

**LAPORAN TUGAS BESAR SMART INVENTORY BOX DENGAN
KECERDASAN BUATAN REGRESI**



Disusun Oleh:

Lintang Layli Pangestu

122490037

Dosen Pengampu:

Ronal, M.Kom

Rizki Yustisia Sari, M.T.I

REKAYASA INSTRUMENTASI DAN AUTOMASI

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI SUMATERA

2025

DAFTAR ISI

DAFTAR ISI	i
BAB I: PENDAHULUAN	1
1.1 Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan	2
1.4. Manfaat	2
1.5. Ruang Lingkup	3
BAB II: LANDASAN TEORI	4
2.1. Studi Literatur	4
2.2. GUI dan tkinter	5
2.3. Konsep Smart Inventory Box	6
BAB III: METODOLOGI	8
3.1. Metode Pengumpulan Data	8
3.2. Desain Sistem dan Alur Data	8
3.2.1. Blok Diagram	8
3.3. Perangkat Keras dan Lunak	9
3.4. Langkah Pengerjaan Proyek	10
BAB IV: IMPLEMENTASI DAN HASIL	12
4.1. Implementasi Sistem	12
4.2. Tampilan GUI	13
4.3. Simulasi dan Uji Coba	14
4.4. Hasil Evaluasi Model	16
BAB V: KESIMPULAN DAN REKOMENDASI	18
5.1. Kesimpulan	18
5.2. Rekomendasi	18

BAB I: PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi telah mendorong berbagai sektor industry untuk mengadopsi sistem digital dalam mengelola data dan proses bisnis mereka. Salah satu aspek penting dalam manajemen rantai pasok dan logistik adalah pengelolaan inventori atau persediaan barang. Sistem inventori tradisional yang bergantung pada pencatatan manual seringkali menghadapi berbagai tantangan seperti keterlambatan dalam deteksi kehabisan stok, ketidaktepatan data, serta kesulitan dalam memprediksi kebutuhan di masa depan. Oleh karena itu, diperlukan pendekatan berbasis teknologi yang lebih akurat, efisien, dan dapat diakses oleh berbagai pihak.

Salah satu pendekatan yang kini banyak digunakan adalah penerapan machine learning dalam prediksi dan klasifikasi data inventori. Machine learning memungkinkan sistem untuk belajar dari data historis dan membuat prediksi tanpa perlu diprogram secara eksplisit untuk setiap kemungkinan. Dalam proyek ini, digunakan dua jenis algoritma supervised learning yang populer, yaitu Linear Regression dan Logistic Regression. Linear Regression digunakan untuk memprediksi nilai kontinu, seperti estimasi jumlah pengambilan barang, sedangkan Logistic Regression digunakan untuk klasifikasi, seperti menentukan apakah suatu barang kemungkinan besar akan habis atau tidak.

Penggunaan antarmuka grafis (Graphical User Interface/GUI) berbasis Python melalui library Tkinter juga menjadi nilai tambah dalam proyek ini. GUI dirancang agar pengguna dari berbagai latar belakang, termasuk yang tidak memiliki kemampuan teknis tinggi, dapat dengan mudah mengakses, melatih, dan mengevaluasi model prediksi inventori hanya melalui tampilan yang interaktif dan intuitif.

Dengan latar belakang tersebut, proyek ini bertujuan untuk membangun sistem prediksi inventori yang tidak hanya cerdas, tetapi juga mudah diakses dan digunakan oleh pemangku kepentingan di lingkungan organisasi atau bisnis.

Dengan latar belakang tersebut, proyek ini bertujuan untuk membangun sistem prediksi inventori yang tidak hanya cerdas, tetapi juga mudah diakses dan digunakan oleh pemangku kepentingan di lingkungan organisasi atau bisnis. Manajemen inventori yang efektif menjadi kebutuhan penting dalam sistem logistik modern. Prediksi kehabisan stok dan identifikasi pola pengambilan barang memungkinkan organisasi mengantisipasi kebutuhan secara tepat waktu. Untuk mendukung hal ini, diterapkan pendekatan machine learning dengan algoritma Linear Regression dan Logistic Regression dalam sistem GUI berbasis Python.

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah yang diangkat dalam tugas besar ini adalah sebagai berikut:

1. Bagaimana merancang sistem prediksi stok berbasis AI dengan tampilan GUI interaktif?
2. Bagaimana membandingkan kinerja algoritma linear regression dan logistic regression dalam konteks inventori?

1.3. Tujuan

Tujuan dari pengerjaan tugas besar ini adalah sebagai berikut:

1. Membangun sistem berbasis GUI untuk prediksi inventori
2. Mengimplementasikan model AI untuk klasifikasi dan regresi
3. Menampilkan hasil evaluasi model dalam format visual dan numerik

1.4. Manfaat

Manfaat bagi pelaksanaan tugas besar ini dapat dirinci sebagai berikut:

1. Meningkatkan efisiensi pengelolaan barang
2. Memberikan insight data pengambilan barang secara otomatis
3. Memberikan alat bantu visual bagi pengguna non-teknis

1.5. Ruang Lingkup

1. Dataset bersumber dari file excel data dummy
2. Model AI dibatasi pada logistic dan linear regression
3. GUI dibangun menggunakan tkinter

BAB II: LANDASAN TEORI

2.1. Studi Literatur

Dalam dunia machine learning dan data science, dua algoritma dasar yang sering digunakan untuk prediksi adalah Linear Regression dan Logistic Regression. Keduanya tergolong dalam kategori supervised learning, di mana model dilatih berdasarkan data berlabel. Meski memiliki nama yang serupa, tujuan penggunaan dan cara kerjanya sangat berbeda.

Linear Regression adalah metode prediksi yang digunakan untuk memodelkan hubungan antara satu atau lebih variabel independen (fitur) dengan satu variabel dependen yang bersifat kontinu. Model ini menghasilkan garis lurus terbaik yang dapat memperkirakan nilai target numerik berdasarkan nilai-nilai input. Linear Regression digunakan dalam berbagai aplikasi seperti prediksi harga, permintaan, atau tren waktu.

Logistic Regression, di sisi lain, digunakan ketika variabel target bersifat kategorikal, khususnya untuk masalah klasifikasi biner (contoh: 0 atau 1). Alih-alih memprediksi nilai langsung, Logistic Regression menghasilkan probabilitas menggunakan fungsi sigmoid, kemudian menentukan kelas target berdasarkan ambang batas tertentu (biasanya 0.5). Ini sangat efektif dalam kasus seperti deteksi stok habis, deteksi penipuan, atau keputusan ya/tidak.

Berikut ini tabel perbandingan antara Logistic Regression dan Linear Regression secara lebih terstruktur dan ringkas:

Aspek	Logistik Regression	Linear Regression
Tujuan	Klasifikasi biner (contoh: ya/tidak)	Prediksi nilai kontinu
Fungsi Aktivasi	Fungsi sigmoid (menghasilkan probabilitas antara 0-1)	Tidak menggunakan fungsi aktivasi khusus
Output	Probabilitas dikonversi ke kelas (0 atau 1)	Nilai numerik (Kontinu)

Tipe	klasifikasi	Regresi
Masalah		
Fungsi	Log-loss/binary Cross-Entropy	Mean Squared Error (MSE)
Kerugian		
Contoh	Deteksi fraud, prediksi stok habis	Estimasi permintaan, prediksi
Aplikasi		harga

Linear Regression digunakan untuk memodelkan hubungan antara satu atau lebih variabel independen dengan variabel dependen yang bersifat kontinu. Tujuan dari metode ini adalah untuk membentuk garis regresi yang merepresentasikan tren umum data. Dalam konteks sistem prediksi inventori, Linear Regression digunakan untuk memperkirakan jumlah pengambilan barang di masa depan berdasarkan data historis. Model ini berguna untuk menganalisis tren jangka panjang seperti menurunnya stok dari waktu ke waktu.

Sementara itu, Logistic Regression digunakan untuk menangani masalah klasifikasi, terutama klasifikasi biner. Model ini memanfaatkan fungsi sigmoid untuk menghasilkan nilai probabilitas yang kemudian dikonversi menjadi kategori (misalnya: stok "akan habis" atau "tidak habis"). Logistic Regression sangat sesuai digunakan untuk pengambilan keputusan berbasis ambang batas (threshold) tertentu, sehingga berguna dalam sistem inventori untuk mendeteksi situasi kritis seperti potensi kehabisan barang.

Pemahaman yang mendalam terhadap kedua algoritma ini sangat penting karena pemilihan model yang tepat dapat meningkatkan akurasi prediksi dan efisiensi sistem. Selain itu, kedua metode tersebut bersifat ringan secara komputasi dan relatif mudah diimplementasikan dengan menggunakan pustaka machine learning populer seperti scikit-learn

2.2. GUI dan tkinter

proyek ini, Tkinter digunakan sebagai kerangka kerja utama untuk membangun tampilan interaktif yang dapat digunakan oleh pengguna dari berbagai latar

belakang, termasuk mereka yang tidak memiliki keahlian teknis dalam pemrograman atau data science.

Penggunaan GUI bertujuan untuk menyederhanakan interaksi antara pengguna dan sistem prediksi. Melalui antarmuka ini, pengguna dapat memilih model prediksi, mengunggah dataset, menjalankan pelatihan (*training*), serta melihat hasil evaluasi model dalam bentuk visual seperti confusion matrix dan nilai metrik evaluasi (misalnya akurasi, precision, recall, dan F1 score). Dalam implementasinya, GUI ini terbagi ke dalam tiga tab utama, yaitu: **Dashboard** (untuk melihat data stok dan pengguna aktif), **Training** (untuk memilih dan melatih model), serta **Evaluasi** (untuk menampilkan hasil analisis).

Tkinter juga mendukung integrasi dengan pustaka visualisasi seperti matplotlib, memungkinkan sistem untuk menampilkan grafik dalam jendela aplikasi secara langsung. Ini menjadi nilai tambah karena pengguna tidak perlu berpindah ke perangkat lunak lain untuk melihat hasil evaluasi.

Penerapan Tkinter dalam proyek ini membuktikan bahwa antarmuka berbasis Python dapat dikembangkan dengan cepat, fleksibel, dan tetap cukup andal untuk penggunaan prototipe maupun aplikasi internal berskala kecil hingga menengah.

2.3. Konsep Smart Inventory Box

Smart Inventory Box adalah solusi teknologi modern dalam pengelolaan persediaan barang yang menggabungkan elemen perangkat keras dan perangkat lunak untuk menciptakan sistem otomatis, responsif, dan prediktif. Konsep ini muncul sebagai respons terhadap kebutuhan industri yang menuntut efisiensi dan akurasi tinggi dalam pengelolaan logistik dan inventaris.

Secara umum, Smart Inventory Box terdiri dari beberapa komponen utama:

- **Sensor Load Cell** untuk mengukur berat barang secara real-time, yang kemudian digunakan untuk memperkirakan jumlah barang yang tersisa di dalam kotak.

- **Sensor RFID** untuk mengidentifikasi pengguna atau barang secara otomatis saat terjadi pengambilan atau pengisian barang.
- **Mikrokontroler ESP32** sebagai unit pemrosesan utama untuk mengolah data dari sensor.
- **Model AI atau algoritma prediksi**, yang berfungsi untuk menganalisis data historis dan memproyeksikan status stok di masa depan.

Tujuan utama dari penerapan Smart Inventory Box adalah untuk **mengurangi intervensi manual**, meminimalkan kesalahan pencatatan, serta memberikan peringatan dini terhadap kondisi stok yang kritis. Selain itu, sistem ini dapat disesuaikan untuk berbagai skenario, mulai dari gudang logistik, kantor, hingga kebutuhan rumah tangga.

Dengan mengintegrasikan sensor-sensor dengan sistem prediksi berbasis machine learning serta tampilan GUI, Smart Inventory Box menawarkan pendekatan **end-to-end** dalam manajemen inventori. Pendekatan ini tidak hanya menyederhanakan proses pemantauan stok, tetapi juga mendukung pengambilan keputusan yang berbasis data secara lebih akurat dan real-time.

BAB III: METODOLOGI

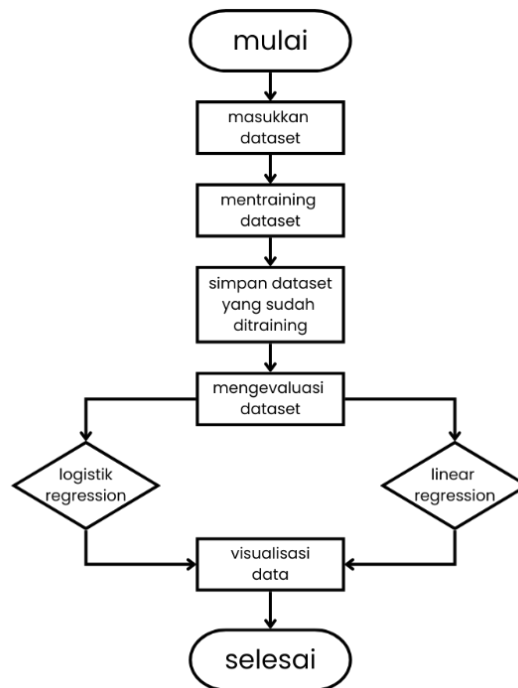
3.1. Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam proyek ini bersifat simulatif menggunakan dataset dummy yang disusun dalam format Excel. Dataset ini berisi data historis pengambilan barang yang mencakup informasi seperti timestamp (waktu pengambilan), nomor kartu RFID yang mewakili identitas pengguna, nama barang yang diambil, jumlah barang, dan berat akhir dari persediaan. Penggunaan data dummy ini bertujuan untuk menguji alur sistem dan algoritma prediksi dalam skenario yang realistis namun terkendali. Dengan format data yang terstruktur, proses preprocessing dan pelatihan model dapat dilakukan dengan lebih mudah dan sistematis.

3.2. Desain Sistem dan Alur Data

3.2.1. Blok Diagram

Sistem dirancang untuk memproses data input dari file Excel, melakukan pelatihan model machine learning (baik regresi maupun klasifikasi), dan menampilkan hasil prediksi serta evaluasi dalam bentuk visual. Alur data dimulai dari pemilihan file dataset oleh pengguna, kemudian sistem akan menjalankan tahap preprocessing data yang mencakup pembersihan, normalisasi, dan pemisahan data menjadi data pelatihan dan pengujian. Selanjutnya, pengguna dapat memilih model (Logistic Regression atau Linear Regression) dan menjalankan proses pelatihan. Setelah model dilatih, sistem akan menampilkan visualisasi hasilnya melalui antarmuka GUI.



3.3. Perangkat Keras dan Lunak

Proyek ini dikembangkan menggunakan perangkat keras berupa laptop. Dari sisi perangkat lunak, sistem dibangun menggunakan bahasa pemrograman Python versi 3.13.2 Beberapa pustaka pendukung utama yang digunakan antara lain: tkinter, scikit-learn, pandas, matplotlib, dan joblib. Konfigurasi perangkat lunak yang ringan ini memungkinkan sistem dijalankan pada perangkat komputer standar tanpa memerlukan infrastruktur khusus. berikut adalah penjelasan dari masing-masing teknologi utama yang digunakan:

- **Python**

Python merupakan bahasa pemrograman utama yang digunakan dalam proyek ini. Bahasa ini populer dalam pengembangan sistem berbasis data dan machine learning karena sintaksnya yang sederhana serta ekosistem pustaka yang sangat lengkap. Python mendukung berbagai paradigma pemrograman dan sangat efisien dalam mengelola alur data, pelatihan model, hingga visualisasi.

- **Tkinter**

Tkinter adalah pustaka GUI standar yang disediakan oleh Python untuk

membangun antarmuka grafis. Dalam proyek ini, Tkinter digunakan untuk membuat aplikasi desktop yang interaktif, memungkinkan pengguna memilih model, melatih model, dan melihat hasil evaluasi tanpa harus menulis kode secara langsung. Keunggulan Tkinter terletak pada kesederhanaannya serta kemampuannya berintegrasi dengan pustaka Python lainnya.

- **Scikit-learn**

scikit-learn adalah pustaka machine learning Python yang menyediakan berbagai algoritma untuk klasifikasi, regresi, dan clustering. Dalam proyek ini, pustaka ini digunakan untuk mengimplementasikan **Logistic Regression** dan **Linear Regression**. Selain itu, scikit-learn juga menyediakan alat evaluasi seperti confusion matrix dan metrik performa (accuracy, precision, recall, dll), serta fungsi pembagian data seperti `train_test_split`.

- **pandas**

pandas digunakan untuk manipulasi dan analisis data dalam format tabular (seperti file Excel atau CSV). Dalam sistem ini, pandas berfungsi membaca dataset, membersihkan data, dan menyiapkan input untuk model machine learning. Kemampuan pandas untuk mengelola data dengan efisien menjadikannya fondasi penting dalam alur preprocessing.

- **matplotlib**

matplotlib adalah pustaka visualisasi data. Visualisasi ini ditampilkan melalui GUI agar memudahkan pengguna dalam memahami performa model tanpa harus melihat data mentah atau nilai numerik saja.

3.4. Langkah Pengerjaan Proyek

Langkah pengerjaan proyek ini dimulai dari tahap perancangan antarmuka GUI menggunakan Tkinter, diikuti dengan pengembangan fungsi untuk pemrosesan data dan pelatihan model. Setelah itu, dilakukan integrasi antar komponen sehingga pengguna dapat mengunggah dataset, memilih model, melatih model, dan melihat hasil evaluasi secara langsung dari GUI. Pengujian dilakukan dengan menggunakan

dataset dummy untuk memastikan bahwa seluruh fungsi berjalan dengan baik, mulai dari input hingga output hasil evaluasi. Proyek ini juga mencakup penyimpanan hasil evaluasi model dalam bentuk gambar untuk dokumentasi dan analisis lebih lanjut.

BAB IV: IMPLEMENTASI DAN HASIL

4.1. Implementasi Sistem

Implementasi sistem dilakukan dengan membangun antarmuka berbasis Tkinter yang terdiri dari beberapa tab, yaitu Dashboard, Training, dan Evaluasi. Setiap tab memiliki fungsi spesifik: tab Dashboard menampilkan data stok dan pengguna aktif; tab Training memungkinkan pengguna untuk memilih algoritma dan melatih model; sedangkan tab Evaluasi menyajikan hasil evaluasi model dalam bentuk numerik dan visual. Proses pembagian data dilakukan menggunakan fungsi `'train_test_split'` dari pustaka `scikit-learn` dengan rasio 80% untuk pelatihan dan 20% untuk pengujian.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,  
random_state=42)
```

Pembagian ini bertujuan untuk menguji generalisasi model terhadap data yang belum pernah dilihat sebelumnya.

Proses pelatihan model dalam sistem dilakukan secara otomatis berdasarkan pilihan pengguna pada antarmuka GUI. Kode berikut digunakan untuk menentukan model yang akan digunakan dan menjalankan pelatihan terhadap data yang telah diproses:

```
if selected_model.get() == "Logistic Regression":
```

```
    model = LogisticRegression(max_iter=1000)
```

```
else:
```

```
    model = LinearRegression()
```

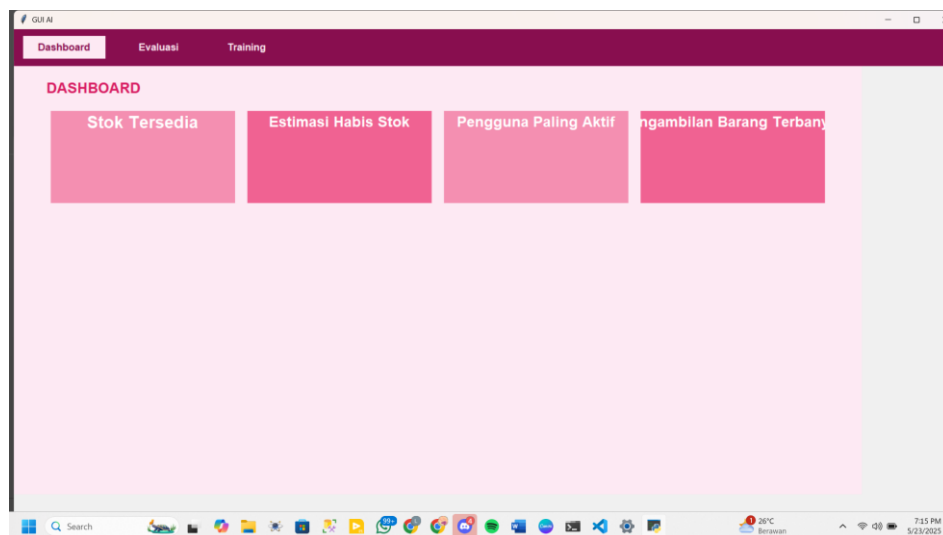
```
model.fit(X_train, y_train)
```

Potongan kode di atas menggunakan logika *conditional* untuk memilih jenis algoritma berdasarkan input pengguna dari GUI. Jika pengguna memilih opsi "Logistic Regression", maka sistem akan menginisialisasi model klasifikasi dengan `LogisticRegression(max_iter=1000)`. Parameter `max_iter=1000` digunakan untuk memastikan proses iterasi cukup panjang agar model dapat konvergen, terutama pada dataset yang kompleks atau bervolume besar.

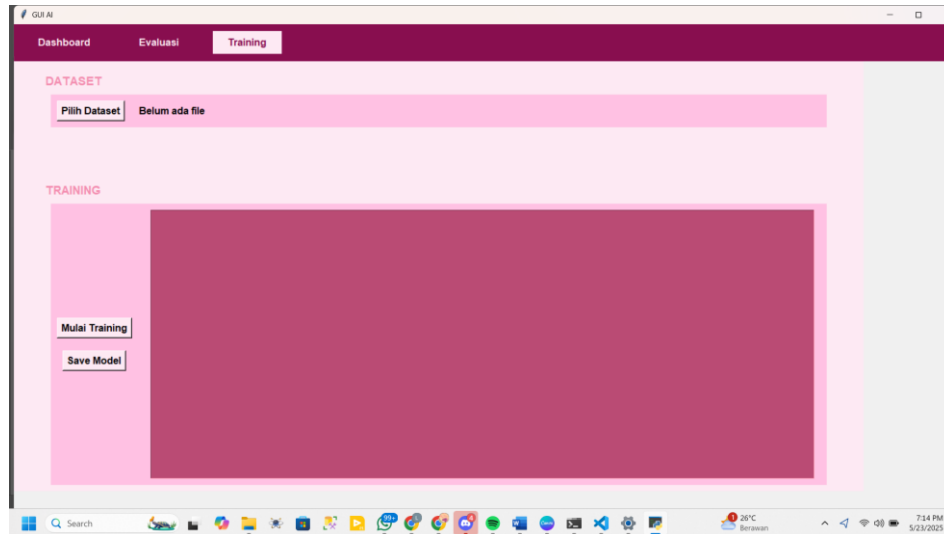
Sebaliknya, jika pengguna memilih model lain (yaitu Linear Regression), maka sistem akan menggunakan `LinearRegression()` sebagai model prediksi numerik. Setelah model dipilih dan dibentuk, proses pelatihan dilakukan menggunakan `model.fit(X_train, y_train)`, di mana `X_train` adalah data fitur dan `y_train` adalah data target hasil pelabelan. Fungsi ini akan melatih model berdasarkan pola dalam data agar dapat digunakan untuk prediksi atau klasifikasi terhadap data baru pada tahap selanjutnya.

4.2. Tampilan GUI

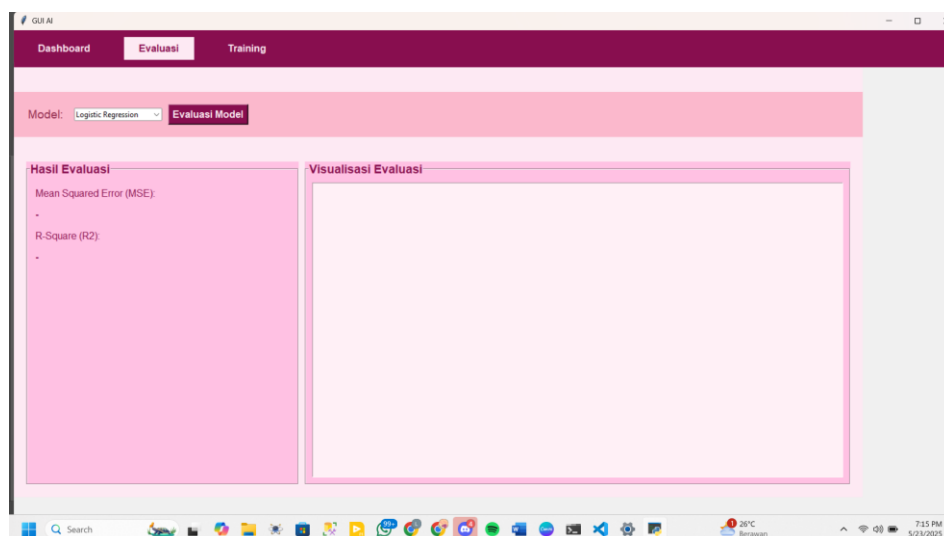
Antarmuka pengguna (GUI) yang dikembangkan bertujuan untuk memberikan pengalaman interaktif dan ramah pengguna. Desain GUI dibagi menjadi tiga bagian utama, yaitu: (1) Dashboard, yang menampilkan informasi ringkas mengenai data inventori; (2) Tab Training, yang memungkinkan pengguna memilih model (Linear atau Logistic Regression), dan menjalankan proses pelatihan; serta (3) Tab Evaluasi, yang menampilkan hasil metrik performa model seperti akurasi, precision, recall. Seluruh fitur ini disusun secara intuitif agar mudah dipahami oleh pengguna non-teknis.



Gambar 4.2.1 tampilan tab dashboard



Gambar 4.2.2 tampilan tab training

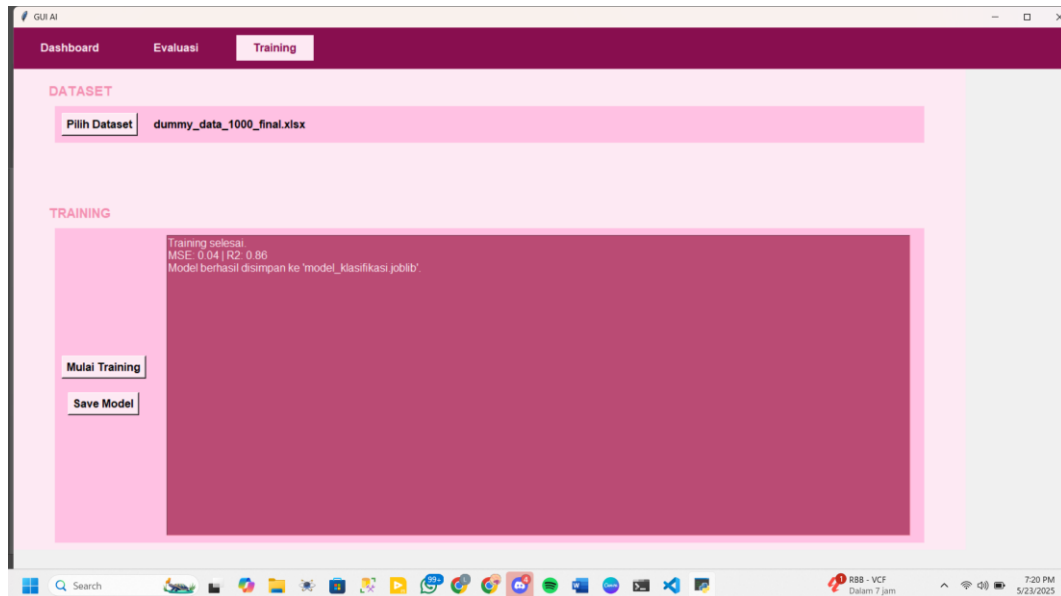


Gambar 4.2.3 tampilan tab evaluasi

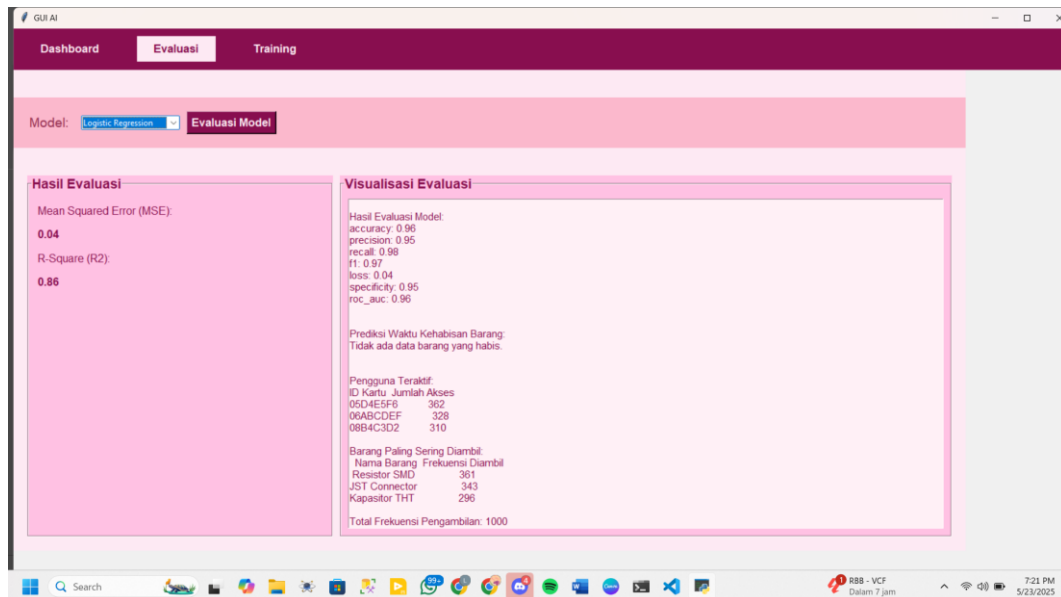
4.3. Simulasi dan Uji Coba

Pengujian dilakukan menggunakan dataset dummy yang terdiri dari 300 baris data. Dataset ini disusun secara acak namun realistis untuk meniru pola interaksi pengguna terhadap sistem inventori. Model diuji dengan metode pembagian data 80:20, di mana 80% data digunakan untuk pelatihan dan 20% sisanya untuk

pengujian. Simulasi dilakukan berulang untuk mengevaluasi konsistensi performa model dan kestabilan sistem dalam menangani proses pelatihan dan prediksi.



Gambar 4.3.1 tampilan saat model ditraining



Gambar 4.3.2 tampilan saat model dievaluasi

4.4. Hasil Evaluasi Model

Setelah proses pelatihan dan pengujian dilakukan, sistem menghasilkan berbagai metrik evaluasi untuk menilai performa model prediksi, baik untuk klasifikasi (Logistic Regression) maupun regresi (Linear Regression). Berikut adalah penjelasan masing-masing metrik yang digunakan:

- **Accuracy (0.96)**

Merupakan proporsi prediksi yang benar dibandingkan dengan seluruh prediksi yang dilakukan. Nilai 0.96 berarti bahwa 96% dari total prediksi klasifikasi berhasil dilakukan dengan benar oleh model Logistic Regression. Metrik ini sangat berguna sebagai ukuran umum performa model, tetapi bisa menyesatkan jika data tidak seimbang.

- **Precision (0.95)**

Precision mengukur seberapa akurat model saat memprediksi kelas positif (misalnya, stok akan habis). Nilai 0.95 menunjukkan bahwa 95% dari prediksi positif yang dibuat oleh model benar-benar merupakan kasus positif. Metrik ini penting ketika kesalahan positif palsu (false positive) perlu diminimalkan.

- **Recall (0.98)**

Recall menunjukkan seberapa baik model dalam menangkap semua kasus positif dari total yang sebenarnya positif. Dengan nilai 0.98, berarti 98% dari semua kejadian stok habis berhasil dideteksi oleh model. Ini sangat penting dalam konteks prediksi inventori karena kegagalan dalam mendeteksi stok habis bisa berdampak signifikan.

- **F1 Score (0.97)**

F1 Score adalah rata-rata harmonik antara precision dan recall, dan digunakan sebagai metrik yang menyeimbangkan keduanya. Dengan nilai 0.97, model memiliki keseimbangan yang baik antara kemampuan mengidentifikasi kasus positif dan menghindari prediksi yang salah.

- **Mean Squared Error – MSE (0.04)**

MSE digunakan untuk mengevaluasi model regresi (Logistik Regression) dan menunjukkan rata-rata dari kuadrat selisih antara nilai prediksi dan nilai aktual. Nilai MSE yang rendah (0.04) menunjukkan bahwa kesalahan prediksi model relatif kecil.

- **R² Score (0.86)**

Juga dikenal sebagai koefisien determinasi, R² Score mengukur seberapa besar variasi dalam data target dapat dijelaskan oleh model. Nilai 0.79 berarti 79% variasi dalam data dapat dijelaskan oleh Linear Regression, yang menandakan model memiliki performa yang cukup baik dalam konteks prediksi numerik.

- **ROC AUC (0.96)**

ROC AUC (Receiver Operating Characteristic – Area Under Curve) mengukur kemampuan model klasifikasi dalam membedakan antara kelas positif dan negatif. Nilai 0.96 menandakan bahwa model memiliki tingkat diskriminasi yang sangat tinggi dan mampu membedakan dengan baik antara stok yang akan habis dan yang tidak.

Secara keseluruhan, hasil evaluasi menunjukkan bahwa kedua model – baik klasifikasi maupun regresi – bekerja dengan baik dalam konteks prediksi inventori, dengan Logistic Regression unggul dalam deteksi kondisi stok dan Linear Regression efektif dalam estimasi kuantitatif pengambilan barang.

BAB V: KESIMPULAN DAN REKOMENDASI

5.1. Kesimpulan

Berdasarkan hasil implementasi dan pengujian pada proyek Smart Inventory Box yang mengintegrasikan algoritma regresi dengan antarmuka GUI berbasis Tkinter, dapat disimpulkan bahwa sistem berhasil memenuhi tujuan utamanya, yaitu melakukan prediksi stok dan pengelolaan inventori secara otomatis dan interaktif.

Dengan antarmuka GUI yang ramah pengguna, sistem memungkinkan pengguna dari berbagai latar belakang untuk mengoperasikan fitur-fitur penting seperti pemilihan model, pelatihan data, dan evaluasi performa tanpa perlu menulis kode secara manual. Hasil evaluasi model menunjukkan performa yang sangat baik, dengan nilai akurasi mencapai 96%, recall 98%, dan F1-score 97% untuk klasifikasi stok. Untuk regresi, model menghasilkan nilai R^2 sebesar 0.86 dan MSE yang sangat rendah yaitu 0.04, menandakan bahwa prediksi sistem mendekati nilai aktual.

Selain itu, integrasi antara load cell, RFID, dan ESP32 dalam konsep Smart Inventory Box mendukung implementasi end-to-end yang praktis dan dapat diterapkan di berbagai skenario dunia nyata seperti gudang logistik, kantor, atau bahkan rumah tangga.

5.2. Rekomendasi

Untuk pengembangan sistem ke depannya, berikut beberapa saran yang dapat dijadikan pertimbangan:

1. **Penggunaan Dataset Nyata:** Disarankan untuk menguji sistem dengan data aktual dari lingkungan kerja yang sesungguhnya untuk mendapatkan hasil evaluasi yang lebih relevan dan representatif.
2. **Penambahan Algoritma Lain:** Sistem dapat ditingkatkan dengan menambahkan algoritma lain seperti Random Forest, Decision Tree, atau bahkan model deep learning untuk menangani data yang lebih kompleks.

3. **Integrasi Cloud dan IoT:** Sistem dapat diperluas dengan integrasi ke platform cloud untuk penyimpanan data dan dashboard monitoring online, serta integrasi IoT untuk pengiriman data sensor secara real-time.
4. **Fitur Notifikasi dan Otomasi:** Dapat ditambahkan fitur notifikasi otomatis (misalnya melalui email atau aplikasi mobile) ketika stok mendekati habis, serta sistem otomatisasi pengisian ulang barang berdasarkan prediksi.
5. **Pengembangan Tampilan GUI:** Tampilan GUI dapat dikembangkan lebih lanjut dengan antarmuka yang lebih modern menggunakan framework tambahan seperti PyQt atau web-based GUI seperti Streamlit atau Flask.

Dengan pengembangan yang berkelanjutan, sistem Smart Inventory Box ini berpotensi menjadi solusi cerdas dan efisien untuk berbagai kebutuhan pengelolaan persediaan barang secara modern dan berbasis data.