis web seperti browser dan melakukan konfigurasi agar sistem berjalan dengan baik.

4. Testing (Pengujian)

Dalam proses pengembangan sistem *inventory* berbasis web, tahap *testing* merupakan bagian krusial yang bertujuan untuk memastikan sistem berfungsi sesuai dengan kebutuhan dan spesifikasi yang telah ditetapkan. Pada penelitian ini digunakan dua metode pengujian utama, yaitu *Black Box Testing* dan *Closed Beta Testing*. Metode *Black Box Testing* dilaksanakan oleh pengembang dan tim *quality assurance* untuk menguji fungsi sistem berdasarkan keluaran yang dihasilkan tanpa melihat kode sumber, meliputi validasi input-output, fungsi utama, serta navigasi antarmuka. Selanjutnya, *Closed Beta Testing* dilakukan oleh pengguna akhir dalam lingkungan terbatas guna menilai keandalan sistem, responsivitas antarmuka, dan kemudahan penggunaan. Hasil umpan balik dari tahap ini digunakan untuk penyempurnaan sistem sebelum diimplementasikan secara menyeluruh.

5. Review (Tinjauan dan Evaluasi)

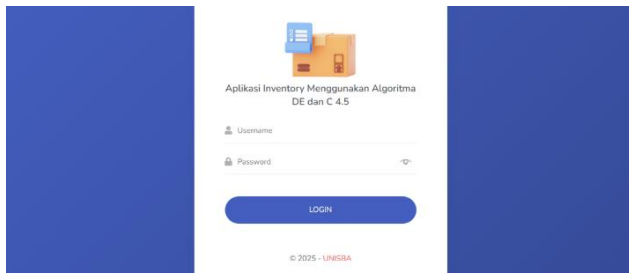
Tahap akhir dalam proses pengembangan sistem merupakan fase *review* yang bertujuan memastikan kesesuaian sistem dengan kebutuhan serta harapan pengguna. Apabila pada tahap ini ditemukan ketidaksesuaian atau kekurangan, maka proses akan kembali ke tahap *Workshop Design* untuk dilakukan penyempurnaan melalui mekanisme *iteration*. Evaluasi dilakukan secara menyeluruh guna menilai efektivitas, fungsionalitas, dan keandalan sistem sebelum diimplementasikan secara penuh. Setelah sistem dinyatakan sesuai, langkah selanjutnya adalah penarikan kesimpulan dan penyusunan saran yang berfungsi sebagai dasar refleksi hasil penelitian serta acuan bagi pengembangan di masa mendatang.

HASIL DAN PEMBAHASAN

Hasil pengembangan

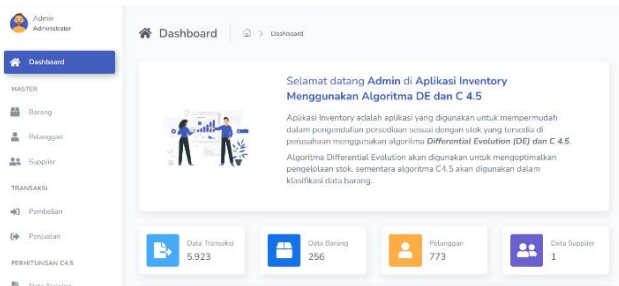
a. Halaman Login

Halaman ini dirancang sederhana dengan menampilkan form input untuk *username* dan *password* yang harus diisi oleh pengguna sebelum mengakses sistem. Tujuannya adalah untuk memastikan keamanan data serta membatasi akses hanya kepada pengguna yang memiliki otorisasi



Gambar 4. Halaman Login
Sumber : Hasil Olah Data

b. Halaman Dashboard

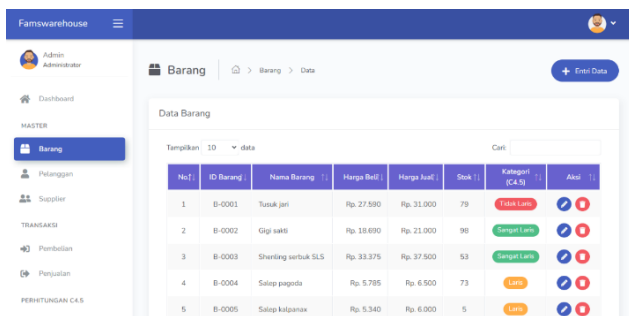


Gambar 5. Halaman Dashboard
Sumber : Hasil Olah Data

Gambar 5 menunjukkan tampilan halaman *dashboard* dari aplikasi *inventory*. Halaman ini menyajikan ringkasan informasi penting seperti jumlah total barang, stok minimum, barang masuk dan keluar. *Dashboard* dirancang sebagai pusat kontrol utama yang memberikan gambaran umum kondisi inventaris secara *real-time* untuk mendukung pengambilan keputusan cepat dan tepat oleh pengguna.

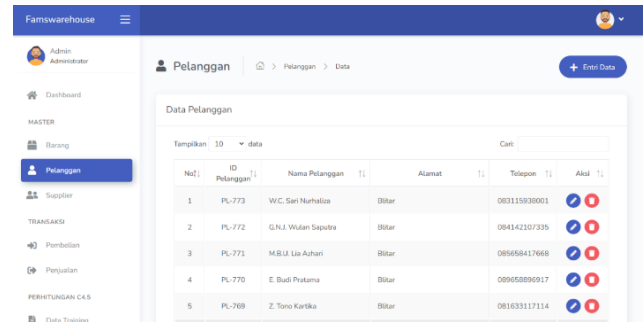
c. Halaman Data Barang

Halaman data barang menampilkan daftar barang beserta informasi seperti nama, harga beli dan jual, stok tersedia, serta stok minimum. Melalui antarmuka ini, pengguna dapat menambah, mengedit, atau menghapus data barang dengan mudah, sehingga pengelolaan data menjadi lebih efisien dan akurat.



Gambar 6. Halaman Data Barang
Sumber : Hasil Olah Data

d. Halaman Data Pelanggan

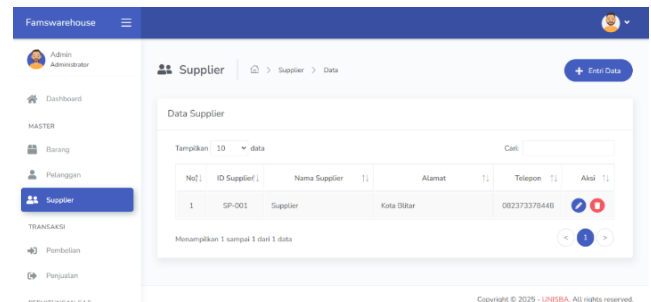


Gambar 7. Halaman Data Pelanggan
Sumber : Hasil Olah Data

Halaman ini digunakan untuk menampilkan dan mengelola informasi pelanggan, seperti nama pelanggan, alamat, dan kontak yang dapat dihubungi. Fitur ini memudahkan admin dalam mencatat data pelanggan rutin serta mempercepat proses transaksi dan distribusi barang sesuai kebutuhan masing-masing pelanggan.

e. Halaman Data Supplier

Halaman ini menampilkan informasi penting terkait *supplier*, meliputi nama, kontak, dan jenis barang yang disuplai. Fitur ini memudahkan admin dalam memantau hubungan kerja sama dengan *supplier* serta memastikan ketersediaan stok barang secara berkelanjutan dan tepat waktu.



Gambar 7. Halaman Data Supplier
Sumber : Hasil Olah Data

f. Halaman Transaksi Pembelian

Halaman transaksi pembelian digunakan untuk mencatat setiap proses pembelian barang dari *supplier*, yang mencakup informasi seperti tanggal pembelian, nama *supplier*, daftar barang yang dibeli, jumlah, dan total harga. Dengan adanya fitur ini, proses pencatatan menjadi lebih sistematis dan transparan, serta membantu dalam memantau histori pembelian untuk kebutuhan audit dan analisis stok.

Gambar 8. Halaman Data Transaksi Pembelian
Sumber : Hasil Olah Data

g. Halaman Transaksi Penjualan

Halaman transaksi penjualan berfungsi untuk mencatat penjualan barang kepada pelanggan, dengan informasi seperti tanggal transaksi, nama pelanggan, daftar barang yang dijual, jumlah, serta total harga penjualan.

Gambar 9. Halaman Data Transaksi Penjualan
Sumber : Hasil Olah Data

h. Halaman Data Training C4.5

Halaman data training c4.5 menampilkan data historis permintaan barang yang digunakan sebagai basis untuk proses pelatihan algoritma C4.5. Setiap entri memuat atribut-atribut seperti nama barang, frekuensi permintaan, rata-rata bulanan, dan kelas kelarisan. Data ini digunakan untuk membentuk pohon keputusan yang akan digunakan dalam proses klasifikasi tingkat kelarisan barang secara otomatis.

No	ID Barang	Nama Barang	Stok	Stok Minimum	Lead Time	Total Penjualan	Rata-rata Penjualan
1	0-0001	Tusuk jeri	79	3	3	93	5.47
2	0-0002	Gigi sakti	98	3	3	136	8.00
3	0-0003	Shending serbuk SLS	53	3	3	129	8.60
4	0-0004	Selap pagoda	73	3	5	74	6.17
5	0-0005	Selap kalpanox	5	3	5	86	6.14
6	0-0006	Gatsby grill	3	3	5	109	8.38
7	0-0007	Payung jumbo pelangi	1	3	5	121	8.07
8	0-0008	Sangobon 4	94	3	5	76	5.43
9	0-0009	Mop 60	69	3	5	80	5.33
10	0-0010	Jas hujan Orisa lengan panjang	61	3	3	70	6.36

Gambar 10. Halaman Data Training C4.5
Sumber : Hasil Olah Data

i. Halaman Proses Klasifikasi C4.5

Halaman proses klasifikasi C4.5 digunakan untuk menjalankan algoritma C4.5 berdasarkan data training yang telah tersedia. Setelah tombol proses ditekan, sistem akan menghitung nilai *entropy* dan *gain* dari setiap atribut, membentuk pohon keputusan, dan menghasilkan output klasifikasi tingkat kelarisan barang seperti "Sangat Laris", "Laris", atau "Tidak Laris". Hasil klasifikasi ini berguna untuk membantu manajemen dalam merencanakan stok dan strategi pengadaan barang.

Step 1: Hasil Pohon Keputusan (Klik untuk sembangkan)
Menentukan aturan dari data ke data berdasarkan atribut tertentu.

Step 2: Proses Entropy dan Gain
Rumus untuk menghitung entropi dan gain dari data.

Step 3: Data Barang Setelah Klasifikasi
Setelah data diolah, kelas barang (Sangat Laris, Laris, Tidak Laris) berdasarkan pohon keputusan.

No	Nama Barang	Stok	Stok Minimum	Lead Time	Total Penjualan	Rata-rata Penjualan	Hasil Kategori
1	Tusuk jeri	79	3	3	93	5.47	Tidak Laris
2	Gigi sakti	98	3	3	136	8.00	Sangat Laris
3	Shending serbuk SLS	53	3	3	129	8.60	Sangat Laris

Gambar 11. Halaman Klasifikasi C4.5
Sumber : Hasil Olah Data

j. Halaman Proses DE

Halaman proses DE digunakan untuk menjalankan proses optimasi pengelolaan stok barang menggunakan algoritma DE. Saat tombol proses ditekan, sistem akan melakukan iterasi terhadap populasi solusi untuk menentukan nilai optimal, seperti jumlah pemesanan barang dan *safety stock*, berdasarkan parameter dan batasan yang telah ditentukan. Hasil dari proses ini bertujuan untuk meminimalkan kekurangan maupun kelebihan stok, sehingga pengelolaan inventaris menjadi lebih efisien dan tepat sasaran.

Step 1: Data Awal Barang
Menampilkan informasi dasar barang dari database.

Tampilkan: 10 data

No.	Nama Barang	Stok	Stok Minimum	Lead Time	Kapasitas Maksimal	Harga Beli
1	Tusuk jari	79	3	3	120	Rp 27.500
2	Gip sakti	98	3	3	120	Rp 18.690
3	Shenting serbuk SLS	53	3	3	120	Rp 33.375
4	Salap pagoda	73	3	5	150	Rp 5.795
5	Salap kalpanax	5	3	5	150	Rp 5.340
6	Getahy gel	3	3	5	150	Rp 4.605
7	Payang jumbo pelangi	1	3	5	150	Rp 25.600
8	Sangoban 4	94	3	5	150	Rp 6.453
9	Hip 60	69	3	5	150	Rp 17.305
10	Jas hujan Office lengan panjang	61	3	3	120	Rp 22.250

Menampilkan 1 sampai 10 dari 256 data

Step 2: Rata-rata & Total Penjualan
Rumus Rata-rata Penjualan: $\text{Rata-rata} = \frac{\text{Total Penjualan}}{\text{Jumlah Bulan}} \times \text{Jumlah Bulan}$

Gambar 12. Halaman Proses DE
Sumber : Hasil Olah Data

k. Halaman Laporan Stok Barang

Halaman laporan stok barang menyajikan rekapitulasi data persediaan barang secara lengkap, mencakup nama barang, jumlah stok tersedia, stok minimum, dan status ketersediaan. Informasi ditampilkan dalam bentuk tabel yang dapat dicetak atau diekspor untuk keperluan dokumentasi dan audit.

Filter Data Stok

Stok:

Laporan Stok Seluruh Barang

Tampilkan: 10 data

No.	ID Barang	Nama Barang	Stok Minimum	Stok
1	B-0001	Tusuk jari	3	79
2	B-0002	Gip sakti	3	98
3	B-0003	Shenting serbuk SLS	3	53
4	B-0004	Salap pagoda	3	73
5	B-0005	Salap kalpanax	3	5
6	B-0006	Getahy gel	3	3
7	B-0007	Payang jumbo pelangi	3	1
8	B-0008	Sangoban 4	3	94
9	B-0009	Hip 60	3	69

Gambar 13. Halaman Laporan Stok Barang
Sumber : Hasil Olah Data

l. Halaman Manajemen User

Data User

Tampilkan: 10 data

No.	Nama User	Username	Hak Akses	Aksi
1	Penjualan	penjualan	Penjualan	Edit Hapus
2	Pembelian	pembelian	Pembelian	Edit Hapus
3	Admin	administrator	Administrator	Edit Hapus

Menampilkan 1 sampai 3 dari 3 data

Gambar 14. Halaman Manajemen User
Sumber : Hasil Olah Data

Halaman ini digunakan untuk mengelola akun pengguna yang memiliki akses ke dalam sistem. Fitur ini memungkinkan penambahan, pengeditan, dan penghapusan akun, serta pengaturan hak akses sesuai peran masing-masing pengguna. Dengan adanya manajemen user, keamanan data dan kontrol terhadap penggunaan sistem dapat terjaga dengan baik.

Hasil pengujian

a. Black Box Testing

Pengujian *Black Box* dilakukan oleh peneliti yang bertujuan untuk memastikan bahwa seluruh fitur aplikasi *inventory* berjalan sesuai fungsi yang dirancang, dengan fokus pada input dan output tanpa melihat struktur kode program. Setiap skenario diuji berdasarkan fungsionalitas antarmuka pengguna, meliputi *login*, pengelolaan data master, transaksi penjualan, serta perhitungan menggunakan algoritma C4.5 dan *Differential Evolution*. Hasil pengujian menunjukkan seluruh fitur, mulai dari proses input, pembaruan, penghapusan, hingga penyimpanan dan perhitungan data, berfungsi sesuai harapan. Adapun hasil pengujian ini adalah dari total 20 skenario uji menunjukkan berhasil sehingga aplikasi *inventory* berjalan sesuai dengan fungsi.

b. Closed Beta Testing

Closed Beta Testing dilakukan kepada 12 orang diantaranya adalah 2 orang ahli IT perusahaan, 5 orang admin gudang serta 5 orang staff operasional yang menghasilkan skor sebagai berikut:

Tabel 2. Tabel Tabulasi Skor Uji *Closed Beta*

Skor Kecapaian	Hasil Skor	Total (Skala*Hasil)
1	0	0
2	1	2
3	10	30
4	33	132
5	17	85
Total skor		249
Skor maksimum		305
Skor maksimum = Jumlah Responden * Jumlah Butir Soal		
* Skala tertinggi		

Hasil yang diperoleh dari hasil tabulasi *closed beta testing* pada tabel 2, kemudian dihitung menggunakan rumus untuk menentukan tingkat kelayakan sistem. Adapun perhitungan kelayakan untuk sistem informasi dan profil desa (SIPROSA) adalah sebagai berikut:

$$\begin{aligned}\text{Presentase Kelayakan} &= \frac{\text{Total skor}}{\text{Skor maksimum}} \times 100 \% \\ &= \frac{249}{305} \times 100 \% = 81,64 \%\end{aligned}$$

Pembahasan

Proses pengembangan sistem *inventory* berbasis web pada Kepinos Warehouse menggunakan model *Rapid Application Development (RAD)* yang bersifat iteratif dan adaptif terhadap perubahan kebutuhan pengguna. Pendekatan ini memungkinkan pengguna terlibat sejak tahap awal untuk memastikan hasil sistem sesuai kebutuhan operasional. Pengembangan dilakukan mulai dari tahap perencanaan kebutuhan, perancangan, hingga implementasi dan pengujian sistem. Hasil observasi menunjukkan bahwa sistem sebelumnya yang berbasis *spreadsheet* masih rentan terhadap kesalahan pencatatan, keterlambatan pelaporan, serta belum memiliki fitur klasifikasi dan rekomendasi pemesanan otomatis. Sistem yang dibangun kemudian mengintegrasikan algoritma C4.5 untuk mengklasifikasikan tingkat kelarisan barang menjadi tiga kategori, yaitu sangat laris, laris, dan tidak laris, sehingga memudahkan pengambilan keputusan pengadaan barang. Sementara itu, algoritma *Differential Evolution (DE)* digunakan untuk memberikan rekomendasi jumlah pemesanan optimal berdasarkan data historis. Hasil pengujian menunjukkan seluruh fungsi berjalan sesuai spesifikasi tanpa kesalahan kritis, serta sistem dinilai layak digunakan dengan beberapa peningkatan yang diperlukan pada aspek tampilan dan navigasi. Secara keseluruhan, sistem yang dikembangkan mampu meningkatkan efisiensi, akurasi, dan adaptivitas dalam pengelolaan persediaan barang di Kepinos Warehouse.

KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa pengembangan aplikasi *inventory* berbasis web pada Kepinos Warehouse dengan metode *Rapid Application Development (RAD)* berhasil dilaksanakan. Penerapan RAD mempercepat proses pengembangan serta meningkatkan keterlibatan pengguna pada setiap tahap. Aplikasi yang dibangun telah mengimplementasikan fitur *safety stock* menggunakan algoritma *Differential Evolution (DE)* dan C4.5 secara efektif untuk mendukung pengambilan keputusan pengelolaan stok. Hasil pengujian *Black Box Testing* menunjukkan seluruh fungsi berjalan sesuai kebutuhan, sedangkan *Closed Beta Testing* membuktikan sistem stabil, mudah digunakan, dan diterima pengguna. Dengan

demikian, aplikasi *inventory* berbasis web yang dikembangkan dinyatakan berhasil dan fungsional sesuai tujuan penelitian.

Untuk pengembangan lebih lanjut, disarankan agar sistem dilengkapi dengan fitur analisis data dan notifikasi cerdas guna membantu pengambilan keputusan secara real-time. Selain itu, perbaikan pada antarmuka pengguna (UI/UX) serta optimalisasi performa dalam pengolahan data berskala besar perlu dilakukan agar sistem menjadi lebih responsif, adaptif, dan siap diterapkan secara luas.

UCAPAN TERIMA KASIH

Penulis mengucapkan syukur ke hadirat Allah SWT atas rahmat dan karunia-Nya sehingga penelitian ini dapat terselesaikan dengan baik. Ucapan terima kasih disampaikan kepada kedua orang tua atas doa dan dukungan yang tiada henti, serta kepada dosen pembimbing atas bimbingan dan arahnya selama proses penelitian. Terima kasih juga diberikan kepada para dosen Program Studi Teknik Informatika dan rekan-rekan yang telah memberikan bantuan, motivasi, serta dukungan dalam penyusunan jurnal ini.

DAFTAR PUSTAKA

- [1] M. W. Tjia and J. Hendrik, "Web-Based Inventory System Design Using Rapid Application Development Method at PT. ALFA SCORPII," *J. Artif. Intell. Eng. Appl.*, vol. 4, no. 3, pp. 2151–2156, 2025.
- [2] BPS(Badan Pusat Statistik Tegal), "Pentingnya sistem inventaris terintegrasi untuk memantau jumlah, nilai, dan kondisi Barang Milik Negara (BMN) secara akurat,," 2021. .
- [3] C. Saha, D. K. Jana, and A. Duary, "Enhancing production inventory management for imperfect items using fuzzy optimization strategies and Differential Evolution (DE) algorithms," *Franklin Open*, vol. 5, p. 100051, 2023.
- [4] H. Mutiara and N. Andayani, "Implementation of Data Mining in Determining Drug Supply Using the C4. 5 Algorithm Case Study at Sambirejo Community Health Center Rejang Lebong Regency," *JANGTEK J. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 1, pp. 37–48, 2024.
- [5] K. Amarullah, R. Astuti, W. Prihartono, and R. Hamonangan, "Penerapan Algoritma C4. 5 untuk Optimalisasi Manajemen Stok Obat di Apotek Nafa Farma," *IJAI (Indonesian J. Appl. Informatics)*, vol. 9, no. 2, pp. 399–407.
- [6] A. Alyshen and R. Harman, "Penerapan Algoritma C4. 5 Untuk Memprediksi Penjualan Barang Pada PT Prima Niaga Indomas," *Comput. Based Inf. Syst. J.*, vol. 12, no. 1, pp. 73–83, 2024.
- [7] S. L. M. Sitio, "Penerapan Metode Rapid Application Development (RAD) Untuk Aplikasi Berbasis Web," vol. 8, no. 1, p. 126 Halaman,

- 2023.
- [8] F. Sudarto and V. N. Shinto, "Perancangan Sistem Informasi Tracking Data Inventory Produksi," *J. Sist. Inf. dan Teknol.*, vol. 4, no. 1, pp. 26–31, 2024.
 - [9] S. Ramingwong, A. Sopadang, K. Y. Tipayawong, and J. Jintana, "Factory Logistics Improvement: A Case Study Analysis of Companies in Northern Thailand, 2022–2024," *Logistics*, vol. 8, no. 3, p. 88, 2024.
 - [10] A. H. Alridha, A. N. H. Mzedawee, and F. H. Abd Alsharify, "Optimization Algorithms for Enhanced Parameter Estimation in Mathematical Modelling," *Indones. J. Math. Appl.*, vol. 2, no. 1, pp. 24–31, 2024.
 - [11] X. Wang and X. Yu, "Differential Evolution Algorithm with Three Mutation Operators for Global Optimization," *Mathematics*, vol. 12, no. 15, p. 2311, 2024.
 - [12] A. Thammano and I. Farda, "An improved differential evolution algorithm for numerical optimization problems," *HighTech Innov. J.*, vol. 4, no. 2, pp. 434–43452, 2023.
 - [13] J.-Y. Li, K.-J. Du, Z.-H. Zhan, H. Wang, and J. Zhang, "Distributed differential evolution with adaptive resource allocation," *IEEE Trans. Cybern.*, vol. 53, no. 5, pp. 2791–2804, 2022.
 - [14] J. Shanthi, D. G. N. Rani, and S. Rajaram, "A C4.5 decision tree classifier based floorplanning algorithm for System-on-Chip design," *Microelectronics J.*, vol. 121, p. 105361, 2022.
 - [15] M. Bagriacik and F. E. B. Otero, "Multiple fairness criteria in decision tree learning," *Appl. Soft Comput.*, vol. 167, p. 112313, 2024.
 - [16] S. Desikan and G. Ramesh, *Software testing: principles and practice*. Pearson Education India, 2006.
 - [17] M. Nuraminudin, M. M. Dewi, S. Suharsono, A. Dahlan, and L. Lukman, "Implementasi Reactjs Pada Pembuatan Sistem Informasi Digital Printing Berbasis Website," *Inf. Syst. J.*, vol. 6, no. 01, pp. 25–32, 2023.
 - [18] A. C. Praniffa, A. Syahri, F. Sandes, U. Fariha, Q. A. Giansyah, and M. L. Hamzah, "Pengujian Black Box Dan White Box Sistem Informasi Parkir Berbasis Web Black Box and White Box Testing of Web-Based Parking Information System," *J. Test. dan Implementasi Sist. Inf.*, vol. 1, no. 1, pp. 1–16, 2023.
 - [19] F. N. Khasanah, D. T. Untari, D. Nurmanto, B. Satria, T. Sukreni, and T. S. Perdhana, "Beta Testing Techniques in Non-Functional Testing of Gamified Learning Applications for Lecture Learning Media During the Covid-19 Pandemic," *J. Internet Serv. Inf. Secur.*, vol. 12, no. 4, pp. 197–203, 2022.
 - [20] H. L. Hakim, D. Faqih, D. Deva, I. F. Hudaya, and M. N. Ilyas, "Pengujian Alpha dan Beta Testing Pada Aplikasi TIJE," *TeknoIS J. Ilm. Teknol. Inf. dan Sains*, vol. 14, no. 2, pp. 285–295, 2024.