TSAQOFAH
Jurnal Penelitian Guru Indonesia

p-ISSN: 2810-0395
e-ISSN: 2810-0042

Terindeks : Dimensions, Scilit, Lens, Semantic, Garuda, Google Scholar, Base, etc

https://doi.org/10.58578/tsaqofah.v5i5.6800

PENGEMBANGAN APLIKASI SISTEM PAKAR BERBASIS WEB UNTUK ANALISIS PENYAKIT TANAMAN CABAI DENGAN MODEL *FORWARD CHAINING*

Development of a Web-Based Expert System Application for Chili Plant Disease Analysis Using the Forward Chaining Model

Rafi Kurnia Ilahi & Agariadne Dwinggo Samala
Universitas Negeri Padang
rafikurniailahi076@gmail.com

Article Info:

Submitted:	Revised:	Accepted:	Published:
Jun 10, 2025	Jul 5, 2025	Jul 17, 2025	Jul 22, 2025

Abstract

Expert systems are a branch of artificial intelligence designed to replicate the reasoning abilities of a human expert in solving specific problems within a particular domain. This study aims to develop a web-based expert system using the forward chaining model to analyze and diagnose diseases in chili plants. The system is built using a rule-based approach, where users select observed symptoms and the system automatically generates a diagnosis through forward reasoning. Testing was conducted through black-box testing to ensure all features functioned as expected, and accuracy testing was performed by experts using ten real case studies. The results showed that the system successfully provided accurate diagnoses for all cases, achieving an accuracy rate of 100%. These findings demonstrate that the developed expert system is effective as a knowledge-based diagnostic tool. Furthermore, the system holds significant potential for expansion to other problem areas through adjustments to its knowledge base.

Tsaqofah journal is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

Keywords: Expert System; Forward Chaining; Artificial Intelligence; Plant Disease Diagnosis; Chili Plants

Abstrak: Sistem pakar merupakan salah satu cabang kecerdasan buatan yang dirancang untuk meniru kemampuan seorang pakar dalam memecahkan masalah spesifik pada domain tertentu. Penelitian ini bertujuan untuk membangun sistem pakar berbasis web dengan model *forward chaining* yang dapat melakukan analisis dan diagnosis terhadap penyakit tanaman cabai. Sistem dikembangkan menggunakan pendekatan berbasis aturan, di mana pengguna memilih gejala yang diamati dan sistem secara otomatis menghasilkan diagnosis berdasarkan proses penalaran ke depan. Pengujian dilakukan melalui *blackbox testing* untuk memastikan semua fitur berfungsi sesuai skenario, serta pengujian akurasi oleh pakar menggunakan sepuluh studi kasus nyata. Hasil pengujian menunjukkan bahwa sistem berhasil memberikan diagnosis yang tepat untuk seluruh kasus dengan tingkat akurasi 100%. Temuan ini membuktikan bahwa sistem pakar yang dibangun efektif digunakan sebagai alat bantu diagnosis berbasis pengetahuan. Selain itu, sistem ini memiliki potensi untuk dikembangkan lebih lanjut pada berbagai permasalahan lain dengan penyesuaian pada basis pengetahuannya.

Kata Kunci: Sistem Pakar; *Forward Chaining*; Kecerdasan Buatan; Diagnosis Penyakit Tanaman; Tanaman Cabai

PENDAHULUAN

Salah satu teknologi yang berkembang dan menjadi pusat perhatian saat ini yaitu Kecerdasan Buatan (*Artificial Intelligence*/AI). Kecerdasan Buatan memiliki beberapa cabang ilmu, salah satu diantaranya yaitu Sistem Pakar (*Expert System*). Sistem Pakar adalah sebuah program komputer yang dirancang untuk meniru atau mensimulasikan pengetahuan (*knowledge*) dan keterampilan (*skill*) dari seorang pakar pada bidang keahlian tertentu. Selanjutnya sistem ini akan mencoba memecahkan suatu permasalahan sesuai dengan kepakarannya (Irawan, 2007:1).

Sistem pakar telah banyak digunakan untuk memecahkan masalah di berbagai bidang, seperti bidang pertanian, perbankan, farmasi, industri dan pendidikan (Indryani et al., 2017). Di bidang industri, misalnya, penelitian yang dilakukan oleh Dedi Suryadi dan tim di Universitas Bengkulu dengan judul "Sistem Pakar untuk Mengidentifikasi Kerusakan Mesin Industri" menyimpulkan bahwa sistem pakar yang dirancang mampu mengindentifikasi kerusakan mesin berdasarkan karakteristik getaran yang menyatakan akurasi mencapai 100% (Suryadi et al., 2018). Sementara itu, di bidang kesehatan, penelitian oleh Nur Aini dan tim dari Universitas Mulawarman, dengan judul "Sistem Pakar Pendiagnosa Penyakit

Tuberkulosis" menunjukkan tingkat akurasi diagnosa sebesar 85% (Aini et al., 2017). Berbagai penelitian tersebut menunjukkan bahwa sistem pakar mampu menyelesaikan permasalahan dengan efektif sesuai domain kepakarannya.

Sistem pakar memerlukan mekanisme logika untuk melakukan proses penalaran (*reasoning*) berdasarkan basis pengetahuan (*knowledge base*) dan data/fakta yang diberikan oleh pengguna. Mekanisme logika ini dikenal sebagai Model Inferensi. Dalam sistem pakar model inferensi dianggap sebagai otak dari sistem pakar, berupa perangkat lunak yang melakukan tugas inferensi penalaran sistem pakar (Kuswanto, 2022:18). Ada 2 bentuk pendekatan basis pengetahuan yang sangat umum digunakan yaitu *Rule-Based Reasoning* dan *Case-Based Reasoning*. Pada *Rule-Based Reasoning*, pengetahuan direpresentasikan dengan menggunakan aturan berbentuk: IF-THEN. Sementara itu *Case-Based Reasoning*, basis pengetahuan akan berisi solusi-solusi yang telah di capai sebelumnya, kemudian akan diturunkan suatu solusi untuk keadaan yang terjadi sekarang (Kuswanto, 2022:17).

Penalaran berbasis aturan (*Rule-Based Reasoning*) memiliki beberapa model inferensi, salah satunya yaitu *Forward Chaining*. Model ini sering disebut sebagai *data-driven*, yaitu proses penyelesaian sebuah masalah dengan mengumpulkan informasi untuk kemudian ditarik kesimpulan (Ardiansyah et al., 2022:18). Pada proses ini akan dilakukan pengecekan terhadap setiap *rule* untuk melihat apakah data yang sedang diobservasi memenuhi premis dari *rule* tersebut. Apabila memenuhi, maka *rule* akan dieksekusi untuk menghasilkan fakta baru yang mungkin akan digunakan oleh *rule* yang lain (Irawan, 2007:35). Model *Forward Chaining* cocok digunakan untuk sistem yang memiliki banyak hipotesa keluaran dan data (Akli, 2017).

Banyak penelitian yang menunjukkan efektivitas metode *Forward Chaining* dalam sistem pakar. Misalnya, penelitian oleh Fransiskus Panca Juniawan mengenai diagnosa gangguan kejiwaan menunjukkan bahwa metode ini mampu bekerja dengan baik (Juniawan, 2017). Penelitian lain oleh Sopi Sapriadi dan tim mengenai diagnosa gaya belajar mahasiswa membuktikan bahwa sistem pakar dengan metode *Forward Chaining* mencapai tingkat kesesuaian hingga 90% (Sapriadi et al., 2023). Selain itu, Doni Abdul Fatah dan tim juga membuktikan bahwa metode ini sangat efektif digunakan dalam diagnosa penyakit pada tanaman bawang merah dan cabai, dengan tingkat akurasi 90% berdasarkan 10 data penyakit yang diuji (Fatah et al., 2023) . Hasil-hasil tersebut menunjukkan bahwa model *Forward Chaining* sangat cocok diterapkan dalam aplikasi sistem pakar, termasuk dalam bidang pertanian.

Aplikasi sistem pakar yang dibangun pada penelitian ini bertujuan untuk membantu petani dalam melakukan analisa penyakit tanaman cabai menggunakan model *forward chaining*. Jenis cabai yang digunakan untuk objek penelitian ini yaitu cabai keriting (*Capsicum annum L.*). Alasan kita memilih cabai keriting (*Capsicum annum L.*) sebagai objek penelitian ini selain disebabkan banyaknya budidaya tanaman cabai keriting di daerah Sumatera Barat juga disebabkan karena tantangan yang dihadapi oleh petani dalam budidaya cabai keriting. Usaha budidaya tanaman cabai yang berhasil memang menjanjikan keuntungan yang menarik. Namun tidak sedikit petani yang gagal panen bahkan mengalami kerugian dalam budidaya cabai. Untuk mengantisipasi kemungkinan kegagalan diperlukan keterampilan dalam penerapan pengetahuan dan teknik budidaya cabai sesuai dengan daya dukung (Piay et al., 2010: 2).

Berdasarkan wawancara dengan bapak Ichsan Kurniawan, pakar tanaman cabai dari Balai Penyuluh Pertanian Kecamatan IV Koto terdapat beberapa tantangan yang dihadapi dalam budidaya tanaman cabai saat ini, diantaranya yaitu kurangnya pengetahuan dalam budidaya tanaman cabai oleh petani, terbatasnya akses informasi terpercaya dan akses ke pakar tanaman cabai, serta keterbatasan waktu dan petugas dari dinas pertanian untuk melakukan penyuluhan. Sehingga pengembangan aplikasi sistem pakar untuk analisis penyakit tanaman cabai menggunakan model *forward chaining* ini diharapkan dapat membantu mengatasi permasalahan tersebut.

Aplikasi sistem pakar ini tidak hanya menyediakan fitur analisis penyakit tanaman cabai, tetapi juga dilengkapi dengan fitur panduan penanaman cabai dan informasi produk pertanian. Penambahan fitur-fitur tersebut menjadi pembeda utama dari penelitian-penelitian sebelumnya dan diharapkan dapat meningkatkan kebermanfaatan aplikasi bagi petani. Aplikasi sistem pakar ini juga dibangun berbasis *website* agar mudah diakses oleh petani menggunakan *smartphone* mereka tanpa harus menginstal aplikasi terlebih dahulu.

METODE

Penelitian ini menggunakan metode prototyping dalam pengembangan sistem pakar berbasis web untuk mendiagnosis penyakit tanaman cabai. Metode prototyping dipilih karena mampu memberikan fleksibilitas dalam pengembangan sistem yang bersifat interaktif, iteratif, dan sesuai kebutuhan pengguna akhir. Tahapan awal dimulai dari proses communication, yaitu identifikasi kebutuhan pengguna melalui analisis sistem yang sedang

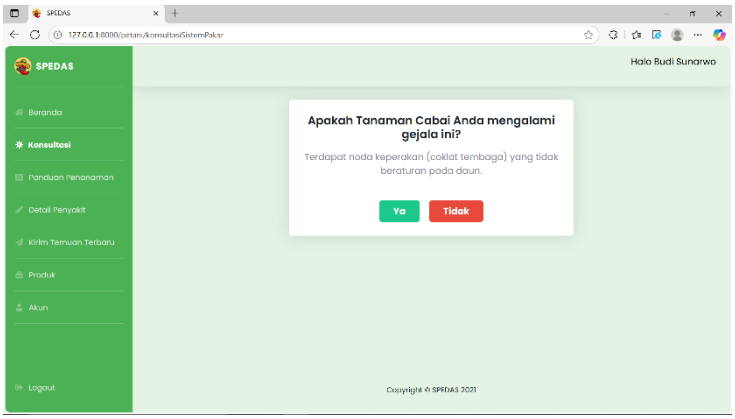
berjalan, analisis pelaku sistem, serta identifikasi masalah dan solusi. Tahapan ini menghasilkan dokumentasi kebutuhan dan perancangan sistem yang akan dikembangkan.

Selanjutnya, pada tahap quick design, dilakukan perancangan model sistem melalui use case diagram, activity diagram, perancangan database (melalui proses normalisasi hingga bentuk 3NF dan BCNF), serta penyusunan Entity Relationship Diagram (ERD). Selain itu, dirancang pula knowledge base dan decision trees sebagai dasar pengambilan keputusan oleh sistem menggunakan metode forward chaining. Perancangan antarmuka pengguna (interface design) juga dilakukan secara menyeluruh untuk mendukung pengalaman pengguna yang optimal, mencakup halaman login, dashboard, fitur konsultasi, panduan penanaman, produk, akun, hingga pengelolaan konten dan pengguna.

Tahap berikutnya adalah prototyping, di mana sistem mulai dikembangkan secara bertahap berdasarkan rancangan yang telah disusun. Pengembangan dilakukan menggunakan framework Laravel untuk sisi backend, Bootstrap untuk antarmuka frontend, serta MySQL sebagai basis data. Setiap komponen diuji secara modular sebelum digabungkan menjadi sistem yang utuh. Tahap akhir adalah testing, yang bertujuan untuk memastikan sistem berfungsi sesuai dengan spesifikasi. Pengujian dilakukan melalui dua pendekatan. Pertama, black box testing digunakan untuk mengevaluasi fungsionalitas sistem tanpa melihat struktur kode internal, dengan fokus pada kesesuaian input dan output berdasarkan skenario penggunaan. Kedua, dilakukan pengujian keakuratan sistem pakar melalui perbandingan hasil diagnosis sistem dengan analisis pakar tanaman cabai pada sepuluh studi kasus. Hasil pengujian menunjukkan bahwa sistem memiliki akurasi diagnosis sebesar 100%, yang menandakan bahwa sistem pakar yang dibangun dapat berfungsi secara efektif dan sesuai dengan tujuan penelitian.

HASIL

Fitur konsultasi penyakit merupakan fitur yang dapat dimanfaatkan oleh petani untuk mengetahui penyakit tanaman cabai mereka. Petani dapat menjawab pertanyaan yang tampil di layar dengan menekan tombol Ya (setuju) dan Tidak (tidak setuju). Jawaban dari pengguna akan diolah oleh sistem pakar berdasarkan *knowledge base* dan algoritma *forward chaining* kemudian hasilnya akan ditampilkan pada layar. Gambar 1 menunjukkan tampilan dari fitur konsultasi penyakit:



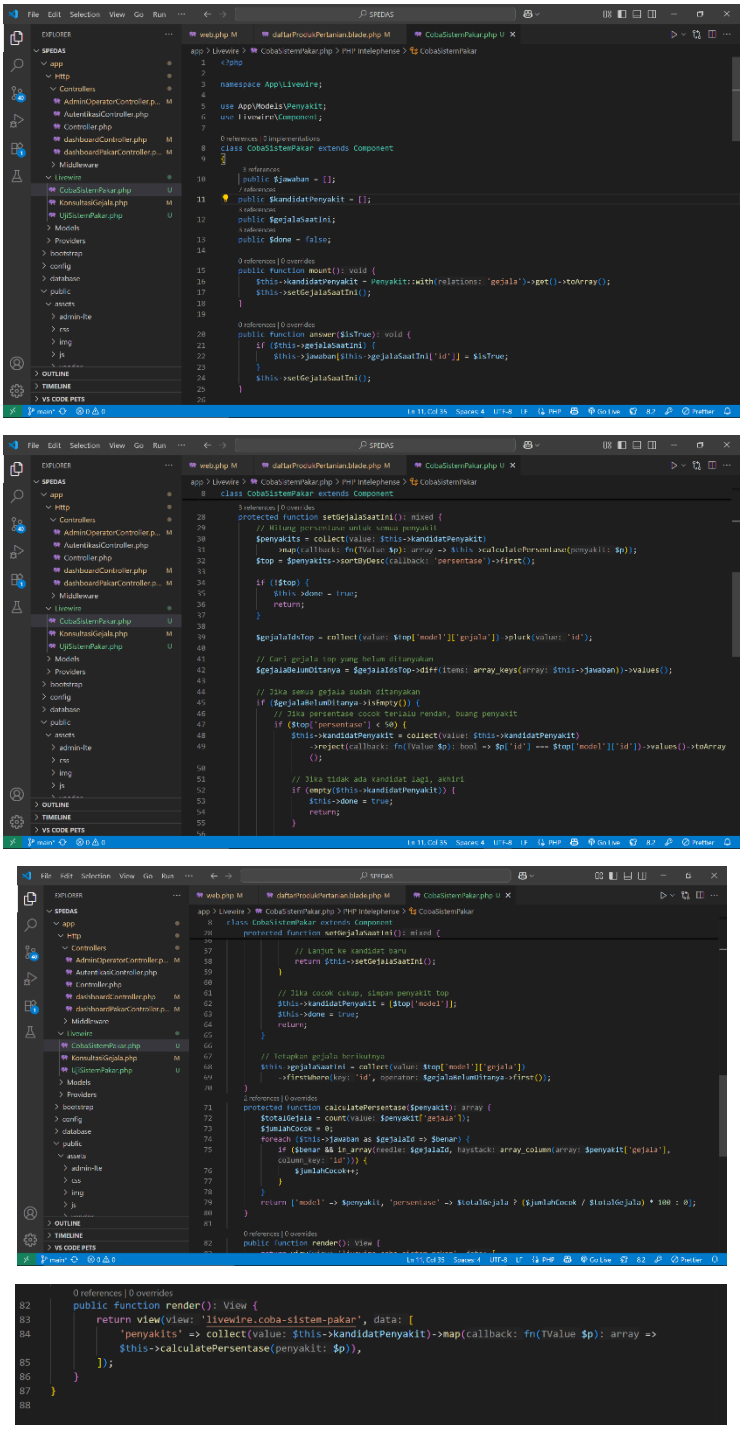
Gambar 1. Fitur Konsultasi Penyakit

Kita menggunakan *code livewire* untuk menganalisis penyakit berdasarkan gejala-gejala yang ditemui petani. Proses analisa dimulai dari *method* mount() yang mengambil seluruh data penyakit beserta relasi gejalanya dari *database*. Data tersebut kemudian disimpan dalam *array* kandidatPenyakit dan langsung memicu proses konsultasi dengan memanggil *method* setGejalaSaatIni() yang akan menentukan gejala pertama yang ditanyakan kepada pengguna.

Ketika pengguna memberikan jawaban terhadap suatu gejala, *method* answer($isTrue) akan dijalankan. Di dalamnya, sistem menyimpan jawaban pengguna dalam *array* jawaban[], dengan format id_gejala => true/false. Setelah itu, sistem kembali memanggil setGejalaSaatIni() untuk mengevaluasi jawaban tersebut dan menentukan langkah berikutnya dalam proses diagnosa. *Method* setGejalaSaatIni() adalah bagian terpenting dalam alur kerja sistem pakar ini. Fungsi ini akan menghitung tingkat kecocokan antara setiap penyakit dalam daftar kandidat terhadap jawaban petani dengan bantuan *method* calculatePersentase(). Penyakit yang memiliki tingkat kecocokan tertinggi akan dianggap sebagai kandidat utama, dan sistem kemudian mencari gejala-gejala yang belum ditanyakan dari penyakit tersebut.

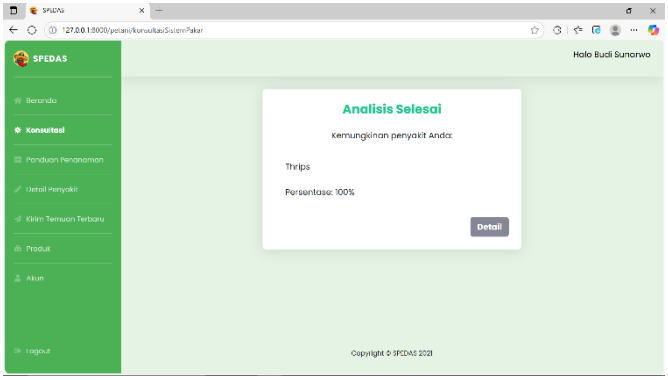
Penyakit akan dieliminasi dari daftar kandidat ketika semua gejala sudah ditanyakan dan tingkat kecocokan penyakit tersebut kurang dari 50%. Apabila tidak ada kandidat penyakit yang tersisa, maka proses analisis akan dihentikan. jika persentase kecocokan penyakit cukup tinggi (diatas 50%) maka penyakit tersebut disimpan sebagai satu-satunya kandidat dan proses dianggap selesai. Sementara itu *method* calculatePersentase() berfungsi untuk menghitung seberapa besar kecocokan antara jawaban pengguna dengan setiap gejala dari suatu penyakit. persentase tersebut dihitung dari jumlah gejala yang cocok dibagi dengan total gejala penyakit, kemudian dikalikan 100.

Terakhir, *method* render() digunakan untuk menampilkan *view* bersama dengan data penyakit yang telah diperbarui dengan nilai persentase kecocokannya. View ini akan menampilkan informasi kepada petani dan mengatur alur interaksi selama proses konsultasi berlangsung. Gambar 2 menunjukkan *code livewire* yang kita gunakan untuk melakukan analisa penyakit tanaman cabai.



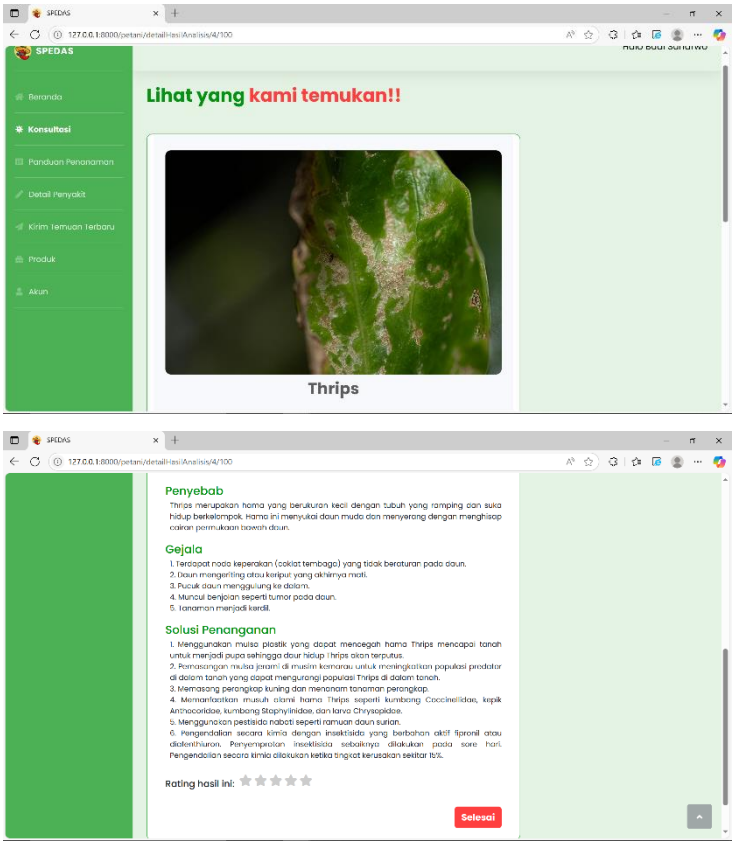
Gambar 2 Penalaran *Forward Chaining*

Penyakit akan ditampilkan setelah proses penalaran selesai. Apabila penyakit ditemukan maka nama penyakit yang cocok dan persentase kecocokan akan ditampilkan pada layar. Petani juga dapat melihat detail penyakit dengan menekan tombol detail. Seperti yang ditunjukkan pada Gambar 3.



Gambar 4 Hasil Diagnosa

Ketika tombol detail ditekan, maka petani akan diarahkan ke halaman detail diagnosa untuk melihat penyakit tanaman cabai yang menyerang tanaman mereka lengkap dengan solusi penangannya. Pada halaman ini petani diminta untuk memberikan rating untuk hasil diagnosa. Berikut gambar 5 menunjukkan tampilan detail diagnosa :



Gambar 5 Detail Diagnosa

Ketika rating diberikan dan tombol selesai ditekan, maka sistem akan menyimpan hasil diagnosa ini sebagai riwayat konsultasi petani. Gambar 63 menunjukkan code controller yang digunakan untuk menyimpan data hasil diagnosa sebagai riwayat konsultasi petani :

```
public function rating(Request $request): RedirectResponse{
    $validator = Validator::make(data: $request->all(), rules: [
        'id_penyakit' => 'required|integer|exists:penyakits,id',
        'persentase' => 'nullable|numeric',
        'rating' => 'required|integer|',
    ]);

    if($validator->fails()){
        return redirect()->back()->with(key: 'error', value: 'Gagal memberikan rating, silakan coba lagi.')
            ->withErrors(provider: $validator)
            ->withInput();
    }

    $data = $validator->validated();
    $penyakit = Penyakit::findOrFail(id: $data['id_penyakit']);
    $persentase = $data['persentase'] ?? null;
    $riwayat = new Riwayat_konsultasi();
    $riwayat->id_penyakit = $data['id_penyakit'];
    $riwayat->id_petani = Auth::guard(name: 'petani')->id();
    $riwayat->detail = "Hasil analisis penyakit menunjukkan bahwa tanaman cabai terserang penyakit ".
    $penyakit->namaPenyakit . " dengan persentase ". $persentase. "%";
    $riwayat->rating = $data['rating'];
    if(!$riwayat->save()){
        return redirect()->back()->with(key: 'error', value: 'Gagal memberikan rating, silakan coba lagi.');
    }

    return redirect()->route(route: 'konsultasi.penyakit')->with(key: 'success', value: 'Rating berhasil
    diberikan, semoga hasil analisis ini bermanfaat bagi Anda.');
}
```

Gambar 6. Code Controller Riwayat

PEMBAHASAN

Blackbox Testing

Aplikasi SPEDAS diuji menggunakan jenis pengujian *blackbox testing* dengan metode *use case testing*. Metode ini memungkinkan kita melakukan pengujian berdasarkan alur skenario penggunaan (*use case*) aplikasi seperti yang dilakukan oleh pengguna.

1. Pengujian Halaman *Landing Page*

   Proses pengujian halaman landing page dilakukan dengan mengakses setiap fitur yang terdapat dalam halaman tersebut. Tabel 1 menunjukkan pengujian pada halaman *landing page*.

Tabel 1. *Test Case Landing Page*

No	Fitur diuji	Kasus pengujian	Hasil yang diharapkan	status
1	Halaman landing page	Mengakses setiap menu yang terdapat pada navbar.	Halaman akan pindah ke section yang sesuai	Pass
		Menekan tombol coba sekarang	Dialihkan ke halaman login	Pass
		Menekan tombol Daftar	Diarahkan ke halaman register petani	Pass
		Menekan tombol Masuk	Diarahkan ke halaman login	Pass

2. Pengujian Halaman *Login*

   Proses pengujian halaman *login* memiliki beberapa skenario seperti yang ditunjukkan pada Tabel 2.

Tabel 2 *Test Case Login*

No	Fitur Diuji	Kasus pengujian	Hasil yang diharapkan	status
1	Halaman login	Menginputkan email dan password benar	Login berhasil dan dialihkan ke halaman dashboard	pass
		Menginputkan email dan password salah	Login gagal, dan muncul pesan gagal	pass

3. Pengujian Dashboard Petani

Pengujian dashboard petani dilakukan dengan menguji setiap menu yang dapat diakses oleh petani pada sidebar. Seperti beranda, konsultasi, panduan penanaman, detail penyakit, kirim temuan terbaru, produk dan akun. Tabel 3 menunjukkan pengujian dashboard petani.

Tabel 3. *Test Case Dashboard* Petani

No	Fitur Diuji	Kasus pengujian	Hasil yang diharapkan	status
1	Beranda	Apakah Semua data dashboard ditampilkan dengan baik?	Semua data dashboard ditampilkan dengan akurat	pass
		Hapus riwayat konsultasi	Petani berhasil menghapus data riwayat konsultasi	pass
2	Konsultasi	Menjawab setiap pertanyaan berdasarkan gejala yang ditemui	Sistem pakar berhasil mendeteksi penyakit berdasarkan gejala yang dimasukkan	pass
		Menjawab tidak untuk semua gejala yang ditanyakan	Muncul pesan bahwa tidak ditemukan penyakit yang cocok	pass
		Menekan tombol detail untuk melihat detail diagnosa penyakit	Dialihkan ke halaman detail diagnosa	pass
		Memberi rating diagnosa	Berhasil memberikan rating diagnosa	pass
3	Panduan Penanaman	Apakah data panduan penanaman muncul ketika menu panduan penanaman ditekan?	Terdapat daftar data panduan yang bisa diakses petani	pass
		Menekan tombol detail	Halaman dialihkan ke detail panduan penanaman yang sesuai	pass
4	Detail Penyakit	Apakah semua penyakit muncul ketika menu detail penyakit ditekan?	Semua penyakit yang tersimpan dalam database	pass

No	Fitur Diuji	Kasus pengujian	Hasil yang diharapkan	status
			muncul pada halaman detail penyakit	
		Menekan card penyakit untuk melihat detail penyakit	Halaman dialihkan ke lihat detail penyakit yang sesuai	pass
5	Kirim Temuan Terbaru	Mengirimkan temuan terbaru pada form yang ada	Berhasil mengirimkan temuan terbaru	pass
6	Produk	Mengakses menu produk	Semua data produk yang terdapat dalam aplikasi SPEDAS ditampilkan	pass
		Menekan card produk yang sesuai untuk melihat detail produk	Berhasil masuk ke halaman detail produk yang sesuai	pass
7	Akun	Mengakses menu akun	Berhasil masuk ke halaman akun, dan data yang ditampilkan sesuai	pass
		Menekan tombol ubah	Halaman dialihkan ke form ubah akun	pass
		Mengisi form ubah akun, mengisi form ubah akun, dan menekan tombol simpan	Berhasil mengubah data akun berdasarkan form yang diisi ketika tombol simpan ditekan	pass

Berdasarkan hasil pengujian yang dapat dilihat pada Tabel 47, semua fitur yang ada pada dashboard petani berjalan dengan baik tanpa *error* dan *bug*.

4. Pengujian Dashboard Pakar

Pengujian *dashboard* pakar dilakukan terhadap semua menu yang terdapat pada *dashboard* pakar seperti beranda, uji sistem pakar, *knowledge base*, *approve* konten, temuan petani, dan akun. Tabel 4 menunjukkan pengujian *dashboard* pakar.

Tabel 4 *Test Case Dashboard* Pakar

No	Fitur Diuji	Kasus Pengujian	Hasil yang diharapkan	Status
1	Beranda	Apakah semua data beranda ditampilkan cocok dengan database?	Data ditampilkan sesuai dengan data yang terdapat di database	pass
2	Uji Sistem Pakar	Menjawab pertanyaan berdasarkan gejala yang ditemui	Sistem pakar berhasil mendianosa penyakit sesuai dengan gejala yang diberikan	pass
		Menekan tombol detail untuk melihat detail diagnosa penyakit	Berhasil masuk ke halaman detail diagnosa	pass

No	Fitur Diuji	Kasus Pengujian	Hasil yang diharapkan	Status
3	Knowledge base	Melihat detail penyakit dan gejala terkait	Sistem menampilkan detail penyakit lengkap dengan gejala dan penanggulangan terkait	pass
		Menekan tombol update untuk mengubah data penyakit	Berhasil masuk ke halaman ubah data penyakit, dan bisa melakukan perubahan data penyakit.	pass
		Menekan tombol tambah untuk menambahkan data penyakit baru	Dialihkan ke halaman tambah penyakit	pass
		Menekan tombol simpan ketika semua form berhasil diisi	Berhasil menyimpan data penyakit baru	pass
4	Approve Konten	Memilih konten untuk disetujui	Berhasil menyetujui isi konten (panduan / produk)	pass
		Menolak konten (panduan/produk)	Berhasil menolak konten dengan menambahkan komentar penolakan	pass
5	Temuan Petani	Menekan menu temuan petani pada sidebar	Sistem menampilkan semua data temuan petani	pass
		Menekan tombol detail untuk melihat detail temuan	Sistem menampilkan popup yang berisi data temuan petani	pass
		Menekan tombol hapus untuk menghapus data temuan petani	Berhasil menghapus data temuan petani	pass
6	Akun	Menekan menu akun yang terdapat pada sidebar	Sistem menampilkan data akun yang terdapat pada aplikasi SPEDAS	pass
		Menekan tombol ubah untuk mengubah data akun	Sistem akan mengarahkan pakar ke form ubah akun, pakar dapat mengisi data yang sesuai kemudian menekan tombol simpan untuk menyimpan perubahan.	pass

Berdasarkan pengujian yang ditunjukkan pada Tabel 48 dapat dilihat bahwa semua fitur pada dashboard pakar berjalan dengan baik tanpa error dan bug.

5. Pengujian *Dashboard* Admin/Operator

Tabel 5 menunjukkan bagaimana proses pengujian *dashboard* admin/operator. Pengujian dilakukan terhadap semua menu yang terdapat pada sidebar seperti beranda, uji sistem pakar, tambah panduan, tambah produk, laporan penolakan konten, akun, dan data pengguna.

Tabel 5 Pengujian *Dashboard* Admin/Operator

No	Fitur Diuji	Kasus Pengujian	Hasil yang diharapkan	Status
1	Beranda	Apakah semua data pada beranda ditampilkan secara akurat?	Semua data pada beranda ditampilkan secara akurat sesuai dengan database	pass
2	Coba sistem pakar	Menjawab pertanyaan terkait gejala yang ditemui dan melihat hasil analisa dari aplikasi	Aplikasi berhasil mendiagnosa penyakit berdasarkan gejala yang ditanyakan	pass
3	Tambah panduan	Menambahkan data panduan baru	Berhasil menambahkan data panduan baru	pass
		Mengubah data panduan dengan menekan tombol update	Berhasil mengubah data panduan dengan mengisi form ubah dan menekan tombol simpan	pass
		Menghapus data panduan	Berhasil menghapus data panduan dengan menekan tombol hapus	pass
4	Tambah Produk	Menambahkan produk baru	Berhasil menambahkan data produk baru dengan mengisi form tambah produk dan menekan tombol simpan	pass
		Mengubah data produk dengan menekan tombol update	Berhasil mengubah data produk dengan mengisi form ubah dan menekan tombol simpan	pass
		Menghapus data produk	Berhasil menghapus data produk dengan menekan tombol hapus	pass
5	Laporan Penolakan Konten	Menghapus konten yang ditolak	Berhasil menghapus konten yang ditolak dengan menekan tombol hapus	pass
		Mengubah konten berdasarkan komentar penolakan konten yang diberikan oleh pakar	Berhasil mengubah data panduan berdasarkan komentar yang diberikan pakar	pass
6	Akun	Menekan menu akun pada sidebar untuk melihat data akun	Berhasil masuk ke menu akun, dan data yang ditampilkan sesuai	pass
		Mengubah data akun dengan menekan tombol ubah	Berhasil mengubah data akun dengan mengisi form ubah dan menekan tombol simpan	pass
7	Data Pengguna	Menghapus atau mengubah data petani	Berhasil menghapus atau mengubah data petani	pass
		Menambahkan, mengubah atau menghapus data pakar	Berhasil menambahkan, mengubah, atau menghapus data pakar	pass

No	Fitur Diuji	Kasus Pengujian	Hasil yang diharapkan	Status
		Menambahkan, mengubah atau menghapus data operator	Berhasil menambahkan, mengubah atau menghapus data operator	pass

Berdasarkan pengujian yang ditunjukkan pada Tabel 49 dapat dilihat bahwa semua fitur pada dashboard admin/operator berjalan dengan baik tanpa error dan bug.

Uji Akurasi oleh Pakar

Untuk memastikan proses diagnosa penyakit pada aplikasi SPEDAS benar-benar akurat, aplikasi SPEDAS akan diuji oleh pakar tanaman cabai yaitu bapak Ichsan Kurniawan, SP. Proses pengujian dilakukan dengan meminta pakar untuk memberikan soal berupa daftar gejala, kemudian aplikasi akan mendiagnosa penyakit berdasarkan gejala tersebut. Hasil diagnosa akan diperiksa oleh pakar dan menentukan apakah jawaban benar atau salah.

Tabel 6 menunjukkan hasil uji akurasi aplikasi SPEDAS dengan pakar tanaman cabai yaitu bapak Ichsan Kurniawan, SP.

Tabel 6 Uji Akurasi

Nomor Uji	Hasil Uji		
	Penyakit	Persentase	Benar/Salah
1	Thrips	60%	Benar
2	Kutu Daun Persik	67%	Benar
3	Tungau Kuning	60%	Benar
4	Lalat Buah	100%	Benar
5	Ulat Penggerek Buah	100%	Benar
6	Layu Bakteri	80%	Benar
7	Layu Fusarium	75%	Benar
8	Busuk Buah Antraknosa	60%	Benar
9	Virus Kuning	100%	Benar
10	Bercak Daun Cercospora	100%	Benar

Berdasarkan Tabel 6 didapat hasil sebagai berikut :

a. Jumlah jawaban benar : 10
b. Jumlah jawaban salah : 0
c. Persentase Akurasi

P = Jumlah jawaban benar / total soal * 100

P = 10 / 10 * 100

P = 100

Persentase akurasi = 100%

Berdasarkan hasil uji akurasi yang dilakukan oleh pakar tanaman cabai yaitu bapak Ichsan Kurniawan, SP dari BPP Kecamatan IV Koto menggunakan 10 buah soal, didapat kesimpulan bahwa aplikasi analisis penyakit tanaman cabai menggunakan metode forward chaining memiliki tingkat akurasi tinggi mencapai 100%. Hasil uji akurasi ini juga menunjukkan bahwa algoritma forward chaining bekerja dengan baik dan dapat digunakan pada aplikasi analisis penyakit tanaman cabai.

KESIMPULAN

Berdasarkan hasil pengujian, sistem pakar untuk analisis penyakit tanaman cabai yang dibangun menggunakan metode *forward chaining* menunjukkan performa yang sangat baik. Hal ini ditunjukkan melalui hasil uji akurasi yang mencapai 100%. Pengujian dilakukan oleh pakar tanaman cabai, Bapak Ichsan Kurniawan, S.P., menggunakan 10 soal studi kasus. Sistem berhasil memberikan diagnosa yang benar pada seluruh soal tersebut. Hasil ini menunjukkan bahwa sistem mampu menganalisis dan mendiagnosis penyakit tanaman cabai secara tepat berdasarkan gejala yang diberikan. Dengan demikian, sistem ini dapat menjadi alat bantu yang efektif bagi petani atau pihak terkait dalam mengidentifikasi penyakit tanaman cabai lebih cepat dan akurat.

Secara keseluruhan, aplikasi berjalan dengan baik, ditunjukkan melalui hasil pengujian *blackbox testing*. Seluruh fitur dalam aplikasi telah diuji menggunakan berbagai skenario yang merepresentasikan kemungkinan interaksi pengguna, dan semuanya berfungsi sesuai dengan yang diharapkan. Hasil ini menunjukkan bahwa aplikasi telah memenuhi standar fungsionalitas serta dapat digunakan secara efektif oleh pengguna sesuai dengan tujuan yang telah dirancang.

DAFTAR PUSTAKA

Aini, N., Ramadiani, R., & Hatta, H. R. (2017). Sistem Pakar Pendiagnosa Penyakit Tuberkulosis. *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, *12*(1), 56. https://doi.org/10.30872/jim.v12i1.224

Akli, I. (2017). Analisa Efektifitas Metode Forward Chaining Dan Backward Chaining Pada Sistem Pakar. *Jurnal Pilar Nusa Mandiri*, *13*(1), 35.

Ardiansyah, I., Efatmi, F., Mardawati, E., Putri, S. H., & Pujianto, T. (2022). *Sistem Pakar Uji Kelayakan Perizinan Pangan Industri Rumah Tangga (P-IRT)* (A. Kamsyach (ed.); 1st ed.).

Fatah, D. A., Rifqi, K., Sawaki, S., & Irhamni, F. (2023). Rancang Bangun Sistem Pakar

Diagnosa Hama Dan Penyakit Tanaman Bawang Merah Dan Cabai Mengunakan Metode Forward Chaining. *Networking Engineering Research Operation*, *8*(2), 157–168. https://doi.org/10.21107/nero.v8i2.26789

Irawan, J. (2007). *Sistem Pakar*.

Juniawan, F. P. (2017). Penggunaan Metode Forward Chaining Dalam Perancangan Sistem Pakar Diagnosa Gangguan Kejiwaan. *Jurnal Ilmiah Informatika Global*, *8*(1), 29–35. https://doi.org/10.36982/jiig.v8i1.220

Kuswanto, J. (2022). *Sistem Pakar : Penyelesaian Kasus Kerusakan Pada Mesin Cuci dengan Metode Backward Chaining* (D. Fadhila (ed.); 1st ed.). CV.Mitra Cendikia Media.

Piay, S. S., Tyasdjaja, A., Ermawati, Y., & Hantoro, F. R. P. (2010). *Budidaya dan Pascapanen Cabai Merah* (Vol. 1).

Sapriadi, S., Hayati, N., Eko Syaputra, A., Septi Eirlangga, Y., Manurung, K. H., & Hayati, N. (2023). Sistem Pakar Diagnosa Gaya Belajar Mahasiswa Menggunakan Metode Forward Chaining. *Jurnal Informasi Dan Teknologi*, *5*(3), 71–78. https://doi.org/10.60083/jidt.v5i3.381

Suryadi, D., Meilianda, R., Suryono, A. F., & Munadi, M. (2018). Sistem Pakar untuk Mengidentifikasi Kerusakan Mesin Industri Menggunakan Metode Certainty Factor. *Rotasi*, *20*(1), 56. https://doi.org/10.14710/rotasi.20.1.56-62