

LAPORAN TUGAS BESAR KECERDASAN BUATAN

Sistem Smart Feeder Udang



Dibuat Oleh :

Monica Reza Faulina 122490007

Dosen Pengampu :

Ronal, M.Kom. dan Rizki Yustisia Sari, S.T., M.T.I.

PROGRAM STUDI REKAYASA INSTRUMENTASI DAN AUTOMASI

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI SUMATERA

2025

DAFTAR ISI

DAFTAR ISI.....	2
BAB I	4
PENDAHULUAN	4
1.1 Latar Belakang	4
1.2 Tujuan.....	4
1.3 Manfaat.....	5
BAB II.....	6
LANDASAN TEORI.....	6
2.1 Teori Dasar	6
2.1.1 Smart Feeder	6
2.1.2 Algoritma Decision Tree	6
2.1.3 Visual Studio Code (VSCode).....	6
2.2 Parameter dalam Pemberian Pakan Udang	7
BAB III	8
METODOLOGI	8
3.1 Alur Pengembangan Sistem	8
3.2 Pembuatan Data Set.....	8
3.3 Pra-pemrosesan Data	9
3.4 Pelatihan Model Decision Tree	9
3.5 Evaluasi Model.....	9
3.6 Pembuatan Antarmuka (GUI)	9
BAB IV	11
IMPLEMENTASI DAN PENGUJIAN	11
4.1 Implementasi	11

4.2 Pengujian Model.....	11
4.3 Analisis Hasil	12
4.4 Pembahasan	16
BAB V.....	18
KESIMPULAN.....	18
5.1 Kesimpulan.....	18
5.2 Saran.....	18
LAMPIRAN.....	19

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi di bidang pertanian dan perikanan, khususnya akuakultur semakin pesat dengan penerapan otomatisasi untuk meningkatkan efisiensi dan produktivitas. Pada budidaya udang, pemberian pakan yang tepat sangat penting untuk kesehatan dan pertumbuhan udang, namun seringkali pemborosan pakan terjadi jika tidak terkelola dengan baik.

Salah satu solusi untuk mengatasi masalah ini adalah dengan mengembangkan Sistem Smart Feeder Udang, yang menggunakan algoritma Decision Tree untuk mengatur pemberian pakan berdasarkan kondisi lingkungan seperti suhu air, pH, dan pemberian pakan udang. Sistem ini secara otomatis memberikan pakan yang sesuai dengan kebutuhan spesifik udang, sehingga dapat mengoptimalkan pertumbuhan dan mengurangi pemborosan.

Sistem ini juga dilengkapi dengan antarmuka pengguna GUI (*Graphical User Interface*) yang memudahkan operator dalam mengontrol dan memantau pemberian pakan, serta menyimpan data ke dalam format Excel untuk analisis lebih lanjut. Diharapkan, sistem ini dapat meningkatkan efisiensi operasional, kualitas budidaya udang, dan mengurangi ketergantungan pada pemberian pakan manual.

1.2 Tujuan

- Mengembangkan sistem otomatisasi pemberian pakan udang berbasis *machine learning*.
- Mengimplementasikan algoritma Decision Tree untuk memprediksi waktu dan jumlah pakan.
- Menyediakan antarmuka pengguna (GUI) untuk memantau dan mengontrol pemberian pakan secara real-time.

1.3 Manfaat

- Sistem dapat memutuskan waktu dan jumlah pakan yang optimal berdasarkan kondisi nyata (umur udang, suhu air, dan sisa pakan), sehingga mengurangi pemborosan dan overfeeding.
- Dengan pemberian pakan yang tepat waktu dan sesuai kebutuhan, pertumbuhan udang menjadi lebih optimal, yang berkontribusi pada peningkatan hasil panen.
- Otomatisasi proses pemberian pakan mengurangi ketergantungan pada tenaga kerja manusia, sehingga menghemat biaya dalam jangka panjang.

BAB II

LANDASAN TEORI

2.1 Teori Dasar

2.1.1 Smart Feeder

Smart feeder adalah sistem otomatis yang digunakan untuk memberikan pakan pada hewan atau ikan secara terjadwal dan berbasis data. Dalam konteks budidaya udang, sistem ini dapat mengatur waktu, frekuensi, dan jumlah pakan yang diberikan.

2.1.2 Algoritma Decision Tree

Algoritma decision tree merupakan salah satu algoritma *supervised learning* yang digunakan untuk klasifikasi dan regresi. Algoritma ini bekerja dengan membagi dataset berdasarkan fitur yang memberikan *information gain* tertinggi pada setiap percabangan, sehingga membentuk pohon keputusan. Keuntungan utama decision tree adalah kemampuannya dalam memberikan interpretasi logika yang jelas dan mudah dipahami.

Kelebihan:

- Mudah dipahami dan divisualisasikan.
- Tidak memerlukan skala data atau normalisasi.
- Dapat digunakan untuk data kategorik dan numerik.

Kekurangan:

- Rentan terhadap overfitting.
- Sensitif terhadap perubahan kecil pada data

2.1.3 Visual Studio Code (VSCode)

Visual Studio Code (VSCode) adalah editor kode sumber (source code editor) yang ringan, gratis, dan sangat populer di kalangan pengembang perangkat lunak dan data scientist. VSCode mendukung banyak bahasa pemrograman termasuk Python, dan menyediakan fitur-fitur seperti debugging, integrasi terminal, ekstensi untuk machine learning, dan

manajemen proyek yang memudahkan pengembangan sistem Smart Feeder berbasis Decision Tree. VSCode digunakan dalam pengembangan sistem ini untuk menulis, menjalankan, dan menguji kode program secara efisien.

2.2 Parameter dalam Pemberian Pakan Udang

Beberapa parameter penting yang perlu diperhatikan dalam sistem pemberian pakan udang antara lain:

- **Umur Udang:** Merupakan indikator utama kebutuhan nutrisi. Udang pada fase awal membutuhkan pakan lebih sering dalam jumlah kecil, sedangkan saat mendekati panen, kebutuhan pakan meningkat seiring pertumbuhan ukuran tubuh.
- **Suhu Air:** Suhu air berpengaruh terhadap metabolisme udang. Suhu yang optimal (biasanya antara 28–32°C) akan meningkatkan nafsu makan dan pertumbuhan udang.
- **Sisa Pakan:** Memonitor sisa pakan penting untuk menilai apakah udang mengonsumsi seluruh pakan yang diberikan. Sisa pakan yang tinggi menunjukkan ketidakefisienan dan potensi pencemaran.
- **Waktu Pemberian:** Udang umumnya lebih aktif makan pada pagi dan sore hari. Oleh karena itu, waktu sangat penting dalam menentukan jadwal pemberian pakan yang optimal.

BAB III METODOLOGI

3.1 Alur Pengembangan Sistem

Alur pengembangan sistem dilakukan secara bertahap dan terstruktur. Tahap pertama dimulai dengan pembuatan dataset berbasis data dummy yang merepresentasikan kondisi nyata dalam budidaya udang. Selanjutnya, dilakukan pra-pemrosesan data untuk memastikan kualitas dan konsistensi data yang akan digunakan dalam pelatihan model. Setelah itu, model Decision Tree dibangun dan dilatih menggunakan data tersebut. Kemudian, dilakukan evaluasi terhadap performa model dengan data pengujian. Terakhir, sistem dilengkapi dengan antarmuka pengguna (GUI) untuk memudahkan penggunaan sistem secara praktis.

3.2 Pembuatan Data Set

Data yang digunakan untuk melatih model Decision Tree tidak berasal dari pengukuran sensor langsung, melainkan dibuat secara simulatif dalam bentuk data dummy. Pembuatan data dummy bertujuan untuk merepresentasikan kondisi nyata dalam budidaya udang berdasarkan pengetahuan umum dan asumsi logis terhadap parameter-parameter penting yang memengaruhi kebutuhan pemberian pakan.

Dataset disusun dengan mempertimbangkan lima fitur utama, yaitu waktu pemberian pakan (pagi, siang, sore), suhu air (dalam °C), sisa pakan (dalam kg), umur udang (dalam hari), dan jumlah pakan yang diberikan (dalam kg). Sementara itu, target output atau label dari setiap baris data adalah keputusan apakah pakan perlu diberikan atau tidak, yang dikategorikan dalam nilai "Ya" atau "Tidak".

Data ini kemudian diekspor dalam format CSV atau langsung dibaca oleh Python untuk proses pra-pemrosesan dan pelatihan model. Dengan menggunakan data dummy yang tersusun rapi dan logis, sistem dapat dikembangkan dan diuji meskipun tanpa adanya sensor fisik atau perangkat keras lainnya.

3.3 Pra-pemrosesan Data

Setelah dataset selesai disusun, dilakukan pra-pemrosesan data untuk menyiapkan data yang bersih dan siap digunakan oleh algoritma machine learning. Proses pra-pemrosesan mencakup pengecekan nilai kosong atau duplikat, penghapusan entri tidak valid, serta pengkodean data kategorikal seperti label keputusan dan waktu menjadi nilai numerik. Data kemudian dipisahkan menjadi dua bagian: fitur (input) yang mencakup waktu, suhu air, sisa pakan, umur udang, dan jumlah pakan, serta label (output) berupa keputusan pemberian pakan.

3.4 Pelatihan Model Decision Tree

Model kecerdasan buatan dibangun menggunakan algoritma Decision Tree dengan bantuan pustaka Scikit-learn pada bahasa pemrograman Python. Model ini dilatih menggunakan data fitur dan label yang telah diproses. Tujuannya adalah agar model dapat mengenali pola dari data dan menghasilkan pohon keputusan yang dapat digunakan untuk menentukan apakah udang perlu diberi pakan atau tidak berdasarkan input yang diberikan.

3.5 Evaluasi Model

Setelah model dilatih, dilakukan evaluasi terhadap performa model menggunakan data uji yang belum pernah dilihat sebelumnya oleh model. Evaluasi dilakukan dengan menghitung metrik akurasi untuk mengetahui seberapa tepat model dalam memberikan prediksi. Selain itu, digunakan confusion matrix untuk melihat distribusi prediksi yang benar dan salah. Hasil evaluasi ini menjadi indikator penting dalam menilai keandalan model dalam pengambilan keputusan.

3.6 Pembuatan Antarmuka (GUI)

Untuk meningkatkan kemudahan penggunaan sistem, dibangun antarmuka pengguna berbasis grafis menggunakan library Tkinter di Python. Antarmuka ini dirancang agar pengguna dapat secara manual memasukkan parameter-parameter seperti waktu, suhu air, sisa pakan, dan umur udang. Setelah memasukkan data, pengguna dapat menekan tombol prediksi untuk menjalankan model Decision Tree dan mendapatkan hasil berupa informasi apakah pakan perlu diberikan atau tidak.

Pembuatan GUI dilakukan di lingkungan pengembangan Visual Studio Code (VSCode). Proyek ini dikembangkan menggunakan bahasa pemrograman Python 3.10 dengan bantuan beberapa pustaka seperti Pandas, NumPy, Scikit-learn, dan Tkinter. Seluruh proses pengkodean dilakukan di lingkungan Visual Studio Code (VSCode), yang mendukung integrasi berbagai pustaka serta tampilan interaktif untuk debugging dan visualisasi hasil.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Sistem dikembangkan dalam beberapa komponen utama, dimulai dari pengolahan dataset hingga penyajian antarmuka pengguna. Dataset yang telah dibuat secara simulasi (data dummy) dimuat dan diproses menggunakan pustaka Pandas untuk analisis dan manipulasi data. Selanjutnya, algoritma Decision Tree dari pustaka Scikit-learn digunakan untuk membangun model klasifikasi berdasarkan fitur-fitur seperti waktu, suhu air, sisa pakan, dan umur udang.

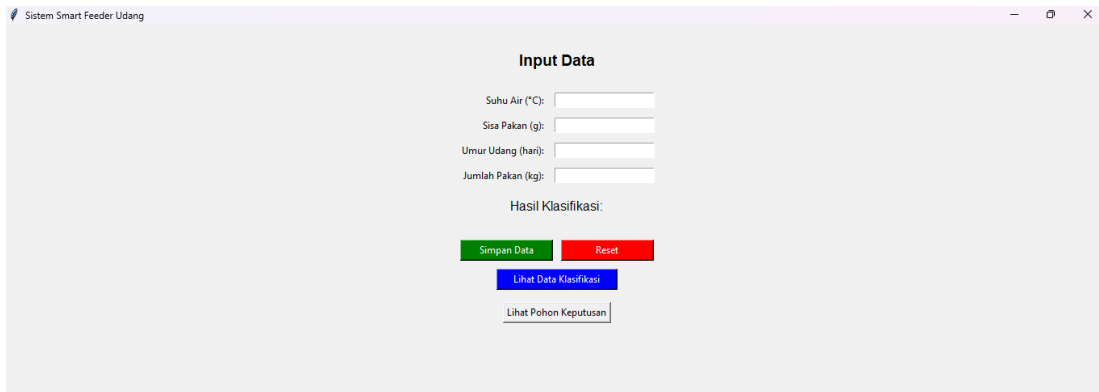
Setelah model selesai dilatih, antarmuka grafis (GUI) dibangun menggunakan pustaka Tkinter. Antarmuka ini memungkinkan pengguna memasukkan nilai-nilai input berupa kondisi tambak secara manual, seperti waktu pemberian pakan, suhu air, sisa pakan, dan umur udang. Setelah input dimasukkan, pengguna dapat menekan tombol “Prediksi” untuk menampilkan keputusan apakah pakan perlu diberikan atau tidak. Lingkungan pengembangan yang digunakan dalam seluruh proses ini adalah Visual Studio Code (VSCode), yang dipilih karena fleksibilitasnya dalam pengelolaan proyek Python dan integrasi pustaka eksternal.

4.2 Pengujian Model

Pengujian model dilakukan untuk mengevaluasi kinerja algoritma Decision Tree dalam memprediksi keputusan pemberian pakan berdasarkan parameter input. Dataset yang digunakan terdiri dari 200 baris data dummy yang merepresentasikan kondisi nyata budidaya udang. Model dilatih menggunakan fitur-fitur penting seperti waktu pemberian pakan, suhu air, sisa pakan, dan umur udang, dengan target output berupa keputusan pemberian pakan ("Ya" atau "Tidak"). Proses pelatihan dilakukan menggunakan pustaka scikit-learn dalam bahasa pemrograman Python. Setelah model dilatih, dilakukan pengujian menggunakan data uji untuk mengukur seberapa akurat model dalam mengenali pola dari data yang belum pernah dilihat sebelumnya.

4.3 Analisis Hasil

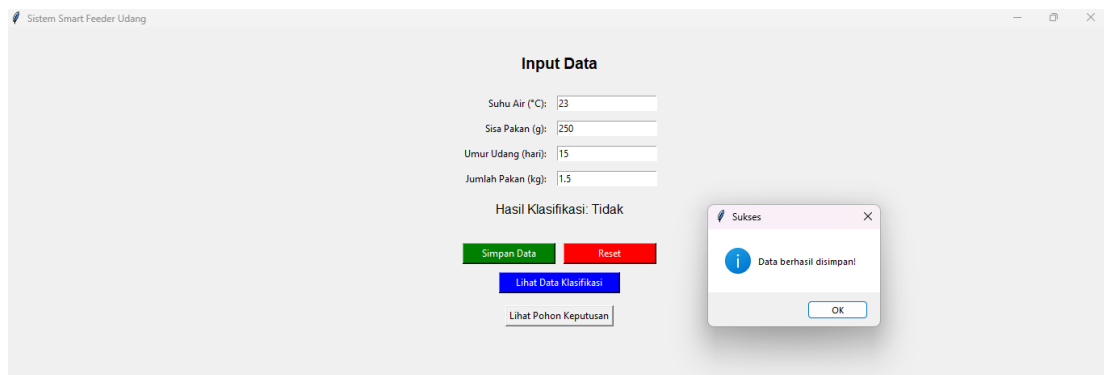
Berikut merupakan tampilan antarmuka dari aplikasi *Sistem Smart Feeder Udang* yang dikembangkan sebagai alat bantu dalam pengambilan keputusan pemberian pakan secara otomatis berdasarkan parameter lingkungan dan kondisi udang.



The screenshot displays the 'Sistem Smart Feeder Udang' application window. The main section is titled 'Input Data' and contains four input fields: 'Suhu Air (°C):', 'Sisa Pakan (g):', 'Umur Udang (hari):', and 'Jumlah Pakan (kg):'. Below these fields is a label 'Hasil Klasifikasi:'. At the bottom of the input section, there are four buttons: 'Simpan Data' (green), 'Reset' (red), 'Lihat Data Klasifikasi' (blue), and 'Lihat Pohon Keputusan' (blue, with a small icon).

Gambar 4.1 Tampilan Antarmuka Sistem Smart Feeder Udang

Antarmuka dari aplikasi Sistem Smart Feeder Udang yang ditampilkan pada Gambar 4.3 menunjukkan desain yang sederhana dan fungsional, dirancang untuk membantu proses pemberian pakan pada udang secara otomatis berdasarkan parameter lingkungan dan kondisi udang. Aplikasi ini memanfaatkan algoritma klasifikasi, seperti Decision Tree, untuk menentukan apakah udang perlu diberi pakan atau tidak. Pengguna dapat memasukkan data berupa suhu air (°C), sisa pakan (g), umur udang (hari), dan jumlah pakan (kg). Keempat parameter tersebut berperan penting dalam menentukan kebutuhan pakan udang, di mana suhu dan umur udang sangat memengaruhi metabolisme dan tingkat konsumsi pakan.



Gambar 4.2 Tampilan Antarmuka Sistem Smart Feeder Udang

Setelah data diinputkan, sistem akan menampilkan hasil klasifikasi pada bagian "Hasil Klasifikasi", yang kemungkinan berisi dua kemungkinan yaitu "Perlu Diberi Pakan" atau "Tidak Perlu Diberi Pakan". Proses klasifikasi ini dijalankan oleh model machine learning yang telah dilatih sebelumnya. Terdapat juga beberapa tombol fungsional pada aplikasi ini, yaitu tombol "Simpan Data" untuk menyimpan input beserta hasil klasifikasinya ke dalam file, tombol "Reset" untuk menghapus semua input, tombol "Lihat Data Klasifikasi" untuk melihat data historis klasifikasi yang telah dilakukan, serta tombol "Lihat Pohon Keputusan" yang menampilkan visualisasi model Decision Tree yang digunakan oleh sistem.

Waktu	Suhu Air (°C)	Sisa Pakan (g)	Umur Udang (hari)	Jumlah Pakan (kg)	Perlu Diberi Pakan
2025-05-04 16:00	30.7	146	3	1.45	Ya
2025-05-05 12:00	29.8	441	4	1.63	Tidak
2025-05-06 20:00	29.3	23	5	1.3	Ya
2025-05-07 08:00	32.0	338	6	1.53	Tidak
2025-05-08 12:00	27.6	101	7	1.19	Ya
2025-05-09 08:00	27.8	378	8	1.25	Tidak
2025-05-10 20:00	32.0	476	9	1.05	Tidak
2025-05-11 16:00	27.1	224	10	1.63	Tidak
2025-05-12 12:00	28.8	93	11	1.15	Ya
2025-05-13 08:00	28.3	105	12	1.81	Ya
2025-05-14 20:00	31.6	95	13	1.89	Ya
2025-05-15 08:00	30.4	219	14	1.81	Tidak
2025-05-16 12:00	28.9	342	15	1.46	Tidak
2025-05-17 16:00	30.3	76	16	1.76	Ya
2025-05-18 08:00	27.6	390	17	1.4	Tidak
2025-05-19 16:00	28.5	102	18	1.74	Ya

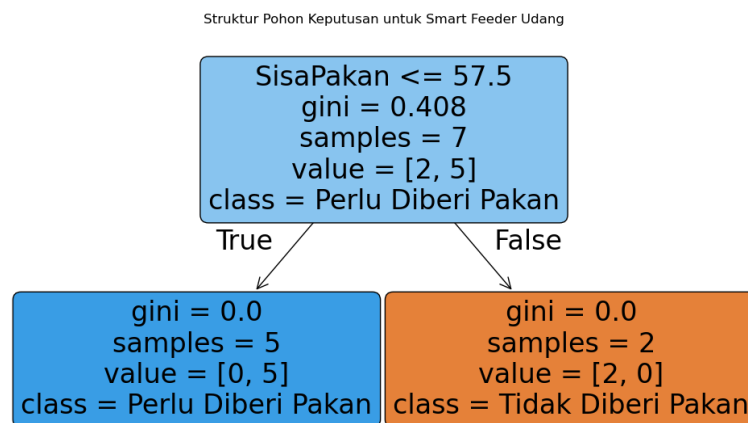
Gambar 4.3 Tampilan Data Klasifikasi

Gambar 4.3 menampilkan tampilan jendela “Data Klasifikasi” yang berisi data historis hasil klasifikasi sistem berdasarkan parameter yang telah dimasukkan oleh

pengguna sebelumnya. Data yang ditampilkan terdiri dari kolom Waktu, Suhu Air (°C), Sisa Pakan (g), Umur Udang (hari), Jumlah Pakan (kg), dan hasil klasifikasi dalam kolom Perlu Diberi Pakan. Setiap baris merepresentasikan satu entri klasifikasi, yang berguna untuk melacak keputusan sistem terhadap kondisi budidaya udang pada waktu tertentu.

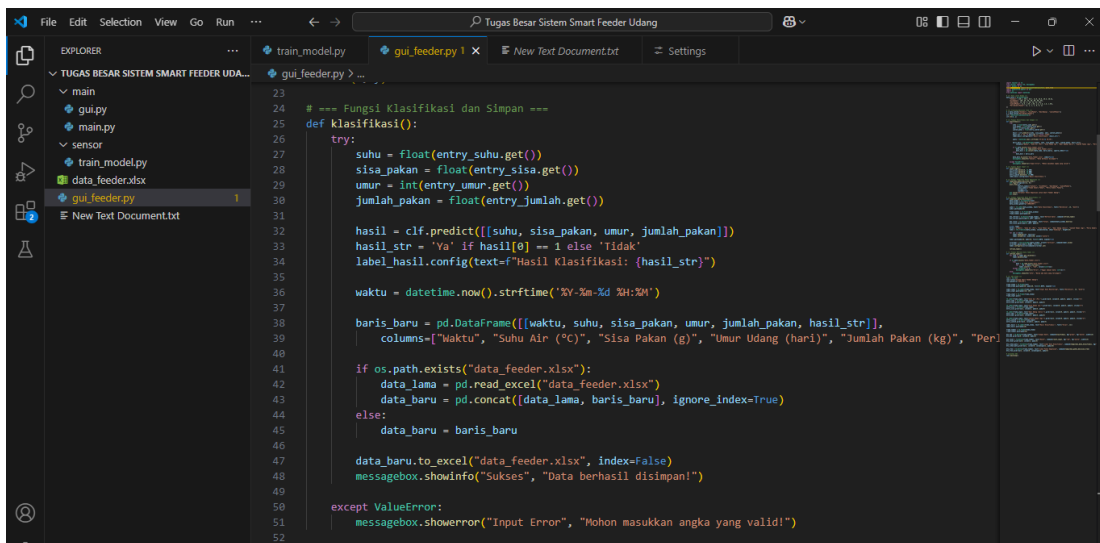
Dari data tersebut terlihat bahwa sistem mampu menyimpan dan mengelola informasi dengan baik serta memberikan keputusan yang konsisten berdasarkan pola dan logika yang telah dipelajari dari model Decision Tree. Sebagai contoh, pada tanggal 2025-05-06 pukul 20:00, dengan suhu air 29,3°C, sisa pakan 23 gram, umur udang 5 hari, dan jumlah pakan 1,3 kg, sistem mengklasifikasikan bahwa udang perlu diberi pakan. Sebaliknya, pada 2025-05-07 pukul 08:00 dengan parameter suhu 32,0°C dan sisa pakan 338 gram, sistem memberikan keputusan “Tidak”. Ini menunjukkan bahwa sistem mempertimbangkan beberapa parameter secara bersamaan untuk menentukan kebutuhan pemberian pakan.

Fitur penyimpanan data klasifikasi ini sangat penting sebagai dasar evaluasi dan pengambilan keputusan jangka panjang dalam manajemen pemberian pakan otomatis pada tambak udang. Data yang tersimpan juga dapat digunakan sebagai bahan pelatihan ulang model agar akurasi semakin meningkat seiring waktu.



Gambar 4.4 Tampilan Struktur Pohon Keputusan

Dari gambar tersebut menunjukkan visualisasi struktur pohon keputusan decision tree yang digunakan dalam sistem Smart Feeder Udang untuk menentukan apakah udang perlu diberi pakan atau tidak, berdasarkan nilai fitur Sisa Pakan. Pohon keputusan dimulai dari node utama yang memeriksa apakah nilai SisaPakan kurang dari atau sama dengan 57,5. Jika nilai tersebut kurang dari atau sama dengan 57,5, maka keputusan yang diambil adalah "Perlu Diberi Pakan" karena mayoritas data pada kondisi ini menunjukkan kebutuhan pakan. Sebaliknya, jika nilai SisaPakan lebih dari 57,5, maka keputusannya adalah "Tidak Diberi Pakan". Setiap node dalam pohon ini juga dilengkapi dengan nilai gini, yang merupakan ukuran ketidakmurnian (impurity) suatu node. Nilai gini = 0 menunjukkan bahwa semua data dalam node tersebut berasal dari satu kelas, artinya node tersebut sepenuhnya murni. Sebaliknya, semakin tinggi nilai gini, semakin bercampur data dari dua kelas di node tersebut. Dalam pohon ini, nilai gini pada node utama adalah 0,408 yang menunjukkan bahwa data dalam node tersebut masih mengandung campuran dua kelas, sedangkan kedua node anak memiliki nilai gini 0, yang berarti data pada masing-masing node anak sepenuhnya homogen. Struktur sederhana ini menunjukkan bahwa fitur SisaPakan sangat menentukan dalam pengambilan keputusan pemberian pakan.



```

23
24 # --- Fungsi Klasifikasi dan Simpan ---
25 def klasifikasi():
26     try:
27         suhu = float(entry_suhu.get())
28         sisa_pakan = float(entry_sisa.get())
29         umur = int(entry_umur.get())
30         jumlah_pakan = float(entry_jumlah.get())
31
32         hasil = clf.predict([[suhu, sisa_pakan, umur, jumlah_pakan]])
33         hasil_str = 'Ya' if hasil[0] == 1 else 'Tidak'
34         label_hasil.config(text=f"Hasil Klasifikasi: {hasil_str}")
35
36         waktu = datetime.now().strftime('%Y-%m-%d %H:%M')
37
38         baris_baru = pd.DataFrame([waktu, suhu, sisa_pakan, umur, jumlah_pakan, hasil_str],
39                                   columns=["waktu", "Suhu Air (°C)", "Sisa Pakan (g)", "Umur Udang (hari)", "Jumlah Pakan (kg)", "Per"])
40
41         if os.path.exists("data_feeder.xlsx"):
42             data_lama = pd.read_excel("data_feeder.xlsx")
43             data_baru = pd.concat([data_lama, baris_baru], ignore_index=True)
44         else:
45             data_baru = baris_baru
46
47         data_baru.to_excel("data_feeder.xlsx", index=False)
48         messagebox.showinfo("Sukses", "Data berhasil disimpan!")
49
50     except ValueError:
51         messagebox.showerror("Input Error", "Mohon masukkan angka yang valid!")
52

```

Gambar 4.5 Tampilan Koding

Kode ini berfungsi untuk mengambil input dari pengguna melalui antarmuka aplikasi, yaitu nilai suhu air, sisa pakan, umur udang, dan jumlah pakan, yang kemudian dikonversi ke tipe data numerik sesuai kebutuhan. Setelah data berhasil diambil dan dikonversi, model machine learning yang telah dilatih (misalnya Decision Tree) melakukan prediksi untuk menentukan apakah udang perlu diberi pakan atau tidak. Hasil prediksi ini kemudian ditampilkan secara langsung pada antarmuka dalam bentuk teks yang mudah dipahami, yaitu "Ya" atau "Tidak". Selanjutnya, data input beserta hasil prediksi dan waktu pengambilan keputusan disimpan dalam sebuah baris data yang diolah menggunakan pandas DataFrame. Kode memeriksa apakah file Excel penyimpanan data sudah ada; jika ada, data baru digabungkan dengan data lama, jika belum, data baru langsung disimpan sebagai file baru. Proses penyimpanan ini memastikan bahwa seluruh riwayat klasifikasi dapat terdokumentasi dengan baik untuk evaluasi lebih lanjut. Seluruh proses ini dibungkus dalam blok try-except untuk menangani kesalahan input, sehingga apabila pengguna memasukkan data yang tidak valid, aplikasi akan menampilkan pesan error yang memberitahukan agar memasukkan angka yang benar. Dengan demikian, kode ini mengintegrasikan pengambilan data, prediksi otomatis, tampilan hasil, dan penyimpanan data secara aman dan efisien dalam aplikasi Smart Feeder Udang.

4.4 Pembahasan

Perkembangan teknologi di bidang akuakultur mendorong diterapkannya otomatisasi untuk meningkatkan efisiensi budidaya, salah satunya melalui pengembangan Sistem Smart Feeder Udang. Sistem ini menggunakan algoritma Decision Tree untuk memprediksi kebutuhan pemberian pakan berdasarkan parameter lingkungan seperti suhu air, umur udang, sisa pakan, dan jumlah pakan, sehingga mampu mengurangi pemborosan dan meningkatkan pertumbuhan udang secara optimal. Data input dimasukkan melalui antarmuka GUI berbasis Python dan Tkinter, yang dirancang sederhana dan fungsional untuk memudahkan pengguna dalam melakukan prediksi, menyimpan hasil ke file Excel, serta melihat visualisasi pohon keputusan dan riwayat klasifikasi. Sistem secara real-time mengklasifikasikan kebutuhan pakan udang dengan hasil yang jelas, dimana sisa pakan menjadi faktor

utama dalam pengambilan keputusan sesuai dengan visualisasi pohon keputusan yang menunjukkan tingkat kemurnian data di setiap cabang pohon. Pengujian menggunakan data dummy sebanyak 200 baris menunjukkan bahwa model mampu mengklasifikasikan kebutuhan pakan secara akurat, sehingga Sistem Smart Feeder Udang menjadi solusi efektif dalam mengotomatisasi proses pemberian pakan serta mendukung pengambilan keputusan manajemen tambak secara efisien dan adaptif.

BAB V

KESIMPULAN

5.1 Kesimpulan

Sistem Smart Feeder Udang yang dikembangkan berhasil menyediakan solusi otomatis untuk pengambilan keputusan pemberian pakan berdasarkan parameter lingkungan dan kondisi udang, seperti suhu air, sisa pakan, umur udang, dan jumlah pakan. Dengan memanfaatkan algoritma Decision Tree, sistem mampu melakukan klasifikasi secara akurat dalam menentukan apakah udang perlu diberi pakan atau tidak, sehingga dapat mengurangi pemborosan pakan dan mendukung pertumbuhan udang yang optimal. Antarmuka berbasis Python dan Tkinter memudahkan pengguna dalam memasukkan data, melihat hasil prediksi secara real-time, menyimpan data klasifikasi ke file Excel, serta mengakses riwayat data dan visualisasi pohon keputusan. Pengujian dengan data dummy menunjukkan bahwa model bekerja dengan baik dan sistem dapat beroperasi secara stabil serta konsisten. Dengan demikian, sistem ini efektif sebagai alat bantu dalam manajemen pemberian pakan otomatis di budidaya udang.

5.2 Saran

Untuk pengembangan lebih lanjut, beberapa aspek dapat ditingkatkan agar sistem lebih optimal dan fungsional, antara lain:

1. Mengintegrasikan sensor otomatis untuk pengambilan data lingkungan secara real-time, sehingga input data tidak bergantung pada masukan manual dari pengguna.
2. Mengembangkan model machine learning yang lebih kompleks atau menggunakan teknik ensemble untuk meningkatkan akurasi prediksi.
3. Menambahkan fitur notifikasi atau alarm otomatis untuk memberitahukan kapan pakan perlu diberikan secara langsung ke perangkat pengguna, seperti smartphone.

LAMPIRAN

```
File Edit Selection View Go Run ... Tugas Besar Sistem Smart Feeder Udang
gui_feeder.py x New Text Document.txt Settings

gui_feeder.py > tampilkan_data_klasifikasi
1 import tkinter as tk
2 from tkinter import ttk, messagebox
3 import pandas as pd
4 from sklearn.tree import DecisionTreeClassifier, plot_tree
5 import matplotlib.pyplot as plt
6 import os
7 from datetime import datetime
8
9 # === Data Latih Dummy ===
10 data_latih = pd.DataFrame({
11     "SuhuAir": [30.1, 30.5, 30.7, 31.0, 28.9, 29.5, 30.0],
12     "SisaPakan": [50, 45, 46, 20, 80, 60, 55],
13     "UmurUdang": [3, 10, 3, 25, 15, 15, 10],
14     "JumlahPakan": [1.5, 1.45, 1.46, 1.2, 1.8, 1.6, 1.55],
15     "PerluDiberiPakan": [1, 1, 1, 1, 0, 0, 1]
16 })
17
18 # === Training Decision Tree ===
19 X = data_latih[["SuhuAir", "SisaPakan", "UmurUdang", "JumlahPakan"]]
20 y = data_latih["PerluDiberiPakan"]
21 clf = DecisionTreeClassifier()
22 clf.fit(X, y)
23
24 # === Fungsi Klasifikasi dan Simpan ===
25 def klasifikasi():
26     try:
27         suhu = float(entry_suhu.get())
28         sisa_pakan = float(entry_sisa.get())
29         umur = int(entry_umur.get())
30         jumlah_pakan = float(entry_jumlah.get())
31
32         hasil = clf.predict([[suhu, sisa_pakan, umur, jumlah_pakan]])
```

```
File Edit Selection View Go Run ... Tugas Besar Sistem Smart Feeder Udang
gui_feeder.py x New Text Document.txt Settings

gui_feeder.py > tampilkan_data_klasifikasi
25 def klasifikasi():
31     # ... (previous code) ...
32     hasil = clf.predict([[suhu, sisa_pakan, umur, jumlah_pakan]])
33     hasil_str = 'Ya' if hasil[0] == 1 else 'Tidak'
34     label_hasil.config(text=f"Hasil Klasifikasi: {hasil_str}")
35
36     waktu = datetime.now().strftime('%Y-%m-%d %H:%M')
37
38     baris_baru = pd.DataFrame([waktu, suhu, sisa_pakan, umur, jumlah_pakan, hasil_str],
39                               columns=["Waktu", "Suhu Air (°C)", "Sisa Pakan (g)", "Umur Udang (hari)", "Jumlah Pakan (kg)", "Perlu Diberi Pakan"])
40
41     if os.path.exists("data_feeder.xlsx"):
42         data_lama = pd.read_excel("data_feeder.xlsx")
43         data_baru = pd.concat([data_lama, baris_baru], ignore_index=True)
44     else:
45         data_baru = baris_baru
46
47     data_baru.to_excel("data_feeder.xlsx", index=False)
48     messagebox.showinfo("Sukses", "Data berhasil disimpan!")
49
50 except ValueError:
51     messagebox.showerror("Input Error", "Mohon masukkan angka yang valid!")
52
53 # === Fungsi Reset Input ===
54 def reset_input():
55     entry_suhu.delete(0, tk.END)
56     entry_sisa.delete(0, tk.END)
57     entry_umur.delete(0, tk.END)
58     entry_jumlah.delete(0, tk.END)
59     label_hasil.config(text="Hasil Klasifikasi:")
60
61 # ... (previous code) ...
```

```
File Edit Selection View Go Run ... Tugas Besar Sistem Smart Feeder Udang
gui_feeder.py 1 X New Text Document.txt Settings
gui_feeder.py > tampilkan_data_klasifikasi
60
61 # === Fungsi Tampilkan Pohon Keputusan ===
62 def tampilkan_pohon_decision_tree():
63     plt.figure(figsize=(12, 8))
64     plot_tree(clf,
65               feature_names=["SuhuAir", "SisaPakan", "UmurUdang", "JumlahPakan"],
66               class_names=["Tidak Diberi Pakan", "Perlu Diberi Pakan"],
67               filled=True,
68               rounded=True)
69     plt.title("Struktur Pohon Keputusan untuk Smart Feeder Udang")
70     plt.show()
71
72 # === Fungsi Tampilkan Data Klasifikasi ===
73 def tampilkan_data_klasifikasi():
74     data_window = tk.Toplevel(root)
75     data_window.title("Data Klasifikasi")
76     data_window.geometry("1000x500")
77
78     judul = tk.Label(data_window, text="Data Klasifikasi", font=("Helvetica", 14, "bold"))
79     judul.pack(pady=10)
80
81     frame_tombol = tk.Frame(data_window)
82     frame_tombol.pack(pady=10)
83
84     btn_refresh = tk.Button(frame_tombol, text="Refresh Data", command=refresh_tabel)
85     btn_refresh.pack(side=tk.LEFT, padx=5)
86
87     btn_close = tk.Button(frame_tombol, text="Tutup", command=data_window.destroy)
88     btn_close.pack(side=tk.LEFT, padx=5)
89
90     global tabel
91     kolom = ["Waktu", "Suhu Air (°C)", "Sisa Pakan (g)", "Umur Udang (hari)", "Jumlah Pakan (kg)", "Perlu Diberi Pakan"]
```

```
File Edit Selection View Go Run ... Tugas Besar Sistem Smart Feeder Udang
gui_feeder.py 1 X New Text Document.txt Settings
gui_feeder.py > tampilkan_data_klasifikasi
73 def tampilkan_data_klasifikasi():
74     kolom = ["Waktu", "Suhu Air (°C)", "Sisa Pakan (g)", "Umur Udang (hari)", "Jumlah Pakan (kg)", "Perlu Diberi Pakan"]
75     tabel = ttk.Treeview(data_window, columns=kolom, show="headings", height=15)
76
77     for col in kolom:
78         tabel.heading(col, text=col)
79         tabel.column(col, width=140, anchor="center")
80
81     tabel.pack(pady=10, padx=10, fill=tk.BOTH, expand=True)
82
83     scrollbar = ttk.Scrollbar(data_window, orient="vertical", command=tabel.yview)
84     scrollbar.pack(side=tk.RIGHT, fill=tk.Y)
85     tabel.configure(yscrollcommand=scrollbar.set)
86
87     refresh_tabel()
88
89 # === Fungsi Refresh Data Tabel ===
90 def refresh_tabel():
91     for item in tabel.get_children():
92         tabel.delete(item)
93
94     if os.path.exists("data_feeder.xlsx"):
95         try:
96             data = pd.read_excel("data_feeder.xlsx")
97             for _, row in data.iterrows():
98                 tabel.insert("", "end", values=list(row))
99         except Exception as e:
100             messagebox.showerror("Error", f"Gagal memuat data: {str(e)}")
101     else:
102         messagebox.showinfo("Info", "Belum ada data yang tersimpan")
103
104 # === GUI Utama ===
```

```
File Edit Selection View Go Run ... Tugas Besar Sistem Smart Feeder Udang
gui_feeder.py x New Text Document.txt Settings
gui_feeder.py > tampilkan_data_klasifikasi
121 # == GUI Utama ==
122 root = tk.Tk()
123 root.title("Sistem Smart Feeder Udang")
124 root.geometry("500x420")
125
126 frame_utama = tk.Frame(root)
127 frame_utama.pack(pady=20, padx=20, fill=tk.BOTH, expand=True)
128
129 judul_input = tk.Label(frame_utama, text="Input Data Monitoring", font=("Helvetica", 14, "bold"))
130 judul_input.pack(pady=(10, 20))
131
132 frame_input = tk.Frame(frame_utama)
133 frame_input.pack()
134
135 tk.Label(frame_input, text="Suhu Air (°C):").grid(row=0, column=0, padx=5, pady=5, sticky="e")
136 entry_suhu = tk.Entry(frame_input)
137 entry_suhu.grid(row=0, column=1, padx=5, pady=5)
138
139 tk.Label(frame_input, text="Sisa Pakan (g):").grid(row=1, column=0, padx=5, pady=5, sticky="e")
140 entry_sisa = tk.Entry(frame_input)
141 entry_sisa.grid(row=1, column=1, padx=5, pady=5)
142
143 tk.Label(frame_input, text="Umur Udang (hari):").grid(row=2, column=0, padx=5, pady=5, sticky="e")
144 entry_umur = tk.Entry(frame_input)
145 entry_umur.grid(row=2, column=1, padx=5, pady=5)
146
147 tk.Label(frame_input, text="Jumlah Pakan (kg):").grid(row=3, column=0, padx=5, pady=5, sticky="e")
148 entry_jumlah = tk.Entry(frame_input)
149 entry_jumlah.grid(row=3, column=1, padx=5, pady=5)
150
151 label_hasil = tk.Label(frame_utama, text="Hasil Klasifikasi:", font=("Arial", 12))
152 label_hasil.pack(pady=10)
```

```
File Edit Selection View Go Run ... Tugas Besar Sistem Smart Feeder Udang
gui_feeder.py x New Text Document.txt Settings
gui_feeder.py > tampilkan_data_klasifikasi
153
154 frame_tombol = tk.Frame(frame_utama)
155 frame_tombol.pack(pady=20)
156
157 btn_oke = tk.Button(frame_tombol, text="Simpan Data", command=klasifikasi, bg="green", fg="white", width=15)
158 btn_oke.grid(row=0, column=0, padx=5)
159
160 btn_reset = tk.Button(frame_tombol, text="Reset", command=reset_input, bg="red", fg="white", width=15)
161 btn_reset.grid(row=0, column=1, padx=5)
162
163 btn_lihat_data = tk.Button(frame_tombol, text="Lihat Data Klasifikasi", command=tampilkan_data_klasifikasi, bg="blue", fg="white", width=20)
164 btn_lihat_data.grid(row=1, column=0, colspan=2, pady=10)
165
166 btn_tree = tk.Button(frame_tombol, text="Lihat Pohon Keputusan", command=tampilkan_pohon_decision_tree)
167 btn_tree.grid(row=2, column=0, colspan=2, pady=5)
168
169 # Jalankan GUI
170 root.mainloop()
171
```