TUGAS AKHIR SEMESTER

MATA KULIAH KECERDASAN BUATAN

SISTEM DIAGNOSA PENYAKIT PADA ANAK

DOSEN PENGAJAR :

Iftitaahul Mufarrihah, S.Si.M.Pd



DI SUSUN OLEH :

1. Muhammad Abdul Jabbar (2395114039)
2. Muhammad Hasda Putra D (2395114003)

KELAS TI A

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS HASYM ASY’ARI TEBUIRENG JOMBANG

TAHUN AJARAN 2024/2025

**1. Pendahuluan**

**1.1 Latar Belakang Masalah**

Ikan lele merupakan salah satu komoditas perikanan air tawar yang banyak dibudidayakan di Indonesia. Permintaan pasar yang tinggi menjadikan budidaya ikan lele sebagai usaha yang menjanjikan. Namun, keberhasilan budidaya ikan lele sangat bergantung pada kondisi kesehatan ikan. Penyakit yang menyerang ikan lele dapat menyebabkan penurunan produktivitas dan kualitas, bahkan bias mengakibatkan kematian massal, yang pada akhirnya merugikan para pembudidaya.

Pendeteksian penyakit pada ikan lele umumnya masih dilakukan secara manual berdasarkan pengalaman atau pengetahuan terbatas para peternak. Hal ini dapat menimbulkan keterlambatan dalam penanganan dan kesalahan dalam diagnosis. Untuk itu, diperlukan suatu sistem yang mampu membantu mendiagnosa penyakit secara cepat dan akurat.

Dengan memanfaatkan teknologi kecerdasan buatan, khususnya sistem pakar berbasis aturan (rule-based expert system), dapat dibangun sebuah sistem berbasis web yang mampu memberikan kemungkinan diagnosa penyakit ikan lele berdasarkan gejala-gejala yang diinputkan oleh pengguna. Sistem ini diharapkan dapat menjadi alat bantu yang efektif dalam proses identifikasi awal penyakit dan memberikan rekomendasi penanganan yang sesuai.

**1.2 Rumusan Masalah**

* Bagaimana merancang sistem pakar berbasis web untuk mendiagnosa penyakit ikan lele?
* Bagaimana sistem dapat memberikan hasil diagnosa yang sesuai dengan gejala yang diinputkan oleh pengguna?

**1.3 Tujuan**

* Membangun sistem pakar berbasis web yang mampu mendiagnosa penyakit pada anak berdasarkan gejala-gejala yang dialami.
* Memberikan rekomendasi penanganan awal terhadap penyakit yang teridentifikasi.

**1.4 Manfaat**

* Membantu ibu-ibu dalam mendeteksi penyakit secara dini dan mengambil tindakan yang tepat.
* Menyediakan antarmuka web yang sederhana dan mudah digunakan sebagai alat bantu diagnosa.
* Menjadi referensi untuk pengembangan sistem pakar lainnya di bidang akuakultur atau peternakan.

**2. Tinjauan Pustaka**

**2.1 Penjelasan Metode AI yang Digunakan: Rule-Based System**

Sistem pakar merupakan salah satu cabang dari kecerdasan buatan (Artificial Intelligence/AI) yang dirancang untuk meniru cara berpikir dan pengambilan keputusan seorang pakar manusia dalam bidang tertentu. Dalam proyek ini, metode yang digunakan adalah **Rule-Based System**, yaitu sistem yang bekerja berdasarkan sekumpulan aturan logika (IF-THEN) yang telah ditentukan sebelumnya.

Sistem rule-based bekerja dengan cara mencocokkan **fakta** (dalam hal ini adalah input gejala dari pengguna seperti warna tubuh ikan, nafsu makan, perilaku, dan kondisi fisik lainnya) dengan **aturan-aturan** yang telah dirancang oleh pakar atau berdasarkan literatur. Jika kondisi dalam aturan terpenuhi, maka sistem akan menghasilkan **kesimpulan** berupa diagnosis penyakit beserta rekomendasi penanganannya.

Contoh aturan dalam sistem ini:

IF Gejala1 = Demam AND Gejala2 = Batuk AND Gejala3 = Pilek

THEN Penyakit = Influenza (Flu),

Penanganan = Istirahat yang cukup, minum banyak air, konsumsi obat pereda gejala (jika perlu), hindari kontak dengan orang lain.

Metode ini sangat cocok digunakan untuk kasus **diagnosa penyakit ikan**, karena jenis penyakit yang umum terjadi biasanya memiliki pola gejala yang spesifik dan tidak terlalu kompleks, sehingga dapat direpresentasikan dalam bentuk aturan-aturan yang terbatas namun jelas.

**2.2 Studi Terdahulu atau Contoh Aplikasi Serupa**

Beberapa penelitian dan aplikasi serupa telah dikembangkan sebelumnya dalam konteks sistem pakar berbasis web, di antaranya:

* **Sistem Pakar Diagnosa Penyakit Tanaman Padi**

Sistem ini menggunakan pendekatan rule-based untuk mendiagnosis penyakit tanaman berdasarkan gejala seperti bercak daun, warna batang, dan bentuk akar. Tujuannya adalah membantu petani dalam mengenali penyakit secara mandiri dengan memberikan saran penanganan awal sebelum berkonsultasi lebih lanjut dengan penyuluh pertanian.

* **Sistem Pakar Diagnosa Penyakit Kucing Berbasis Web**

Sistem ini dirancang untuk pemilik hewan peliharaan, khususnya kucing, dengan metode diagnosis berbasis gejala seperti nafsu makan menurun, muntah, atau perubahan perilaku. Sistem mencocokkan gejala dengan basis aturan dan memberikan kemungkinan penyakit serta saran penanganan dasar. Pendekatan ini sangat berguna dalam konteks konsultasi awal sebelum ke dokter hewan.

* **Sistem Pakar Diagnosa Penyakit Ikan Lele Berbasis Web**

Sistem ini berfokus pada dunia perikanan dan mengidentifikasi penyakit pada ikan lele berdasarkan perubahan fisik dan perilaku. Dengan input dari pakar perikanan, sistem memberikan hasil diagnosis yang disesuaikan dengan kondisi umum kolam budidaya di Indonesia.

**Keunikan sistem yang dikembangkan dalam proyek ini** terletak pada fokusnya pada **penyakit manusia** dengan pendekatan yang sederhana namun efektif. Sistem ini dibangun berbasis web menggunakan **Python Flask** dan memanfaatkan **rule-based system** untuk mencocokkan input gejala pengguna dengan data penyakit dalam format CSV.  
Antarmuka yang mudah digunakan memungkinkan siapa saja—baik masyarakat umum maupun tenaga kesehatan pemula—untuk memperoleh gambaran awal terhadap kemungkinan penyakit dan langkah penanganan umum yang dapat dilakukan sebelum memutuskan berkonsultasi ke fasilitas kesehatan.

Dengan demikian, sistem ini diharapkan dapat menjadi **alat bantu awal yang praktis dan informatif** dalam mendeteksi berbagai penyakit ringan hingga menengah berdasarkan gejala yang umum terjadi.

**3. Metodologi**

**3.1 Desain Sistem**

Sistem pakar diagnosa penyakit ikan lele ini dirancang untuk membantu pengguna (khususnya peternak) mengenali penyakit yang menyerang ikan lele berdasarkan gejala yang diamati. Sistem ini berbasis web dan dibangun menggunakan pendekatan **rule-based**, di mana sistem akan mencocokkan input gejala dari pengguna dengan sekumpulan aturan penyakit yang telah didefinisikan sebelumnya.

Secara umum, sistem terdiri dari tiga komponen utama:

1. **Antarmuka Pengguna (Frontend)** – Formulir input gejala yang dirancang agar pengguna dapat memilih kondisi fisik dan perilaku ikan seperti warna tubuh, luka, nafsu makan, posisi berenang, dll.
2. **Proses Pemrosesan Data (Backend)** – Kode program Python (menggunakan Flask) yang membaca input gejala, mencocokkannya dengan aturan yang tersedia, dan menghasilkan diagnosa serta saran penanganan.
3. **Basis Data (CSV atau Dictionary)** – Menyimpan aturan-aturan diagnosis yang terdiri dari kombinasi gejala dan hasil diagnosa yang sesuai.

Diagram alur sistem digambarkan sebagai berikut:

Pengguna

│

▼

Form Input Gejala (3 dropdown)

│

▼

Pemrosesan Rule-Based

└─> Bandingkan input dengan data di CSV

└─> Jika cocok, simpan hasil diagnosis dan saran

│

▼

Hasil Diagnosa + Saran Penanganan

**3.2 Langkah-langkah Implementasi**

Berikut adalah tahapan implementasi sistem:

1. **Pengumpulan dan Penyusunan Data**

* Data dikumpulkan berdasarkan literatur atau pengalaman praktisi perikanan.
* Setiap entri berisi kombinasi gejala dan kemungkinan penyakit serta rekomendasi penanganan.
* Contoh data disimpan dalam format dictionary atau CSV dengan kolom:

Gejala1, Gejala2, Gejala3, Penyakit, Penanganan.

1. **Pembuatan Antarmuka**

* Form HTML dirancang agar pengguna dapat memilih atau mencentang gejala yang dialami ikan lele.
* Input dikirim ke backend Flask untuk diproses.

1. **Implementasi Logika Rule-Based**

* Backend Flask menerima data dari formulir dan mencocokkannya dengan aturan yang sudah dibuat.
* Jika ditemukan kecocokan gejala, maka sistem akan menampilkan kemungkinan penyakit dan saran penanganan.

for rule in daftar\_aturan:

if (

rule['Gejala1'] in input\_gejala and

rule['Gejala2'] in input\_gejala and

rule['Gejala3'] in input\_gejala

):

hasil.append({

'penyakit': rule['KemungkinanPenyakit'],

'penanganan': rule['PenangananUmum']

})

**Tampilan Hasil Diagnosa**

Sistem akan menampilkan kemungkinan penyakit berdasarkan gejala, beserta saran penanganan awal seperti penggunaan obat, atau karantina..

**3.3 Alat dan Bahasa Pemrograman yang Digunakan**

1. **Bahasa Pemrograman:**

* Python (Flask framework) – untuk logika pemrosesan dan backend web
* HTML, CSS – untuk pembuatan antarmuka pengguna (frontend)

1. **Tools dan Library:**

* Flask – Framework Python ringan untuk web
* CSV / Dictionary – Untuk menyimpan data aturan diagnose
* Jinja2 – Untuk menampilkan data dinamis di HTML

1. **Editor dan Lingkungan Pengembangan:**

* Visual Studio Code – Editor kode
* Browser (Chrome/Firefox) – Untuk menguji antarmuka web
* Sistem operasi: Windows 11

**4. Implementasi**

**4.1 Penjelasan Kode Program Secara Umum**

Aplikasi ini dibangun menggunakan bahasa pemrograman Python dengan framework Flask sebagai fondasi pengembangan aplikasi web. Tujuan utama sistem adalah memberikan rekomendasi produk makanan kucing berdasarkan input pengguna menggunakan pendekatan rule-based.

File utama:

1. **app.py** – Mengatur proses input, rule-checking, dan pengembalian hasil ke frontend HTML.

Berikut potongan kode utama dari app.py:

**from flask import Flask, render\_template, request**

**import csv**

**import os**

**app = Flask(\_\_name\_\_)**

**# --------------------------------------------------------------------------- #**

**# Helpers**

**# --------------------------------------------------------------------------- #**

**BASE\_DIR = os.path.dirname(os.path.abspath(\_\_file\_\_))**

**def load\_rules(filepath=None):**

**"""**

**Load the CSV and return every row as a dict.**

**"""**

**if filepath is None:**

**filepath = os.path.join(BASE\_DIR, "data\_penyakit.csv")**

**with open(filepath, newline="", encoding="utf-8") as csvfile:**

**return list(csv.DictReader(csvfile))**

**# --------------------------------------------------------------------------- #**

**# Routes**

**# --------------------------------------------------------------------------- #**

**@app.route("/", methods=["GET", "POST"])**

**def index():**

**# These are the form fields shown to the user**

**gejala\_list = [**

**"warna\_tubuh",**

**"nafsu\_makan",**

**"luka",**

**"perilaku",**

**"posisi\_berenang",**

**]**

**input\_gejala = []**

**hasil = []**

**if request.method == "POST":**

**# Grab whatever the user selected/typed**

**input\_gejala = [request.form.get(g, "").strip() for g in gejala\_list]**

**# Compare with CSV rules**

**for rule in load\_rules():**

**# In the CSV we only store 3 gejala columns (Gejala1–3)**

**if all(rule[f"Gejala{i}"] in input\_gejala for i in range(1, 4)):**

**hasil.append(**

**{**

**"penyakit": rule["Penyakit"],**

**"penanganan": rule["Penanganan"],**

**}**

**)**

**return render\_template(**

**"index.html", gejala\_list=gejala\_list, hasil=hasil, gejala\_terpilih=input\_gejala**

**)**

**# --------------------------------------------------------------------------- #**

**# Main entry point**

**# --------------------------------------------------------------------------- #**

**if \_\_name\_\_ == "\_\_main\_\_":**

**# Browse to http://127.0.0.1:5001/**

**app.run(debug=True, port=5001)**

Berikut kode lengkap dari data\_penyakit.csv:

**Gejala1,Gejala2,Gejala3,Penyakit,Penanganan**

**Pucat,Menurun,Ada\_luka,Aeromonas hydrophila,Perbaiki kualitas air dan berikan antibiotik**

**Merah,Normal,Menggosok tubuh,Trichodina,Perendaman dengan larutan formalin 25 ppm**

**Pucat,Tidak Makan,Ada\_luka,Infeksi Bakteri,Pindahkan ikan ke kolam karantina dan berikan antibiotik sesuai dosis.**

**Merah,Menurun,Tidak\_ada\_luka,Parasit Kulit,Berikan larutan formalin dengan konsentrasi rendah sesuai prosedur.**

**Normal,Normal,Tidak\_ada\_luka,Sehat,Tidak diperlukan penanganan.**

Berikut kode lengkap dari index.html:

**<!DOCTYPE html>**

**<html lang="id">**

**<head>**

**<meta charset="UTF-8">**

**<title>Sistem Pakar Diagnosis Penyakit Ikan Lele</title>**

**<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">**

**<style>**

**body {**

**background-color: #f8f9fa;**

**}**

**.container {**

**margin-top: 40px;**

**}**

**.card {**

**border-radius: 1rem;**

**box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);**

**}**

**.btn-primary {**

**width: 100%;**

**}**

**</style>**

**</head>**

**<body>**

**<div class="container">**

**<div class="card p-4">**

**<h2 class="mb-4 text-center">Sistem Pakar Diagnosis Penyakit Ikan Lele</h2>**

**<form method="POST">**

**<div class="mb-3">**

**<label for="warna\_tubuh" class="form-label">Warna Tubuh</label>**

**<select name="warna\_tubuh" class="form-select" required>**

**<option value="">Pilih...</option>**

**<option value="Pucat">Pucat</option>**

**<option value="Merah">Merah</option>**

**<option value="Normal">Normal</option>**

**</select>**

**</div>**

**<div class="mb-3">**

**<label for="nafsu\_makan" class="form-label">Nafsu Makan</label>**

**<select name="nafsu\_makan" class="form-select" required>**

**<option value="">Pilih...</option>**

**<option value="Menurun">Menurun</option>**

**<option value="Normal">Normal</option>**

**<option value="Tidak Makan">Tidak Makan</option>**

**</select>**

**</div>**

**<div class="mb-3">**

**<label for="luka" class="form-label">Kondisi Luka</label>**

**<select name="luka" class="form-select" required>**

**<option value="">Pilih...</option>**

**<option value="Ada\_luka">Ada Luka</option>**

**<option value="Tidak\_ada\_luka">Tidak Ada Luka</option>**

**</select>**

**</div>**

**<div class="mb-3">**

**<label for="perilaku" class="form-label">Perilaku</label>**

**<select name="perilaku" class="form-select" required>**

**<option value="">Pilih...</option>**

**<option value="Menggosok tubuh">Menggosok tubuh</option>**

**<option value="Lemah">Lemah</option>**

**<option value="Normal">Normal</option>**

**</select>**

**</div>**

**<div class="mb-3">**

**<label for="posisi\_berenang" class="form-label">Posisi Berenang</label>**

**<select name="posisi\_berenang" class="form-select" required>**

**<option value="">Pilih...</option>**

**<option value="Tengkurap">Tengkurap</option>**

**<option value="Miring">Miring</option>**

**<option value="Normal">Normal</option>**

**</select>**

**</div>**

**<button type="submit" class="btn btn-primary mt-3">Diagnosa</button>**

**</form>**

**{% if hasil %}**

**<div class="mt-4">**

**<h5>Hasil Diagnosis:</h5>**

**<ul class="list-group">**

**{% for h in hasil %}**

**<li class="list-group-item">**

**<strong>Penyakit:</strong> {{ h.penyakit }}<br>**

**<strong>Penanganan:</strong> {{ h.penanganan }}**

**</li>**

**{% endfor %}**

**</ul>**

**</div>**

**{% elif gejala\_terpilih %}**

**<div class="alert alert-warning mt-4" role="alert">**

**Tidak ditemukan penyakit yang cocok dengan gejala yang dipilih.**

**</div>**

**{% endif %}**

**</div>**

**</div>**

**</body>**

**</html>**

**Untuk Index.html**

**index.html**

**<!DOCTYPE html>**

**<html lang="id">**

**<head>**

**<meta charset="UTF-8">**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

**<title>Sistem Pakar Diagnosis Penyakit Manusia</title>**

**<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">**

**<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600&display=swap" rel="stylesheet">**

**<style>**

**body {**

**background-color: #f4f7f6;**

**font-family: 'Poppins', sans-serif;**

**color: #333;**

**}**

**.container {**

**margin-top: 40px;**

**margin-bottom: 40px;**

**max-width: 800px;**

**}**

**.card {**

**border-radius: 12px;**

**box-shadow: 0 8px 25px rgba(0, 0, 0, 0.1);**

**border: none;**

**}**

**.card-header {**

**background: linear-gradient(to right, #007bff, #0056b3); /\* Gradien biru \*/**

**color: white;**

**border-bottom: none;**

**border-top-left-radius: 12px;**

**border-top-right-radius: 12px;**

**padding: 1.5rem 2rem;**

**font-size: 1.75rem;**

**font-weight: 600;**

**}**

**.btn-diagnosa {**

**background-color: #007bff;**

**border-color: #007bff;**

**color: white;**

**font-weight: 500;**

**padding: 0.75rem 1.5rem;**

**border-radius: 8px;**

**transition: background-color 0.3s ease;**

**}**

**.btn-diagnosa:hover {**

**background-color: #0056b3;**

**border-color: #0056b3;**

**}**

**.btn-reset {**

**background-color: #6c757d;**

**border-color: #6c757d;**

**color: white;**

**font-weight: 500;**

**padding: 0.75rem 1.5rem;**

**border-radius: 8px;**

**transition: background-color 0.3s ease;**

**}**

**.btn-reset:hover {**

**background-color: #545b62;**

**border-color: #545b62;**

**}**

**.form-label {**

**font-weight: 500;**

**color: #495057;**

**margin-bottom: 0.5rem;**

**}**

**.form-select {**

**border-radius: 8px;**

**padding: 0.75rem 1rem;**

**}**

**.alert-info {**

**background-color: #e6f7ff;**

**border-color: #91d5ff;**

**color: #004085;**

**border-radius: 8px;**

**}**

**.alert-info .alert-heading {**

**color: inherit;**

**}**

**.alert-warning {**

**background-color: #fff3cd;**

**border-color: #ffeeba;**

**color: #856404;**

**border-radius: 8px;**

**}**

**.alert-warning .alert-heading {**

**color: inherit;**

**}**

**.list-group-item strong {**

**color: #0056b3;**

**}**

**.symptom-selector {**

**margin-bottom: 1.5rem;**

**}**

**.card-footer {**

**background-color: #f8f9fa;**

**border-top: 1px solid #e9ecef;**

**border-bottom-left-radius: 12px;**

**border-bottom-right-radius: 12px;**

**}**

**</style>**

**</head>**

**<body>**

**<div class="container">**

**<div class="card">**

**<div class="card-header text-center">**

**🩺 Sistem Diagnosa Penyakit (Manusia) 🩺**

**</div>**

**<div class="card-body p-4 p-md-5">**

**<p class="text-muted text-center mb-4">**

**Pilih hingga 3 gejala utama yang Anda rasakan untuk mendapatkan kemungkinan diagnosa.**

**</p>**

**<form method="POST">**

**<div class="row">**

**<div class="col-md-4 symptom-selector">**

**<label for="input\_gejala\_1" class="form-label">Gejala Utama 1</label>**

**<select name="input\_gejala\_1" id="input\_gejala\_1" class="form-select" required>**

**<option value="">-- Pilih Gejala --</option>**

**{% for option\_text, option\_value in gejala\_options.items() %}**

**{% if option\_value %}**

**<option value="{{ option\_value }}">{{ option\_text }}</option>**

**{% endif %}**

**{% endfor %}**

**</select>**

**</div>**

**<div class="col-md-4 symptom-selector">**

**<label for="input\_gejala\_2" class="form-label">Gejala Utama 2 (Opsional)</label>**

**<select name="input\_gejala\_2" id="input\_gejala\_2" class="form-select">**

**<option value="">-- Pilih Gejala --</option>**

**{% for option\_text, option\_value in gejala\_options.items() %}**

**<option value="{{ option\_value }}">{{ option\_text }}</option>**

**{% endfor %}**

**</select>**

**</div>**

**<div class="col-md-4 symptom-selector">**

**<label for="input\_gejala\_3" class="form-label">Gejala Utama 3 (Opsional)</label>**

**<select name="input\_gejala\_3" id="input\_gejala\_3" class="form-select">**

**<option value="">-- Pilih Gejala --</option>**

**{% for option\_text, option\_value in gejala\_options.items() %}**

**<option value="{{ option\_value }}">{{ option\_text }}</option>**

**{% endfor %}**

**</select>**

**</div>**

**</div>**

**<div class="row mt-4">**

**<div class="col-md-6 mb-2 mb-md-0 d-grid">**

**<button type="submit" class="btn btn-diagnosa">Diagnosa Penyakit</button>**

**</div>**

**<div class="col-md-6 d-grid">**

**<button type="reset" class="btn btn-reset" onclick="window.location.href='/'">Reset Form</button>**

**</div>**

**</div>**

**</form>**

**{% if form\_submitted %}**

**<div class="mt-5">**

**<h5>Gejala yang Anda Pilih:</h5>**

**{% if selected\_symptoms\_for\_display %}**

**<ul class="list-group mb-4 shadow-sm">**

**{% for item in selected\_symptoms\_for\_display %}**

**<li class="list-group-item d-flex justify-content-between align-items-center">**

**{{ item.value }}**

**<span class="badge bg-primary rounded-pill">Dipilih</span>**

**</li>**

**{% endfor %}**

**</ul>**

**{% else %}**

**<p class="text-muted fst-italic">Anda belum memilih gejala apapun.</p>**

**{% endif %}**

**</div>**

**{% if hasil\_diagnosis %}**

**<div class="alert alert-info mt-4">**

**<h5 class="alert-heading fw-bold">🔍 Hasil Kemungkinan Diagnosa:</h5>**

**<p>Berdasarkan gejala yang Anda masukkan, berikut adalah beberapa kemungkinan penyakit.**

**<strong>Ini BUKAN diagnosis medis pasti.</strong>**

**</p>**

**<hr>**

**<ul class="list-group list-group-flush">**

**{% for h in hasil\_diagnosis %}**

**<li class="list-group-item bg-transparent px-0">**

**<strong class="d-block fs-5">{{ h.penyakit }}</strong>**

**<span class="text-muted">Saran Penanganan Umum:</span> {{ h.penanganan }}**

**</li>**

**{% endfor %}**

**</ul>**

**<hr>**

**<p class="mb-0 fw-bold">**

**Sangat disarankan untuk berkonsultasi dengan dokter atau tenaga medis profesional**

**untuk diagnosis yang akurat dan penanganan yang tepat.**

**</p>**

**</div>**

**{% elif form\_submitted and not selected\_symptoms\_for\_display %}**

**{# Tidak ada output khusus jika form disubmit tanpa gejala #}**

**{% elif form\_submitted %}**

**<div class="alert alert-warning mt-4" role="alert">**

**<h5 class="alert-heading fw-bold">⚠️ Tidak Ditemukan Penyakit yang Cocok</h5>**

**Mohon maaf, tidak ditemukan penyakit yang cocok dengan kombinasi gejala yang Anda pilih dalam basis data kami.**

**Silakan periksa kembali input Anda atau konsultasikan langsung dengan dokter.**

**</div>**

**{% endif %}**

**{% endif %}**

**</div>**

**<div class="card-footer text-center text-muted small py-3">**

**Sistem Pakar Sederhana &copy; 2024. Hanya untuk tujuan demonstrasi dan edukasi.**

**</div>**

**</div>**

**</div>**

**<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>**

**</body>**

**</html>**

**4.2 Antarmuka Pengguna**

Antarmuka pengguna pada sistem ini dirancang secara **sederhana dan intuitif**, agar mudah digunakan oleh siapa saja, termasuk pengguna non-teknis. Antarmuka terdiri dari beberapa komponen utama, yaitu:

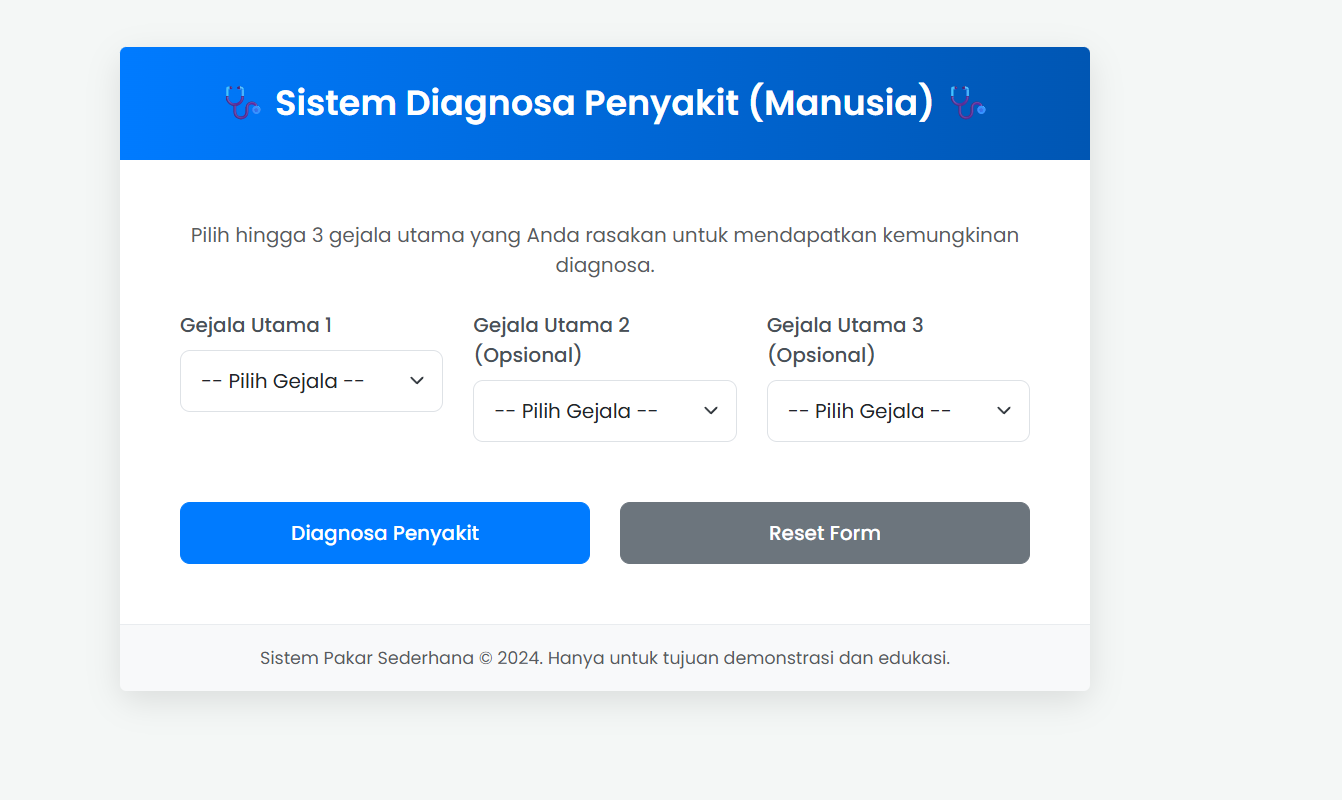
**Formulir input gejala yang dirasakan**  
Pengguna dapat memilih hingga tiga gejala dari daftar dropdown, seperti demam, batuk, mual, sakit kepala, dan lain sebagainya. Opsi gejala yang tersedia telah disesuaikan dengan data penyakit yang terdapat pada sistem.

**Tombol untuk memproses gejala**  
Setelah memilih gejala, pengguna cukup menekan tombol "Diagnosa" untuk mengirimkan data ke sistem dan memprosesnya berdasarkan aturan yang ada.

**Tampilan hasil diagnosis**  
Hasil diagnosis ditampilkan secara langsung di halaman yang sama, mencakup:

* Nama kemungkinan penyakit yang terdeteksi berdasarkan kombinasi gejala yang dimasukkan.
* Rekomendasi penanganan awal atau saran tindakan lanjutan, seperti istirahat, minum air hangat, atau segera ke fasilitas kesehatan.

Dengan pendekatan ini, sistem memberikan **pengalaman pengguna yang ringan namun informatif**, sehingga bisa digunakan sebagai alat bantu awal untuk mengenali penyakit berdasarkan gejala umum.



Foto/ss tampilan

**4.3 Alur Penggunaan Sistem**

* Pengguna membuka halaman sistem pakar diagnosis ikan lele.
* Pengguna mengisi form berdasarkan gejala yang diamati pada ikan.
* Sistem mengirimkan data ke backend Flask.
* Flask mencocokkan input dengan aturan dalam file CSV.
* Hasil diagnosa dan saran ditampilkan ke pengguna.

**5. Dataset**

**5.1 Sumber Data**

Data yang digunakan dalam sistem ini diperoleh melalui studi literatur, artikel ilmiah, dan referensi daring yang membahas gejala, jenis penyakit, dan metode penanganan pada ikan lele. Beberapa sumber berasal dari situs pertanian, forum peternak ikan, dan artikel dari Kementerian Kelautan dan Perikanan. Data yang dikumpulkan kemudian disusun secara manual ke dalam file data\_penyakit.csv yang disimpan dalam folder data.

**5.2 Contoh Struktur Data**

Kolom-kolom dalam dataset berisi informasi gejala dan diagnosis penyakit pada ikan lele, disertai dengan penanganan yang disarankan. Berikut contoh struktur dan isinya:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Gejala1** | **Gejala2** | **Gejala3** | **KemungkinanPenyakit** | **PenangananUmum** |
| Demam | Batuk | Pilek | Influenza (Flu) | Istirahat yang cukup, minum banyak air, konsumsi obat pereda gejala, hindari kontak. |
| Mual | Muntah | Diare | Keracunan Makanan / Gastroenteritis | Minum larutan oralit, hindari makanan padat, istirahat. Jika parah, ke dokter. |
| Demam | Sakit Kepala | Nyeri Otot | Demam Berdarah Dengue (DBD) - Fase Awal | Segera ke fasilitas kesehatan, banyak minum, istirahat total. |
| Pusing | Lemas | Kembung | "Masuk Angin" (Indisposisi Umum) | Istirahat, minum air hangat, kerokan, hindari angin langsung. |
| Sakit Tenggorokan | Demam |  | Radang Tenggorokan (Faringitis) | Minum air hangat, kumur antiseptik, hindari makanan iritatif, istirahat. |

File data\_penyakit.csv ini menggunakan format **Comma Separated Values (CSV)** dan memiliki struktur seperti contoh di atas.

**5.3 Penjelasan Kolom/Fitur**

### Penjelasan Kolom Dataset

**Gejala1**: Gejala pertama yang muncul atau dirasakan oleh pengguna (contoh: Demam, Batuk, Mual).  
**Gejala2**: Gejala kedua yang menyertai gejala pertama (contoh: Pilek, Sakit Kepala, Nyeri Sendi).  
**Gejala3**: Gejala ketiga yang memperkuat kemungkinan diagnosa (contoh: Lemas, Gatal, Muncul Bintik Merah).  
**KemungkinanPenyakit**: Nama penyakit yang mungkin diderita berdasarkan kombinasi gejala yang dimasukkan (contoh: Influenza, Demam Berdarah Dengue, Tifus, Pneumonia).  
**PenangananUmum**: Saran penanganan awal atau langkah-langkah yang dapat dilakukan oleh pengguna (contoh: Istirahat cukup, minum banyak air, konsultasi ke dokter, konsumsi obat pereda gejala).

Dataset ini dirancang untuk mendukung sistem pakar berbasis aturan (rule-based system) dengan cara **mencocokkan gejala-gejala yang dipilih pengguna** terhadap kombinasi gejala dalam dataset. Jika gejala yang dimasukkan sesuai dengan salah satu kombinasi aturan, maka sistem akan menampilkan diagnosa dan penanganan yang relevan.

**6. Hasil dan Analisis**

**6.1 Hasil Output Sistem**

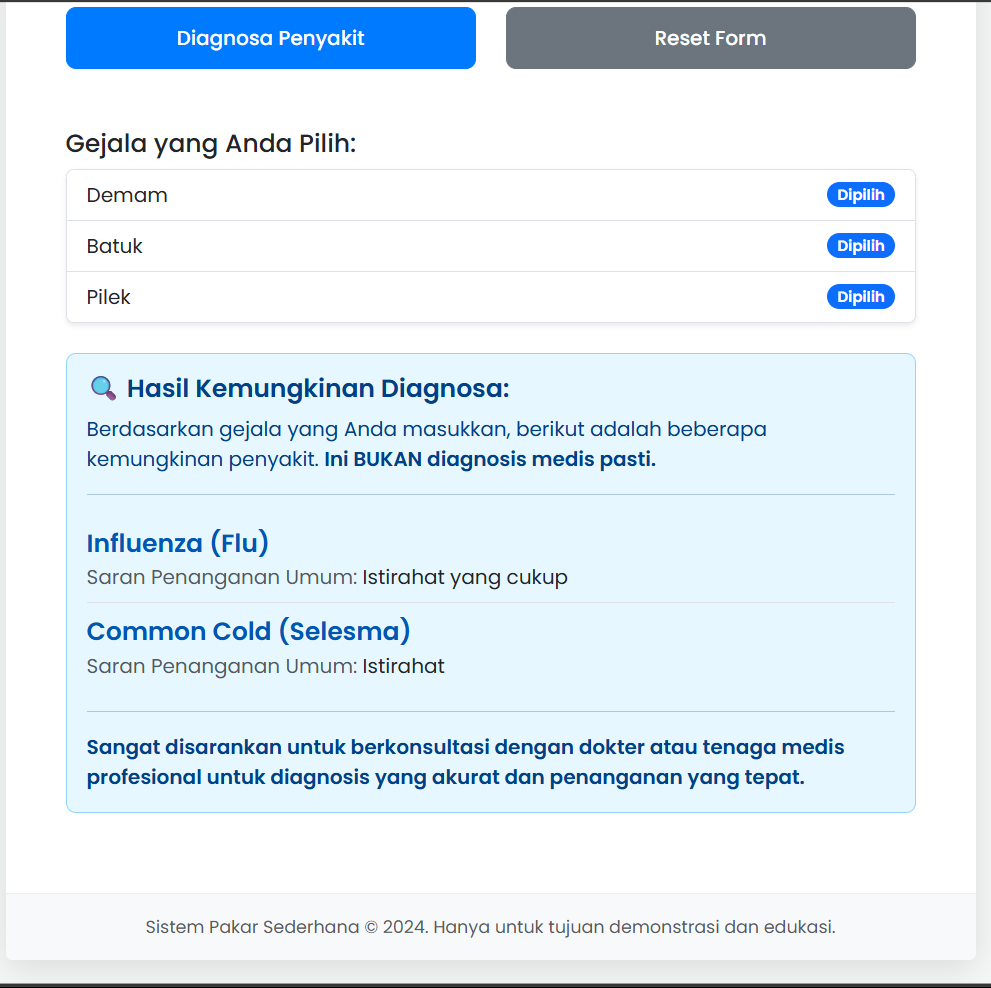
Setelah sistem diimplementasikan, pengguna dapat mengakses antarmuka web melalui browser. Pengguna akan diminta mengisi beberapa input berupa gejala-gejala yang dialami oleh ikan lele. Berdasarkan gejala yang dipilih, sistem akan mencocokkannya dengan aturan yang telah ditentukan dari dataset, lalu memberikan hasil diagnosis penyakit beserta saran penanganannya.

Contoh tampilan hasil ketika pengguna memilih:

* **Gejala 1**: Demam
* **Gejala 2**: Batuk
* **Gejala 3**: Pilek

Maka sistem akan menampilkan hasil seperti:

* **Diagnosa Penyakit**: Influenza (Flu)
* **Penanganan**: Istirahat yang cukup, minum banyak air, konsumsi obat pereda gejala (jika perlu), hindari kontak dengan orang lain.



Foto/ss hasil

**6.2 Analisis Hasil**

Dari pengujian yang dilakukan terhadap berbagai kombinasi input gejala, sistem menunjukkan hasil diagnosis yang **relevan** dan **konsisten** dengan basis aturan yang telah ditentukan. Sistem dapat memberikan diagnosis yang:

* Mendeteksi penyakit berdasarkan kombinasi gejala fisik dan perilaku
* Memberikan saran penanganan yang sesuai dengan penyakit terdeteksi
* Menangani kondisi "Sehat" apabila tidak ada gejala yang mencurigakan

Namun, terdapat beberapa hal penting yang perlu dicatat:

* **Keterbatasan Data**: Hasil diagnosis sangat bergantung pada kelengkapan data pada data\_penyakit.csv. Jika hanya mencakup beberapa kombinasi gejala, maka hasilnya akan terbatas dan tidak mencakup semua kasus nyata di lapangan.
* **Konsistensi Format**: Pada tahap awal, terdapat kesalahan pada pembacaan CSV karena adanya penulisan nilai yang tidak konsisten (seperti kapitalisasi atau spasi berlebih). Masalah ini telah diatasi dengan normalisasi input dan pembersihan data.
* **Sistem Rule-based**: Karena sistem ini bersifat rule-based, maka ia tidak mampu belajar atau beradaptasi dengan gejala baru secara otomatis seperti pada sistem berbasis machine learning. Namun demikian, sistem ini tetap **efektif** untuk digunakan dalam skenario dengan **jumlah gejala terbatas** dan **aturan yang jelas**, serta mudah diperbarui secara manual jika ada penambahan gejala atau penyakit baru.

Secara keseluruhan, sistem bekerja **dengan baik** untuk kebutuhan dasar **diagnosis penyakit ikan lele** berbasis gejala yang mudah dikenali oleh peternak atau pengguna umum.

**7. Kesimpulan dan Saran**

**7.1 Kesimpulan**

Sistem pakar **diagnosis penyakit ikan lele berbasis web** yang telah dikembangkan menggunakan bahasa pemrograman **Python** dan framework **Flask** ini mampu memberikan diagnosis penyakit berdasarkan input gejala yang dimasukkan oleh pengguna, seperti warna tubuh, nafsu makan, perilaku, luka, dan posisi berenang.

Sistem ini menggunakan pendekatan **berbasis aturan (rule-based)**, yang memiliki keunggulan dalam hal **kesederhanaan logika**, **kemudahan perawatan**, dan **kecepatan proses diagnosis**. Data aturan dan penyakit disimpan dalam format file **CSV**, yang fleksibel untuk diperbarui atau dikembangkan lebih lanjut.

Berdasarkan pengujian, sistem dapat menghasilkan diagnosis yang **relevan** dan **akurat** berdasarkan kombinasi gejala yang diinput, serta memberikan **saran penanganan** yang sesuai dengan penyakit terdeteksi. Antarmuka pengguna juga dirancang agar **mudah diakses** oleh peternak atau pengguna awam.

**7.2 Saran**

Berikut beberapa saran pengembangan lebih lanjut yang dapat diterapkan ke sistem di masa depan:

1. **Penambahan Data Penyakit dan Gejala**  
   Saat ini, jumlah kombinasi gejala dan penyakit masih terbatas. Menambahkan lebih banyak variasi penyakit dan gejala akan meningkatkan akurasi diagnosis.
2. **Penerapan Pembelajaran Mesin (Machine Learning)**  
   Untuk diagnosis yang lebih dinamis, penerapan algoritma seperti Decision Tree, Naive Bayes, atau K-Nearest Neighbor dapat dipertimbangkan agar sistem mampu belajar dari data kasus sebelumnya.
3. **Penyimpanan Riwayat Diagnosa**  
   Menambahkan fitur login dan database untuk menyimpan riwayat diagnosa pengguna akan sangat membantu untuk evaluasi dan tindak lanjut penyakit pada ikan.
4. **Peningkatan Antarmuka Pengguna (UI/UX)**  
   Tampilan antarmuka dapat ditingkatkan dengan framework frontend seperti Bootstrap atau Tailwind CSS agar terlihat lebih profesional, responsif, dan nyaman digunakan di perangkat mobile.
5. **Integrasi dengan Data Cuaca atau Kualitas Air**  
   Mengintegrasikan sistem dengan API eksternal seperti informasi suhu air, pH, atau cuaca dapat memperkaya proses diagnosis dan memberikan rekomendasi yang lebih kontekstual.

Link Video Demo Google Drive:  
https://drive.google.com/drive/folders/11xm\_wx1Dzm0r0kNGUpGN55Cl2JMUcND-?usp=drive\_link