



Contents lists available at [JESA](https://jesa.aks.or.id/index.php/jesa/index)

Journal of Engineering and Science Application

journal homepage: <https://jesa.aks.or.id/index.php/jesa/index>



Sistem Pakar Diagnosa Penyakit dan Hama Tanaman Terong Berbasis Website untuk Petani Pemula

Muhammad Al Rifzki^a, Jaka Sutresna^b

^aProgram Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Pamulang, Tangerang Selatan 15310, Indonesia

^bProgram Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Pamulang, Tangerang Selatan 15310, Indonesia

Article Info

Keywords:

Expert System
Dempster-Shafer
Eggplant
Disease and Pest Diagnosis
Website
Waterfall

ABSTRACT

Novice farmers often have difficulty identifying diseases and pests of eggplant because of limited knowledge and restricted access to agricultural experts, which delays treatment and increases the risk of crop failure. This research aims to design and develop a web-based expert system for diagnosing eggplant diseases and pests using the Dempster-Shafer method to handle symptom uncertainty. Data were collected through observation, expert interviews, and literature review, resulting in 10 types of diseases and pests, 25 symptoms, rules, and belief values for each symptom. The system was developed using the Waterfall method with PHP and MySQL and modeled using the Unified Modeling Language (UML). In the diagnosis process, users select the observed symptoms and the system combines their belief values using Dempster's rule of combination to display the disease or pest with the highest confidence level and its treatment. In the calculation example with three symptoms, Bacterial Wilt obtained the highest belief value of 0.924. Black Box Testing achieved a 100% success rate from 28 test scenarios, while White Box Testing produced cyclomatic complexity values between 1 and 10 across 11 modules. A questionnaire completed by 13 respondents gave a feasibility score of 90%, indicating that the system is feasible.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Muhammad Al Rifzki
Program Studi Teknik Informatika
Universitas Pamulang
Jalan Raya Puspipetek Nomor 46, Buaran, Pamulang, Tangerang Selatan, Banten, 15310
Email: malrifzki03@gmail.com

1. INTRODUCTION

Tanaman terong merupakan salah satu komoditas hortikultura yang banyak dibudidayakan dan dikonsumsi rumah tangga sebagai bahan sayur [1], [2]. Dalam budidayanya, tanaman terong rentan terserang berbagai penyakit dan hama yang merusak daun, batang, hingga buah sehingga tanaman mati dan petani mengalami gagal panen [3]. Gangguan seperti busuk buah, thrips, tungau, dan ulat grayak telah banyak dilaporkan menyerang tanaman terong [4], [5], [6], [7], sehingga pengendaliannya memerlukan pengetahuan yang memadai tentang gejala serangan [8], [9].

Petani pemula sering mengalami kesulitan dalam mengenali jenis penyakit maupun hama yang menyerang tanaman karena pengalaman yang masih terbatas dan sulitnya akses terhadap tenaga ahli pertanian,

terutama di wilayah pedesaan. Proses diagnosa yang berjalan saat ini masih dilakukan secara konvensional: petani mencatat gejala dan menyampaikannya kepada pakar, kemudian pakar melakukan wawancara dan observasi langsung ke lahan sebelum menganalisisnya secara manual. Proses tersebut memerlukan waktu dan biaya, bergantung pada ketersediaan pakar, serta hasilnya kurang konsisten ketika gejala yang muncul saling mirip, sehingga penanganan sering terlambat.

Sistem pakar merupakan sistem yang mengadopsi pengetahuan dan keahlian manusia ke dalam komputer sehingga mampu menyelesaikan masalah layaknya seorang pakar [10], [11]. Gejala yang teramati di lapangan sering tidak lengkap atau tidak spesifik terhadap satu jenis gangguan, sehingga diperlukan metode yang mampu menangani ketidakpastian. Metode Dempster-Shafer menggabungkan beberapa bukti (*evidence*) berdasarkan derajat keyakinan masing-masing bukti sehingga dapat digunakan pada penalaran dengan informasi yang tidak pasti [12].

Beberapa penelitian telah menerapkan metode Dempster-Shafer pada sistem pakar. Diagnosa penyakit tanaman kakao berbasis web mencapai akurasi 90% [13], diagnosa penyakit ginjal berhasil mendiagnosa 29 dari 31 kasus (93,54%) [14], diagnosa penyakit kopi robusta memperoleh akurasi 94% dari 13 kasus uji [15], dan diagnosa penyakit jambu kristal memperoleh akurasi rata-rata 83,33% [16]. Metode yang sama juga digunakan pada diagnosa ensefalitis [17] dan penyakit tanaman nanas [18]. Pada tanaman terong belanda, penelitian [19] hanya berfokus pada penyakit tanpa mencakup hama dan belum mengakomodasi kebutuhan petani pemula dalam menghadapi ketidakpastian input gejala.

Berdasarkan permasalahan tersebut, penelitian ini membangun sistem pakar diagnosa penyakit dan hama tanaman terong berbasis website dengan metode Dempster-Shafer. Sistem mencakup 10 jenis penyakit dan hama dengan 25 gejala, pengguna memilih minimal dua gejala yang terlihat pada tanaman, dan hasil diagnosa ditampilkan berupa nama penyakit atau hama, tingkat kepercayaan, serta solusi penanganannya. Kontribusi penelitian ini terletak pada pencakupan penyakit sekaligus hama tanaman terong dalam satu sistem yang ditujukan bagi petani pemula, dengan nilai keyakinan (*belief*) yang ditetapkan berdasarkan pengetahuan pakar, penyimpanan riwayat diagnosa, serta pengelolaan basis pengetahuan oleh admin.

2. RESEARCH METHOD

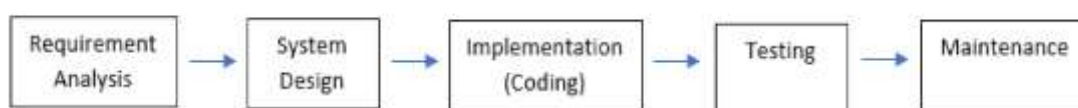
Penelitian ini menggunakan pendekatan pengembangan sistem pakar berbasis website yang memadukan akuisisi pengetahuan pakar dengan metode Dempster-Shafer. Tahapan penelitian meliputi pengumpulan data, pengembangan sistem menggunakan metode *Waterfall*, penerapan perhitungan Dempster-Shafer, serta pengujian terhadap sistem yang telah dikembangkan [20].

2.1 Data Collection Method

Pengumpulan data dilakukan melalui observasi, wawancara, dan studi pustaka. Observasi dilakukan secara langsung di perkebunan budidaya terong untuk memahami kondisi lapangan serta permasalahan yang dihadapi petani. Wawancara dilakukan dengan narasumber pakar pertanian untuk memperoleh data penyakit dan hama, gejala, aturan (*rules*), serta tingkat keyakinan pakar terhadap keterkaitan setiap gejala dengan penyakit atau hama. Studi pustaka dilakukan dengan mengkaji buku, jurnal, dan artikel ilmiah yang berkaitan dengan sistem pakar, metode Dempster-Shafer, serta penyakit dan hama tanaman terong. Hasil pengumpulan data berupa 10 jenis penyakit dan hama, 25 gejala, aturan yang menghubungkan gejala dengan penyakit atau hama, serta nilai *belief* pada setiap gejala.

2.2 System Development Method

Metode pengembangan sistem yang digunakan adalah *System Development Life Cycle* (SDLC) model *Waterfall* yang dikerjakan secara berurutan, sehingga tahap berikutnya baru dikerjakan setelah tahap sebelumnya selesai [20], [21]. Tahapan yang digunakan terdiri dari *Requirement Analysis*, *System Design*, *Implementation*, *Testing*, dan *Maintenance* seperti ditunjukkan pada Gambar 1. Pada tahap *Requirement Analysis*, kebutuhan sistem ditentukan berdasarkan proses diagnosa yang berjalan. Tahap *System Design* dilakukan dengan merancang basis data (normalisasi, *Entity Relationship Diagram*, dan *Logical Record Structure*), pemodelan UML berupa *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram* [22], [23], [24], serta rancangan antarmuka. Tahap *Implementation* dilakukan dengan bahasa pemrograman PHP dan basis data MySQL [25], [26] menggunakan *Visual Studio Code* [27], sedangkan tahap *Testing* dilakukan menggunakan *Black Box Testing*, *White Box Testing*, dan *User Response*.



Gambar 1. Tahapan metode Waterfall

2.3 Dempster-Shafer Method

Metode Dempster-Shafer digunakan untuk menggabungkan bukti dari beberapa gejala yang dipilih pengguna dalam kondisi ketidakpastian [12]. Himpunan seluruh kemungkinan penyakit dan hama disebut *frame of discernment* (θ), yaitu P1 sampai P10. Setiap gejala memiliki nilai densitas atau *belief* $m\{X\}$ terhadap himpunan penyakit atau hama X yang didukungnya, dan sisa keyakinan yang tidak terdistribusi diberikan kepada $\emptyset$ sesuai persamaan (1). Nilai *belief* ditetapkan pakar secara kualitatif (sangat rendah, rendah, sedang, tinggi, sangat tinggi), kemudian dikonversi ke rentang numerik 0–1: 0–0,2 sangat rendah, 0,2–0,4 rendah, 0,4–0,6 sedang, 0,6–0,8 tinggi, dan 0,8–1 sangat tinggi.

$$m\{\emptyset\} = 1 - m\{X\} \quad (1)$$

Keterangan:

$m\{X\}$ = nilai *belief* gejala terhadap himpunan penyakit atau hama X;

$m\{\emptyset\}$ = nilai ketidakpastian (keyakinan yang tidak terdistribusi).

Apabila terdapat dua sumber bukti, nilai densitas gabungan dihitung menggunakan aturan kombinasi Dempster pada persamaan (2). Hasil kombinasi kemudian dikombinasikan kembali dengan bukti berikutnya hingga seluruh gejala terpilih diproses. Himpunan penyakit atau hama dengan nilai densitas tertinggi ditetapkan sebagai hasil diagnosa.

$$m_3(Z) = \frac{\sum_{X \cap Y = Z} m_1(X) \cdot m_2(Y)}{1 - \sum_{X \cap Y = \emptyset} m_1(X) \cdot m_2(Y)} \quad (2)$$

Keterangan:

$m_3(Z)$ = densitas gabungan untuk himpunan Z;

$m_1(X)$, $m_2(Y)$ = densitas bukti pertama dan kedua;

$X \cap Y$ = irisan himpunan penyakit atau hama yang didukung kedua bukti;

penyebut = 1 dikurangi nilai konflik, yaitu jumlah perkalian densitas dengan irisan kosong ($\emptyset$).

2.4 System Testing

Pengujian sistem dilakukan menggunakan *Black Box Testing*, *White Box Testing*, dan *User Response*. *Black Box Testing* menguji fungsi sistem berdasarkan masukan dan keluaran tanpa melihat struktur internal kode program [28], [29]. *White Box Testing* menganalisis struktur dan alur logika program melalui *flowgraph* serta perhitungan *Cyclomatic Complexity* untuk menentukan jalur independen yang perlu diuji [29]. *User Response* dilakukan menggunakan kuesioner melalui *Google Form* yang terdiri dari 8 pertanyaan dan diisi oleh 13 responden dengan pilihan jawaban Baik (bobot 3), Cukup (bobot 2), dan Kurang Baik (bobot 1).

3. RESULTS AND DISCUSSION

Bagian ini menyajikan hasil perancangan sistem, basis pengetahuan dan perhitungan metode Dempster-Shafer, implementasi antarmuka, serta hasil pengujian untuk mengetahui kesesuaian fungsi dan penerimaan pengguna terhadap sistem pakar yang dikembangkan.

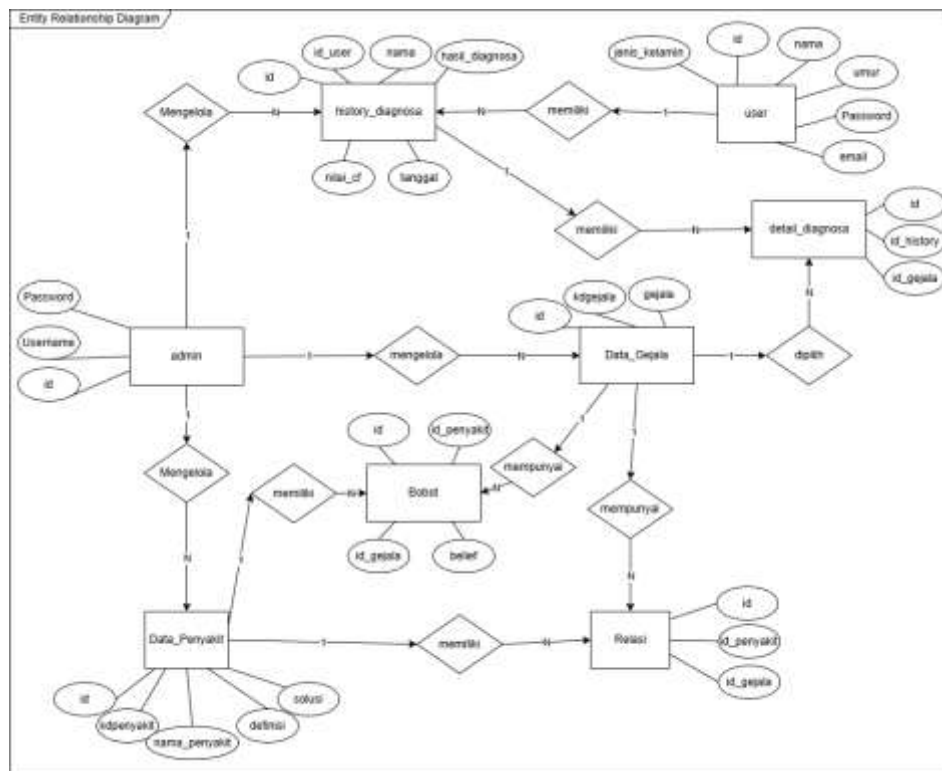
3.1 System Design Results

Analisis sistem berjalan menunjukkan bahwa diagnosa masih bergantung pada kehadiran pakar di lapangan sehingga lambat dan kurang konsisten. Sistem yang diusulkan memungkinkan pengguna memilih gejala melalui antarmuka website, kemudian sistem memproses *belief* setiap gejala dengan aturan kombinasi Dempster dan menampilkan penyakit atau hama dengan kepercayaan tertinggi beserta solusi penanganannya. Pemodelan kebutuhan dan interaksi sistem menggunakan UML [22], sedangkan hubungan antarentitas pada basis data digambarkan menggunakan ERD.



Gambar 2. Use Case Diagram sistem pakar

Use Case Diagram menggambarkan interaksi dua aktor dengan sistem. User dapat melakukan pendaftaran, *login*, konsultasi diagnosa, melihat daftar penyakit dan hama, serta *logout*. Admin dapat melakukan *login*, mengelola data penyakit, gejala, relasi, bobot, dan laporan riwayat diagnosa, serta *logout*.



Gambar 3. Entity Relationship Diagram sistem pakar

ERD pada Gambar 3 terdiri atas delapan entitas, yaitu admin, user, data penyakit, data gejala, relasi, bobot, *history* diagnosa, dan detail diagnosa. Entitas relasi menghubungkan penyakit dengan gejala sebagai basis aturan, entitas bobot menyimpan nilai *belief* setiap gejala terhadap penyakit, sedangkan *history* dan detail diagnosa menyimpan hasil konsultasi pengguna beserta gejala yang dipilih. Struktur data dinormalisasi hingga bentuk normal ketiga (3NF) untuk meminimalkan pengulangan data dan menjaga konsistensi.

3.2 Dempster-Shafer Calculation Results

3.2.1 Knowledge Base

Basis pengetahuan hasil akuisisi dari pakar terdiri atas 10 penyakit dan hama (Tabel 1), 25 gejala beserta nilai *belief*-nya (Tabel 2), dan 34 relasi gejala-penyakit. Penyakit terdiri dari layu bakteri, busuk buah, bercak daun, antraknosa, dan rebah semai, sedangkan hama terdiri dari kutu daun, ulat grayak, tungau, lalat buah, dan *thrips*. Nilai *belief* berkisar antara 0,4 sampai 0,9. Gejala dengan nilai tertinggi (0,9) merupakan ciri khas suatu penyakit atau hama, misalnya tanaman layu (G2), akar menghitam atau membusuk (G12), terdapat kutu pada daun (G13), terdapat ulat pada daun (G16), dan terdapat larva pada buah (G21). Sebaliknya, gejala daun menguning (G1) hanya bernilai 0,4 karena muncul pada tiga penyakit atau hama yang berbeda (P1, P6, dan P8).

Tabel 1. Data penyakit, hama, dan gejala penyusun aturan

Kode	Nama Penyakit/Hama	Jenis	Kode Gejala
P1	Layu Bakteri	Penyakit	G1, G2, G3, G6, G7
P2	Busuk Buah	Penyakit	G17, G18, G19
P3	Bercak Daun	Penyakit	G6, G8, G9
P4	Antraknosa	Penyakit	G19, G24, G25
P5	Rebah Semai	Penyakit	G3, G7, G12
P6	Kutu Daun	Hama	G1, G4, G5, G13, G14
P7	Ulat Grayak	Hama	G15, G16
P8	Tungau	Hama	G1, G9, G10, G11, G22
P9	Lalat Buah	Hama	G20, G21
P10	<i>Thrips</i>	Hama	G4, G22, G23

Tabel 2. Data gejala dan nilai belief

Kode	Gejala	Belief	Kode	Gejala	Belief
G1	Daun menguning	0,4	G14	Daun lengket	0,8
G2	Tanaman layu	0,9	G15	Daun berlubang	0,8
G3	Pertumbuhan tanaman terhambat	0,6	G16	Terdapat ulat pada daun	0,9
G4	Daun keriting	0,6	G17	Buah membusuk	0,8
G5	Daun menggulung	0,8	G18	Buah berair	0,8
G6	Daun rontok	0,6	G19	Buah terdapat bercak hitam	0,6
G7	Batang menghitam	0,6	G20	Buah berlubang	0,8
G8	Bercak coklat pada daun	0,8	G21	Terdapat larva pada buah	0,9
G9	Bercak putih pada daun	0,6	G22	Daun keperakan	0,6
G10	Daun terlihat kusam	0,8	G23	Bunga atau pucuk rusak	0,8
G11	Permukaan daun mengering	0,8	G24	Buah mengering	0,8
G12	Akar menghitam/membusuk	0,9	G25	Buah keriput	0,8
G13	Terdapat kutu pada daun	0,9			

3.2.2 Calculation Example

Sebagai contoh, pengguna memilih tiga gejala, yaitu daun menguning (G1), tanaman layu (G2), dan batang menghitam (G7). Nilai densitas awal setiap gejala adalah $m_1\{P1, P6, P8\} = 0,4$ dan $m_1\{\emptyset\} = 0,6$ untuk G1; $m_2\{P1\} = 0,9$ dan $m_2\{\emptyset\} = 0,1$ untuk G2; serta $m_4\{P1, P5\} = 0,6$ dan $m_4\{\emptyset\} = 0,4$ untuk G7. Kombinasi G1 dan G2 menghasilkan $m_3\{P1\} = 0,36 + 0,54 = 0,9$, $m_3\{P1, P6, P8\} = 0,04$, dan $m_3\{\emptyset\} = 0,06$. Karena tidak terdapat irisan kosong, nilai konflik sama dengan 0 sehingga penyebut persamaan (2) bernilai 1. Kombinasi m_3 dengan G7 menghasilkan $m_5\{P1\} = 0,54 + 0,36 + 0,024 = 0,924$, $m_5\{P1, P5\} = 0,036$, $m_5\{P1, P6, P8\} = 0,016$, dan $m_5\{\emptyset\} = 0,024$, seperti ditunjukkan pada Tabel 3.

Tabel 3. Hasil kombinasi evidence G1, G2, dan G7

Himpunan	Setelah G1 + G2 (m_3)	Setelah G1 + G2 + G7 (m_5)
$\{P1\}$	0,900	0,924
$\{P1, P5\}$	—	0,036
$\{P1, P6, P8\}$	0,040	0,016
$\emptyset$	0,060	0,024
Total	1,000	1,000

Nilai densitas tertinggi adalah 0,924 pada himpunan $\{P1\}$, sehingga sistem menyimpulkan tanaman terserang Layu Bakteri dengan tingkat kepercayaan 92,4%. Penambahan gejala G7 meningkatkan nilai P1 dari 0,9 menjadi 0,924 karena G7 mendukung P1, sedangkan P5 hanya memperoleh 0,036 karena hanya didukung oleh satu gejala. Pada tahap awal, dengan hanya dua gejala (G1 dan G2), sistem sudah menghasilkan P1 sebagai kandidat terkuat dengan nilai 0,9. Hal ini menunjukkan bahwa metode Dempster-Shafer tetap dapat memberikan diagnosa meskipun gejala yang dipilih belum lengkap, dengan sisa ketidakpastian yang terukur melalui nilai $m\{\emptyset\}$.

3.3 System Implementation

Sistem diimplementasikan menggunakan PHP dan MySQL pada sistem operasi Windows 11 dengan Visual Studio Code sebagai editor, Laragon sebagai server, serta Enterprise Architect 15.1 dan draw.io untuk perancangan. Pengembangan dilakukan pada laptop HP x360 14DY dengan prosesor Intel Core i5-1135G7 2,40 GHz, RAM 16 GB, dan penyimpanan 512 GB.

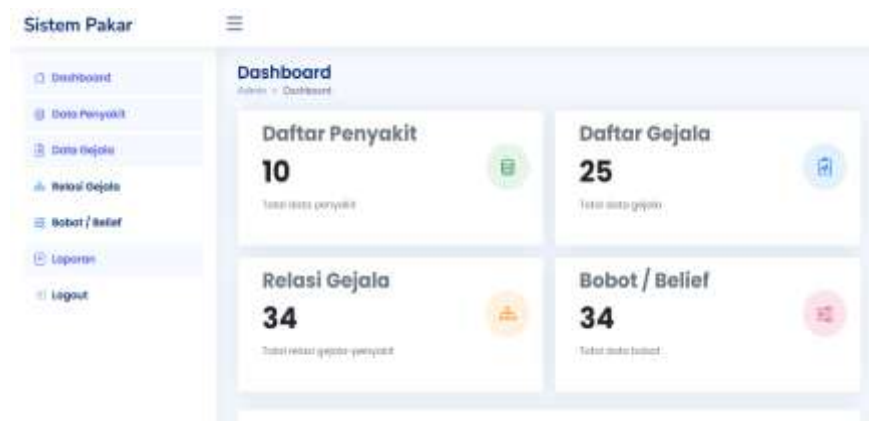


Gambar 4. Tampilan halaman Home

Gambar 4 menunjukkan halaman *Home* yang menjadi tampilan awal sistem, memuat judul, deskripsi singkat, serta tombol “Mulai Konsultasi” dan “Lihat Daftar Penyakit” untuk mengakses fitur utama.

Gambar 5. Tampilan halaman Konsultasi

Pada halaman Konsultasi (Gambar 5), pengguna memilih gejala yang terlihat pada tanaman melalui *checkbox*, minimal dua gejala, lalu menekan tombol “Diagnosa” untuk memproses perhitungan Dempster-Shafer. Pengguna yang sudah *login* memperoleh riwayat diagnosa yang tersimpan secara otomatis, sedangkan pengguna tamu tetap dapat berkonsultasi tanpa penyimpanan riwayat.



Gambar 6. Tampilan halaman Dashboard admin

Gambar 6 menampilkan *dashboard* admin yang memuat ringkasan data, yaitu 10 penyakit dan hama, 25 gejala, 34 relasi gejala-penyakit, dan 34 data bobot.

No	Kode Gejala	Nama Gejala	Nilai Belief (0-1)	Hapus
1	01	Daun menguning	0.40	[X]
2	02	Tanaman layu	0.90	[X]
3	03	Pertumbuhan tanaman terhambat	0.80	[X]

Gambar 7. Tampilan halaman Bobot/Belief

Halaman Bobot/Belief (Gambar 7) digunakan admin untuk mengatur nilai *belief* 0–1 pada setiap gejala terhadap penyakit tertentu, sehingga basis pengetahuan dapat diperbarui tanpa mengubah kode program.

3.4 System Testing Results

3.4.1 Black Box Testing

Black Box Testing dilakukan pada 12 halaman atau fungsi sistem dengan total 28 skenario pengujian, meliputi pendaftaran, *login*, konsultasi, daftar penyakit, pengelolaan data penyakit, gejala, relasi, bobot, dan laporan, serta *logout*. Seluruh skenario berhasil dijalankan sesuai hasil yang diharapkan sehingga tingkat keberhasilan mencapai 100% (Tabel 4). Pada fungsi konsultasi, sistem menampilkan pesan kesalahan ketika pengguna tidak memilih gejala atau hanya memilih satu gejala, sesuai batasan minimal dua gejala.

Tabel 4. Rekapitulasi Black Box Testing

No	Halaman/Fungsi	Skenario	Berhasil	Gagal
1	Daftar User	2	2	0
2	Login User	2	2	0
3	Konsultasi	3	3	0
4	Daftar Penyakit	1	1	0
5	Logout User	1	1	0
6	Login Admin	2	2	0
7	Mengelola Data Penyakit	4	4	0
8	Mengelola Data Gejala	4	4	0
9	Mengelola Data Relasi	4	4	0
10	Mengelola Data Bobot	3	3	0
11	Mengelola Data Laporan	1	1	0
12	Logout Admin	1	1	0
	Total	28	28 (100%)	0 (0%)

3.4.2 White Box Testing

White Box Testing dilakukan pada 11 modul dengan menghitung *Cyclomatic Complexity* berdasarkan jumlah *edge* (E) dan *node* (N) pada *flowgraph* menggunakan persamaan (3). Hasilnya ditunjukkan pada Tabel 5.

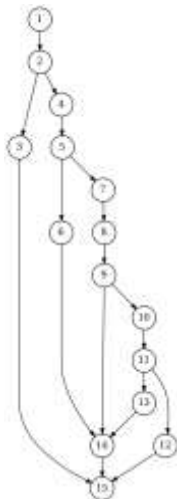
$$V(G) = E - N + 2 \quad (3)$$

Keterangan:

V(G) = nilai *Cyclomatic Complexity*;

E = jumlah *edge*;

N = jumlah *node*.



Gambar 8. Flowgraph proses konsultasi

Tabel 5. Hasil Cyclomatic Complexity per modul

Modul	E	N	V(G)
Daftar User	20	16	6
Login User	13	11	4
Konsultasi	18	15	5
Logout User	4	5	1
Login Admin	11	10	3
Data Penyakit	24	19	7
Data Gejala	32	24	10
Data Relasi	27	21	8
Data Bobot	20	16	6
Data Laporan	10	9	3
Logout Admin	3	4	1

Proses konsultasi sebagai inti perhitungan Dempster-Shafer memiliki *flowgraph* seperti pada Gambar 8 dengan E = 18 dan N = 15, sehingga $V(G) = 18 - 15 + 2 = 5$. Nilai tersebut menunjukkan terdapat lima jalur independen, yaitu: (1) tidak ada gejala yang dipilih; (2) gejala dipilih tetapi belum memiliki data bobot; (3) hasil dihitung pada konsultasi tamu; (4) hasil dihitung, pengguna *login*, namun riwayat gagal disimpan; dan (5) hasil dihitung, pengguna *login*, dan riwayat berhasil disimpan. Kompleksitas tertinggi terdapat pada modul pengelolaan data gejala dengan $V(G) = 10$, sedangkan modul *logout* memiliki $V(G) = 1$ karena hanya memiliki satu jalur eksekusi.

3.4.3 User Response

Pengujian *User Response* menggunakan kuesioner dengan 8 pertanyaan yang diisi oleh 13 responden sehingga diperoleh 104 jawaban. Skor hasil pengujian dihitung dengan mengalikan jumlah jawaban dengan bobotnya, kemudian persentase kelayakan dihitung menggunakan persamaan (4).

$$P = \frac{S}{S_{maks}} \times 100\% \quad (4)$$

Keterangan:

P = persentase kelayakan;

S = skor hasil pengujian;

S_{maks} = skor tertinggi = jumlah pertanyaan $\times$ bobot maksimal $\times$ jumlah responden.

Tabel 6. Hasil perhitungan *User Response*

Kriteria Jawaban	Jumlah Jawaban	Bobot	Skor
Baik	76	3	228
Cukup	27	2	54
Kurang Baik	1	1	1
Total	104	–	283

Skor hasil pengujian sebesar 283 dari skor tertinggi 312 ($8 \times 3 \times 13$) menghasilkan persentase kelayakan 90,7% atau dibulatkan menjadi 90%, dengan rata-rata skor 2,7 dari 3. Dengan demikian sistem dinyatakan layak digunakan dan telah memenuhi harapan pengguna. Hasil ini sebanding dengan sistem pakar Dempster-Shafer pada penelitian terdahulu yang melaporkan akurasi 83,33% hingga 94% [13], [14], [15], [16], walaupun ukuran yang digunakan berbeda: penelitian ini mengukur kelayakan menurut pengguna, bukan kesesuaian diagnosa terhadap diagnosa pakar.

4. CONCLUSION

Penelitian ini menghasilkan sistem pakar diagnosa penyakit dan hama tanaman terong berbasis website yang membantu petani pemula mengenali 10 jenis penyakit dan hama berdasarkan 25 gejala beserta solusi penanganannya. Metode Dempster-Shafer berhasil diterapkan untuk menangani ketidakpastian gejala dengan menggabungkan nilai *belief* setiap gejala; pada contoh perhitungan tiga gejala, Layu Bakteri diperoleh sebagai hasil diagnosa dengan nilai tertinggi 0,924. *Black Box Testing* terhadap 28 skenario berhasil 100%, *White Box Testing* menghasilkan *Cyclomatic Complexity* antara 1 dan 10 pada 11 modul, dan *User Response* dari 13 responden menghasilkan persentase kelayakan 90% sehingga sistem layak digunakan. Pengembangan selanjutnya dapat dilakukan dengan menambah data penyakit, hama, gejala, dan nilai *belief* dari beberapa pakar, serta menguji sistem dengan lebih banyak kasus dan membandingkan hasil diagnosa sistem dengan diagnosa pakar untuk mengukur akurasi.

REFERENCES

- [1] S. Sulardi, T. Hakim, M. Wasito, and N. Lubis, *Agribisnis Budidaya Tanaman Terong Ungu*. PT Dewangga Energi Internasional, 2022.
- [2] A. M. Ansari, W. Hasan, and M. Prakash, Eds., *Solanum melongena: Production, Cultivation and Nutrition*. Nova Science Publishers, 2021.
- [3] S. Ardila and A. A. Fajrin, "Sistem pakar mendiagnosis penyakit pada tanaman terong berbasis web," *Jurnal Comasie*, vol. 6, no. 5, pp. 45–51, 2022.
- [4] M. Rohini, N. Jayapala, H. Pushpalatha, H. Gavirangappa, H. Puttaswamy, and N. S. Ramachandrappa, "Biochemical, pathological and molecular characterisation of *Phomopsis vexans*: A causative of leaf blight and fruit rot in brinjal," *Microb. Pathog.*, vol. 179, Art. no. 106114, 2023.
- [5] S. K. Ghosh, "Thrips pest (Thrips palmi) infestation on eggplant/brinjal (*Solanum melongena* Linn.) and their sustainable management," in *Physical, Chemical and Biological Sciences: Emerging Trends and Milestones in 2020*, K. Muthuchelian et al., Eds. Virudhunagar Hindu Nadars' Senthikumara Nadar College, 2021, pp. 21–36.
- [6] S. K. Ghosh and W. Hasan, "Red spider mite (*Tetranychus urticae* Koch) infestation on brinjal/eggplant (*Solanum melongena* L.) and environmental sustainability," in *Hi-tech Crop Production and Pest Management*. Biotech Books, 2021, pp. 299–312.
- [7] L. Prabaningrum and T. K. Moekasan, *Ulat Grayak, Spodoptera spp.: Hama Polifag, Bioekologi dan Pengendaliannya*. IAARD Press, 2021.
- [8] S. Sutarman, A. E. Prihatiningrum, and D. Roeswitawati, Eds., *Pengelolaan Hama dan Penyakit Tanaman Hortikultur*. UMSIDA Press, 2022.
- [9] J. N. Srivastava and A. K. Singh, Eds., *Diseases of Horticultural Crops: Diagnosis and Management*, vol. 2, Vegetable Crops. CRC Press, 2022.
- [10] R. Destaria and Y. Yulmaini, "Sistem pakar diagnosa penyakit dispepsia menggunakan metode depth first search," in *Seminar Nasional Hasil Penelitian dan Pengabdian Masyarakat*, 2022, pp. 170–183.
- [11] M. Lutfi, S. Surorejo, and P. Septiana, "Systematic literature review: Penerapan algoritma naives Bayes dalam sistem pakar," *Jurnal Minfo Polgan*, vol. 11, no. 2, pp. 7–13, 2022.
- [12] H. Mayatopani, R. Subekti, N. Yudaningsih, and M. Sanwasih, "Pengembangan sistem pakar diagnosa gangguan mental dengan mesin inferensi menggunakan algoritma Dempster-Shafer theory," *Jurnal Buana Informatika*, vol. 13, no. 1, pp. 66–76, 2022, doi: 10.24002/jbi.v13i1.5568.

- [13] A. Heri and A. P. Irfan, "Sistem pakar diagnosa penyakit pada tanaman kakao menggunakan metode Dempster Shafer," *J. Comput. Inf. Syst. (J-CIS)*, vol. 6, no. 2, pp. 9–18, 2023.
- [14] R. P. A. Parapak, K. Saputra, H. Nasution, Z. Indra, and I. Taufik, "Sistem pakar diagnosa penyakit ginjal menggunakan metode Dempster Shafer di RSUD Pirngadi Medan," *Innovative: J. Soc. Sci. Res.*, vol. 4, no. 5, pp. 8457–8468, 2024.
- [15] A. Sidauruk, P. Suseno, B. Satria, and M. Sulistiyono, "Diagnosis penyakit tanaman kopi robusta menggunakan metode Dempster Shafer berbasis sistem pakar," *Indones. J. Comput. Sci.*, vol. 13, no. 4, 2024.
- [16] S. Rohman, R. Mahmudati, and W. A. Ansoruloh, "Sistem pakar diagnosa penyakit pada tanaman jambu kristal menggunakan metode Dempster Shafer berbasis web," *Jurnal Ilmiah Teknik Mesin, Elektro dan Komputer*, vol. 4, no. 1, pp. 65–73, 2024.
- [17] M. Syahputra, "Sistem pakar mendiagnosa penyakit ensefalitis menggunakan metode Dempster Shafer," *Jurnal SANTI - Sistem Informasi dan Teknik Informasi*, vol. 2, no. 1, pp. 1–9, 2022.
- [18] W. Wigel, B. Basri, and U. Khairat, "Penerapan metode Dempster Shafer untuk diagnosa penyakit pada tanaman nanas," *Journal Pegguruang: Conference Series*, vol. 5, no. 2, pp. 844–849, 2024.
- [19] D. S. Lumbanbatu, B. Anwar, and M. Dahria, "Sistem pakar mendiagnosa penyakit tanaman *Solanum betaccum* menggunakan metode Dempster Shafer," *Jurnal Sistem Informasi Triguna Dharma (JURSI TGD)*, vol. 1, no. 1, pp. 1–9, 2022.
- [20] F. R. Putri and A. Suharso, "Systematic literature review penggunaan metodologi pengembangan sistem informasi," *INFOTECH Journal*, vol. 9, no. 2, pp. 377–382, 2023, doi: 10.31949/infotech.v9i2.6270.
- [21] E. Rahmi, E. Yumami, and N. Hidayasari, "Analisis metode pengembangan sistem informasi berbasis website: Systematic literature review," *Remik: Riset dan E-Jurnal Manajemen Informatika Komputer*, vol. 7, no. 1, pp. 821–834, 2023.
- [22] M. R. Wayahdi and F. Ruziq, "Pemodelan sistem penerimaan anggota baru dengan Unified Modeling Language (UML) (Studi kasus: Programmer Association of Battuta)," *Jurnal Minfo Polgan*, vol. 12, no. 1, pp. 1514–1521, 2023, doi: 10.33395/jmp.v12i1.12870.
- [23] S. Nabila, A. R. Putri, A. Hafizhah, F. H. Rahmah, and R. Muslikhah, "Pemodelan diagram UML pada perancangan sistem aplikasi konsultasi hewan peliharaan berbasis Android (Studi kasus: Alopel)," *Jurnal Ilmu Komputer dan Bisnis*, vol. 12, no. 2, pp. 130–139, 2021, doi: 10.47927/jikb.v12i2.150.
- [24] S. W. Ramdany, S. A. Kaidar, B. Aguchino, C. A. A. Putri, and R. Anggie, "Penerapan UML class diagram dalam perancangan sistem informasi perpustakaan berbasis web," *Journal of Industrial and Engineering System*, vol. 5, no. 1, 2024, doi: 10.31599/2e9afp31.
- [25] R. Noviana, "Pembuatan aplikasi penjualan berbasis web Monja Store menggunakan PHP dan MySQL," *Jurnal Teknik dan Science*, vol. 1, no. 2, pp. 112–124, 2022, doi: 10.56127/jts.v1i2.128.
- [26] C. Christian and A. Voutama, "Implementasi aplikasi antrian pencucian mobil berbasis web menggunakan PHP, JavaScript, HTML, CSS dan UML," *Jurnal Mahasiswa Teknik Informatika*, vol. 8, no. 1, pp. 1–27, 2024.
- [27] N. A. Murani, S. Wahyuni, R. A. Pratiwi, T. Firdaus, and Wardianto, "Analisis penggunaan Visual Studio Code sebagai editor pemrograman terbaik untuk pemula pada mahasiswa informatika Universitas Nurul Huda," *Jurnal Rekayasa Informatika (JRIT)*, vol. 2, no. 2, pp. 1–10, 2025.
- [28] A. Fauzi and A. H. Yunial, *Testing dan QA Perangkat Lunak: Memahami Konsep, Metode, Teknik, dan Pendekatan dalam Pengujian*, 2025.
- [29] I. R. Dhaifullah, M. Muttanifudin H., A. Ananda Salsabila, and M. Ainul Yaqin, "Survei teknik pengujian software," *Journal Automation Computer Information System*, vol. 2, no. 1, pp. 31–38, 2022, doi: 10.47134/jacis.v2i1.42.