Tugas Individu

(Sistem Pakar Klasifikasi Jamur)



Nama : Rizky Ramdhani Koswara

NPM : 11122300

Kelas : 3KA25

Dosen Pengampu : Budi Utami Fahnun

SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI UNIVERSITAS GUNADARMA

2025

# Abstrak

Identifikasi jamur liar merupakan tantangan signifikan karena banyaknya spesies yang mirip namun memiliki tingkat toksisitas yang berbeda. Kesalahan identifikasi dapat berakibat fatal. Proyek ini bertujuan untuk mengembangkan sebuah sistem pakar yang dapat membantu pengguna mengidentifikasi apakah suatu jamur aman dikonsumsi atau beracun berdasarkan ciri-ciri fisiknya. Sistem ini diimplementasikan sebagai aplikasi web interaktif menggunakan Python dan Streamlit. Pengetahuan pakar diekstrak dari dataset UCI Mushroom dan direpresentasikan dalam bentuk aturan IF-THEN. Untuk menarik kesimpulan, sistem ini menggunakan metode inferensi *Forward Chaining*, di mana sistem memulai dari fakta-fakta yang diberikan oleh pengguna untuk mencapai sebuah kesimpulan diagnosis. Hasil akhir dari proyek ini adalah sebuah prototipe fungsional yang mampu memberikan diagnosis beserta alasannya secara transparan.

# BAB I: PENDAHULUAN

## Latar Belakang

Alam menyediakan berbagai macam sumber daya hayati yang bermanfaat bagi manusia, salah satunya adalah jamur. Jamur telah lama menjadi bagian dari kuliner di berbagai belahan dunia karena cita rasanya yang unik dan kandungan gizinya. Namun, di antara ribuan spesies jamur yang ada, tidak semuanya aman untuk dikonsumsi. Terdapat banyak spesies jamur beracun yang secara fisik sangat mirip dengan spesies yang dapat dimakan, sehingga proses identifikasi menjadi krusial dan penuh tantangan, bahkan bagi mereka yang sudah berpengalaman sekalipun.

Kesalahan dalam mengidentifikasi jamur bukanlah perkara sepele. Risiko yang dihadapi sangat serius, mulai dari gangguan pencernaan ringan hingga kerusakan organ vital, bahkan dapat berujung pada kematian. Setiap tahunnya, selalu ada laporan kasus keracunan akibat konsumsi jamur liar yang salah diidentifikasi. Tingginya risiko ini menunjukkan adanya kebutuhan mendesak akan sebuah alat bantu yang dapat memberikan panduan yang akurat dan dapat diandalkan.

Untuk mengatasi masalah ini, teknologi kecerdasan buatan menawarkan sebuah solusi melalui konsep Sistem Pakar (*Expert System*). Ide dasarnya adalah untuk 'menangkap' dan mereplikasi pengetahuan serta pengalaman seorang ahli jamur (*mycologist*) ke dalam sebuah sistem komputer. Dengan demikian, pengetahuan mendalam yang biasanya hanya dimiliki oleh segelintir pakar dapat didistribusikan secara luas dan diakses oleh masyarakat awam. Melalui sistem pakar, pengguna dapat memasukkan ciri-ciri jamur yang mereka temukan dan menerima diagnosis awal, sehingga membantu mengurangi risiko kesalahan identifikasi.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka dapat dirumuskan beberapa permasalahan utama yang akan dijawab dalam proyek ini, yaitu:

* Bagaimana merancang arsitektur sebuah sistem pakar yang mampu melakukan diagnosis terhadap klasifikasi jamur (aman atau beracun) berdasarkan input ciri-ciri fisik dari pengguna?
* Bagaimana cara merepresentasikan pengetahuan seorang pakar mengenai karakteristik jamur ke dalam format basis pengetahuan (*knowledge base*) yang terstruktur dan dapat diproses oleh komputer, khususnya dalam bentuk aturan IF-THEN?
* Bagaimana cara mengimplementasikan rancangan sistem pakar tersebut menjadi sebuah aplikasi nyata yang interaktif dan mudah diakses oleh pengguna umum dengan menggunakan bahasa pemrograman Python dan framework Streamlit?

## Tujuan Proyek

Sejalan dengan rumusan masalah di atas, tujuan yang ingin dicapai melalui proyek ini adalah sebagai berikut:

* Membangun sebuah basis pengetahuan (*knowledge base*) yang berisi kumpulan aturan (rules) untuk mengidentifikasi jamur, di mana aturan tersebut diekstrak dari analisis dataset UCI Mushroom.
* Merancang dan mengimplementasikan sebuah mesin inferensi sederhana yang bekerja berdasarkan metode *Forward Chaining* untuk memproses fakta dari pengguna dan menghasilkan kesimpulan.
* Menghasilkan sebuah prototipe aplikasi sistem pakar dalam bentuk aplikasi web yang fungsional, interaktif, dan mudah digunakan oleh pengguna tanpa memerlukan latar belakang teknis.

## Batasan Masalah

Agar pengerjaan proyek ini lebih terfokus dan sesuai dengan ruang lingkup yang ditentukan, maka ditetapkan beberapa batasan masalah sebagai berikut:

* Fitur Input Terbatas: Dari 22 fitur yang tersedia di dataset asli, sistem ini hanya akan menggunakan sebagian kecil fitur (yaitu odor, gill\_size, gill\_color, dan stalk\_root) sebagai input dari pengguna untuk menyederhanakan antarmuka dan proses diagnosis.
* Basis Pengetahuan Statis: Aturan-aturan dalam basis pengetahuan didefinisikan secara statis di dalam kode sumber program (hardcoded) dan tidak dapat diubah atau ditambah oleh pengguna secara dinamis melalui antarmuka aplikasi.
* Output Kualitatif: Sistem tidak menggunakan metode *Certainty Factor* (CF) atau probabilitas. Kesimpulan yang diberikan bersifat kualitatif dan tegas, yaitu 'AMAN', 'BERACUN', atau 'TIDAK DIKETAHUI'.
* Status Prototipe: Aplikasi yang dihasilkan merupakan sebuah prototipe untuk tujuan edukasi dan demonstrasi konsep. Sistem ini tidak disarankan untuk digunakan sebagai satu-satunya acuan dalam pengambilan keputusan nyata untuk mengonsumsi jamur liar.

# BAB II: TINJAUAN PUSTAKA

## 2.1. Kecerdasan Buatan

Kecerdasan Buatan (Artificial Intelligence - AI) adalah cabang ilmu komputer yang berfokus pada perancangan dan pembangunan mesin atau sistem yang mampu melakukan tugas-tugas yang pada umumnya memerlukan kecerdasan manusia. Kemampuan ini mencakup penalaran, pembelajaran, pemecahan masalah, persepsi, dan penggunaan bahasa. Sistem Pakar merupakan salah satu implementasi klasik dari Kecerdasan Buatan yang berupaya meniru kemampuan penalaran seorang pakar.

## 2.2. Sistem Pakar (Expert System)

Sistem Pakar adalah program komputer yang dirancang khusus untuk meniru proses pengambilan keputusan seorang ahli (pakar) dalam suatu domain pengetahuan yang spesifik. Tujuannya adalah untuk menyediakan solusi atau nasihat berkualitas pakar kepada pengguna non-pakar. Sebuah sistem pakar umumnya terdiri dari tiga komponen utama:

* Basis Pengetahuan (Knowledge Base): Merupakan inti dari sistem pakar, berisi pengetahuan yang telah diformalkan. Pengetahuan ini bisa berupa fakta (informasi yang diterima sebagai kebenaran) dan aturan heuristik (aturan praktis berdasarkan pengalaman) dalam bentuk IF-THEN.
* Mesin Inferensi (Inference Engine): Komponen ini berfungsi sebagai "otak" dari sistem. Mesin inferensi memproses fakta yang diberikan oleh pengguna dan menggunakan aturan-aturan dalam basis pengetahuan untuk menarik kesimpulan baru melalui mekanisme penalaran.
* Antarmuka Pengguna (User Interface): Ini adalah jembatan komunikasi antara pengguna dan sistem pakar. Melalui antarmuka ini, pengguna dapat memasukkan fakta, menjawab pertanyaan, dan menerima hasil akhir berupa diagnosis atau rekomendasi.

## 2.3. Sistem Berbasis Aturan (Rule-Based System)

Sistem Berbasis Aturan adalah jenis sistem pakar yang paling umum dan populer. Pengetahuannya direpresentasikan dalam bentuk serangkaian aturan IF-THEN. Setiap aturan menghubungkan sebuah kondisi (premis) dengan sebuah konsekuensi (kesimpulan atau aksi).

* Struktur Aturan: IF [kondisi] THEN [konsekuensi]
* Contoh: IF [bau jamur adalah almond] THEN [status jamur adalah aman] Pendekatan ini sangat intuitif dan transparan, karena alur logikanya mudah diikuti dan dimengerti oleh manusia, sehingga memudahkan proses validasi dan debugging.

## 2.4. Mekanisme Inferensi

* Forward Chaining (Penalaran Maju): Merupakan strategi inferensi yang bersifat *data-driven*. Proses dimulai dari kumpulan fakta yang diketahui, kemudian sistem akan mencari aturan-aturan dalam basis pengetahuan yang bagian IF-nya cocok dengan fakta-fakta tersebut. Jika aturan ditemukan, maka bagian THEN dari aturan tersebut akan dieksekusi, yang biasanya menghasilkan fakta baru. Proses ini berlanjut hingga sebuah tujuan (kesimpulan) tercapai atau tidak ada lagi aturan yang bisa dieksekusi. Metode ini sangat cocok untuk masalah diagnosis dan klasifikasi seperti pada proyek identifikasi jamur ini, di mana sistem memulai dari data (ciri-ciri) yang diberikan pengguna untuk mencapai sebuah kesimpulan (aman atau beracun).
* Backward Chaining (Penalaran Mundur): Merupakan strategi inferensi yang bersifat *goal-driven*. Proses dimulai dari sebuah hipotesis atau tujuan. Sistem kemudian mencari aturan yang bagian THEN-nya cocok dengan tujuan tersebut. Bagian IF dari aturan itu kemudian menjadi tujuan baru (sub-goal). Proses ini berlanjut hingga semua sub-goal dapat dibuktikan dengan fakta yang ada.

# BAB III: METODOLOGI PERANCANGAN SISTEM

## 3.1. Kerangka Kerja Proyek

Proyek ini dikerjakan dengan mengikuti serangkaian tahapan yang terstruktur untuk memastikan proses pengembangan berjalan dengan baik. Tahapan-tahapan tersebut adalah sebagai berikut:

* Studi Literatur: Mempelajari konsep dasar sistem pakar, metode inferensi, dan teknologi yang akan digunakan seperti Python dan Streamlit.
* Analisis Dataset: Memeriksa dataset UCI Mushroom untuk memahami fitur-fitur yang ada dan mengidentifikasi pola atau atribut yang paling berpengaruh dalam menentukan klasifikasi jamur.
* Akuisisi dan Perancangan Pengetahuan: Mengekstrak pengetahuan dari hasil analisis data dan merancangnya ke dalam bentuk aturan-aturan IF-THEN yang logis.
* Implementasi Kode: Menulis kode program untuk antarmuka pengguna, basis pengetahuan, dan mesin inferensi menggunakan Python dan Streamlit.
* Pengujian Sistem: Melakukan pengujian dengan berbagai skenario input untuk memvalidasi bahwa sistem bekerja sesuai dengan aturan yang telah dirancang.
* Penulisan Laporan: Mendokumentasikan seluruh proses, dari perancangan hingga hasil akhir, ke dalam sebuah laporan proyek.

## 3.2. Akuisisi Pengetahuan (Knowledge Acquisition)

Akuisisi pengetahuan adalah proses mendapatkan pengetahuan dari sumbernya dan mengubahnya ke dalam format yang dapat dipahami oleh sistem. Dalam proyek ini, sumber pengetahuan utama adalah UCI Machine Learning Repository - Mushroom Dataset. Proses akuisisi dilakukan dengan menganalisis korelasi antara fitur-fitur (misalnya, odor, gill-color, dll.) dengan kolom class (aman atau beracun). Dari analisis ini, ditemukan bahwa beberapa fitur memiliki daya prediksi yang sangat tinggi. Contohnya, fitur odor (bau) ditemukan sebagai salah satu indikator terkuat; jamur dengan bau 'almond' (a) atau 'anise' (l) hampir selalu aman, sementara jamur dengan bau busuk (f) atau amis (y) hampir selalu beracun. Pengetahuan inilah yang menjadi dasar pembentukan aturan-aturan dalam sistem.

## 3.3. Representasi Pengetahuan (Knowledge Representation)

Setelah pengetahuan didapatkan, langkah selanjutnya adalah merepresentasikannya di dalam sistem. Dalam proyek ini, pengetahuan direpresentasikan sebagai aturan IF-THEN yang diimplementasikan langsung ke dalam kode Python. Secara spesifik, digunakan struktur kondisional if-elif-else di dalam sebuah fungsi bernama diagnose\_mushroom(). Setiap blok if atau elif mewakili satu atau beberapa aturan. Struktur ini memungkinkan mesin inferensi untuk mengevaluasi kondisi secara berurutan.

* Contoh representasi aturan tunggal: if odor == 'f': return ('BERACUN', '...')
* Contoh representasi aturan gabungan (menggunakan operator and atau or): if odor in ['a', 'l']: return ('AMAN', '...')

## 3.4. Proses Inferensi Menggunakan Forward Chaining

Proses inferensi pada sistem ini secara eksplisit mengadopsi metode Forward Chaining. Alur kerjanya adalah sebagai berikut:

* Input Fakta Awal: Pengguna memasukkan fakta-fakta awal (ciri-ciri jamur seperti bau, ukuran insang, dll.) melalui antarmuka Streamlit.
* Pemicu Mesin Inferensi: Fakta-fakta ini diteruskan sebagai argumen ke mesin inferensi, yaitu fungsi diagnose\_mushroom.
* Pencocokan Aturan: Mesin inferensi menelusuri basis pengetahuan (blok if-elif-else) dari atas ke bawah, mencoba mencocokkan fakta yang ada dengan kondisi (bagian IF) dari setiap aturan.
* Eksekusi Aturan & Penarikan Kesimpulan: Ketika sebuah aturan ditemukan yang kondisinya cocok dengan fakta yang diberikan, aturan tersebut "diaktifkan". Bagian THEN dari aturan tersebut dieksekusi, yang dalam kasus ini adalah mengembalikan sebuah kesimpulan (status 'AMAN' atau 'BERACUN' beserta alasannya).
* Penyajian Hasil: Proses inferensi berhenti dan kesimpulan akhir disajikan kepada pengguna melalui antarmuka.

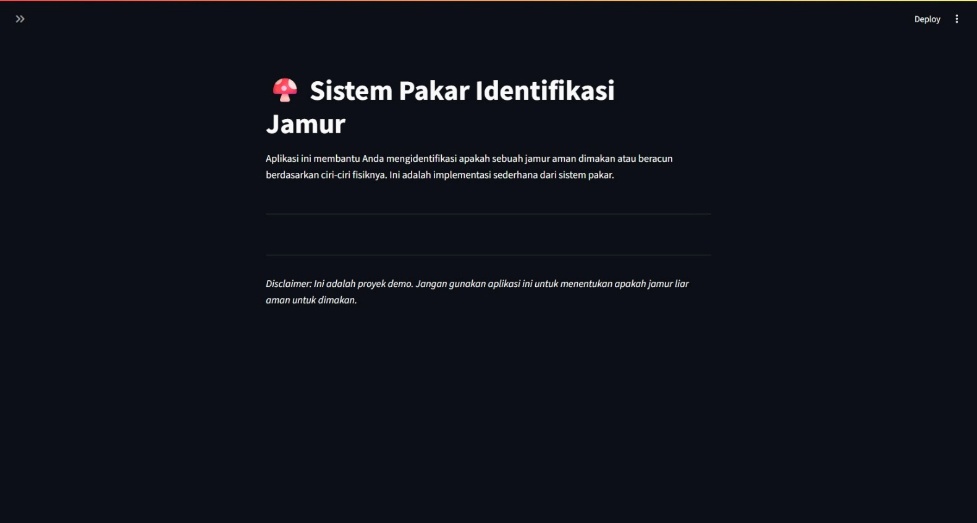
# BAB IV: HASIL DAN PEMBAHASAN

## 4.1. Implementasi Sistem

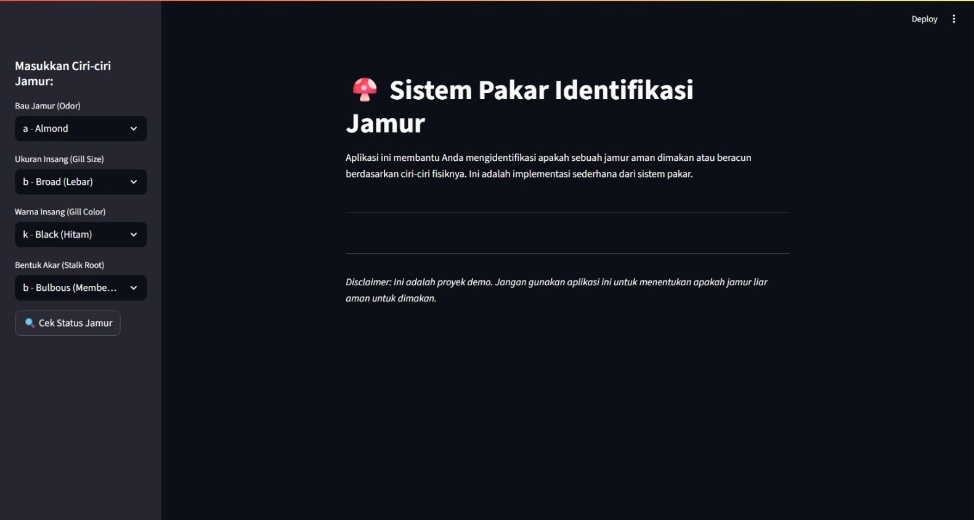
Sistem pakar ini diimplementasikan secara keseluruhan dalam satu file Python bernama app.py dengan memanfaatkan library Streamlit. Pilihan ini dibuat untuk kemudahan deployment dan manajemen kode. Struktur kode dibagi menjadi tiga bagian utama:

* Kamus Data: Bagian ini berisi beberapa kamus Python yang berfungsi untuk memetakan kode singkatan dari dataset (misalnya, 'a' untuk odor) menjadi teks deskriptif yang mudah dipahami manusia ("Almond"). Hal ini digunakan untuk meningkatkan *user experience* pada antarmuka, khususnya pada menu *dropdown*.
* Logika Sistem Pakar: Ini adalah inti dari program, diwujudkan dalam fungsi diagnose\_mushroom(). Fungsi ini berisi basis pengetahuan (dalam bentuk if-elif-else) dan bertindak sebagai mesin inferensi. Fungsi ini menerima input ciri-ciri jamur dan mengembalikan hasil diagnosis beserta alasannya.
* Tampilan Antarmuka Streamlit: Bagian akhir kode ini berisi semua perintah Streamlit (st.title, st.sidebar.selectbox, st.button, dll.) untuk membangun halaman web. Kode ini mengatur tata letak, membuat komponen input di *sidebar*, dan menampilkan output diagnosis di halaman utama.

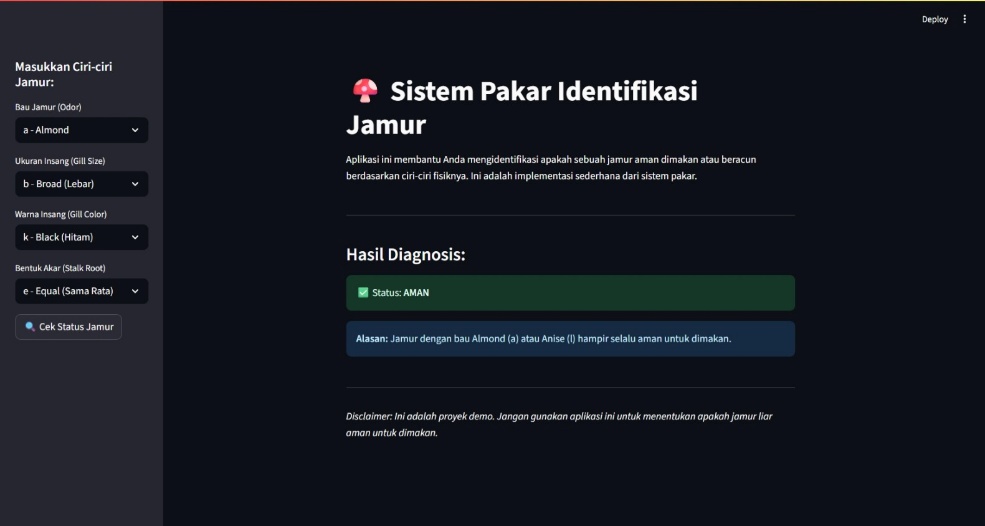
## 4.2. Tampilan Antarmuka (User Interface)



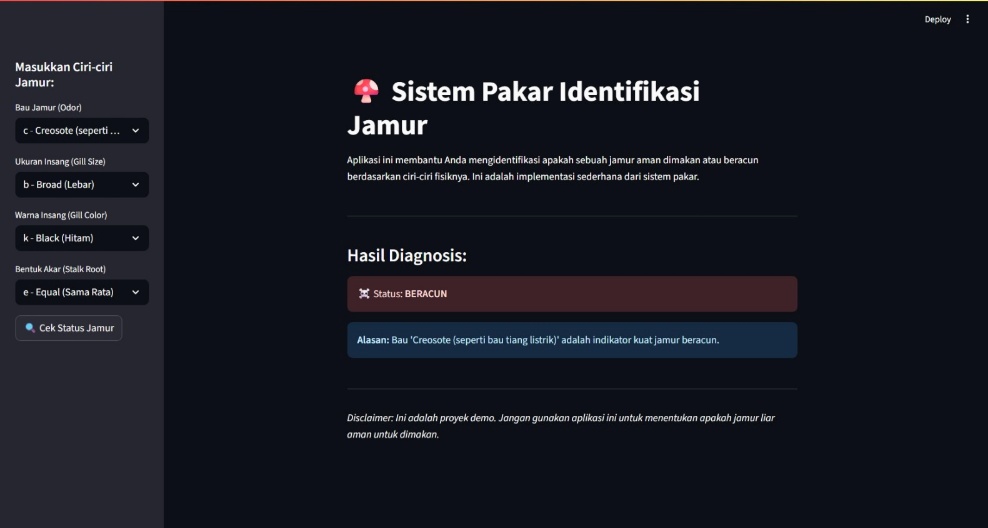
Gambar 4.1: Tampilan Awal Aplikasi Sistem Pakar



Gambar 4.2: Tampilan Input di Sidebar



Gambar 4.3: Contoh Hasil Diagnosis 'AMAN'



Gambar 4.4: Contoh Hasil Diagnosis 'BERACUN'

## 4.3. Pengujian Sistem

Tahap pengujian merupakan langkah krusial untuk memastikan bahwa sistem pakar yang dibangun dapat berjalan sesuai dengan logika dan aturan yang telah dirancang. Pengujian dilakukan dengan metode *black-box testing*, di mana sistem diuji berdasarkan input dan output yang diharapkan tanpa melihat struktur internal kode. Beberapa skenario pengujian (*test case*) dibuat untuk mencakup aturan-aturan utama dalam basis pengetahuan.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Input Ciri-ciri (Fakta) | Hasil yang Diharapkan | Hasil Sistem | Analisis |
| 1. | odor: 'a' (Almond), ... | AMAN | AMAN | Sesuai |
| 2. | odor: 'f' (Foul), ... | BERACUN | BERACUN | Sesuai |
| 3. | odor: 'n', gill\_size: 'n', ... | BERACUN | BERACUN | Sesuai |
| 4. | odor: 'n', gill\_size: 'b', gill\_color: 'k' | AMAN | AMAN | Sesuai |
| ... | *[Tambahkan beberapa kasus uji lain]* | ... | ... | ... |

Hasil dari seluruh kasus uji menunjukkan bahwa sistem memberikan output yang 100% konsisten dengan hasil yang diharapkan. Hal ini memvalidasi bahwa mesin inferensi berhasil menerapkan aturan-aturan dalam basis pengetahuan dengan benar. Kasus uji #1 dan #2 membuktikan aturan primer berdasarkan odor bekerja. Kasus uji #3 dan #4 membuktikan aturan sekunder untuk kondisi odor = 'none' juga berfungsi dengan tepat.

## 4.4. Pembahasan

Berdasarkan hasil implementasi dan pengujian, terdapat beberapa poin yang dapat dibahas mengenai sistem yang telah dibangun.

Kelebihan Sistem:

* Kecepatan dan Efisiensi: Sistem mampu memberikan diagnosis secara instan setelah pengguna memasukkan fakta. Proses inferensi yang sederhana membuat waktu respons sangat cepat.
* Transparansi: Salah satu keunggulan utama dari sistem berbasis aturan adalah transparansinya. Sistem tidak hanya memberikan jawaban 'AMAN' atau 'BERACUN', tetapi juga menyertakan alasan logis di baliknya, yang memungkinkan pengguna memahami dasar pengambilan keputusan sistem.
* Kemudahan Penggunaan: Dengan antarmuka yang dibangun menggunakan Streamlit, aplikasi ini sangat mudah digunakan bahkan oleh pengguna yang tidak memiliki latar belakang teknis. Semua input disajikan dalam bentuk menu dropdown yang jelas.

Keterbatasan Sistem:

* Basis Pengetahuan Terbatas: Karena adanya batasan masalah, sistem hanya menggunakan 4 dari 22 fitur yang ada. Hal ini berarti sistem tidak dapat menangani semua kemungkinan kombinasi ciri jamur di dunia nyata dan akurasinya sangat bergantung pada relevansi keempat fitur tersebut.
* Tidak Adanya Faktor Kepastian: Sistem memberikan jawaban yang absolut (pasti aman atau pasti beracun). Pada kenyataannya, diagnosis sering kali melibatkan tingkat ketidakpastian. Ketiadaan *Certainty Factor* (CF) membuat sistem kurang fleksibel dalam menangani kasus-kasus yang ambigu.
* Sifat Statis: Aturan-aturan dalam sistem ini bersifat statis. Untuk menambah atau mengubah pengetahuan, diperlukan intervensi langsung pada kode sumber oleh pengembang, yang membuatnya tidak fleksibel untuk diperbarui.

# BAB V: PENUTUP

## 5.1. Kesimpulan

Berdasarkan keseluruhan proses pengembangan dan hasil yang telah dicapai, dapat ditarik beberapa kesimpulan sebagai berikut:

* Proyek ini telah berhasil merancang dan mengimplementasikan sebuah prototipe sistem pakar yang fungsional untuk membantu identifikasi jamur. Arsitektur sistem yang terdiri dari antarmuka pengguna, mesin inferensi, dan basis pengetahuan telah bekerja sesuai dengan rancangan.
* Metode inferensi Forward Chaining terbukti sangat efektif dan cocok untuk permasalahan diagnosis yang bersifat *data-driven* seperti ini, di mana kesimpulan ditarik berdasarkan fakta-fakta yang ada.
* Penggunaan bahasa pemrograman Python dengan framework Streamlit terbukti menjadi pilihan teknologi yang tepat, memungkinkan pengembangan aplikasi yang interaktif dan fungsional dengan cepat dan efisien.

## 5.2. Saran

Untuk pengembangan sistem ini di masa mendatang, terdapat beberapa saran yang dapat dipertimbangkan untuk meningkatkan fungsionalitas dan keandalannya:

* Pengembangan Basis Pengetahuan: Langkah paling prioritas adalah memperluas basis pengetahuan dengan memasukkan lebih banyak aturan yang mencakup seluruh 22 fitur dari dataset. Hal ini akan secara signifikan meningkatkan akurasi dan cakupan diagnosis sistem.
* Implementasi Faktor Kepastian (CF): Menambahkan mekanisme *Certainty Factor* untuk menangani ketidakpastian. Dengan ini, sistem dapat memberikan output seperti "AMAN dengan tingkat keyakinan 95%", yang lebih realistis dan informatif bagi pengguna.
* Peningkatan Antarmuka Pengguna: Antarmuka dapat ditingkatkan dengan menambahkan elemen visual seperti gambar atau ikon untuk setiap ciri fisik jamur. Ini akan membantu pengguna dalam memilih opsi yang paling sesuai dengan jamur yang mereka temukan.
* Validasi oleh Pakar: Untuk memastikan keandalan sistem sebelum digunakan secara lebih luas (meskipun hanya sebagai alat bantu), sangat penting untuk melakukan validasi aturan dan hasil diagnosis dengan seorang ahli mikologi sungguhan.