

ANALISIS PENGUJIAN SISTEM PAKAR PENYAKIT SELADA MENGGUNAKAN METODE BLACK BOX & WHITE BOX TESTING

Nela Puspita Florensia ^{1*}, Rangga ², Yulia Patimah³, Yusie Nur Chusnul Khotimah,⁴

^{1,2}Program Studi Teknik Informatika, Universitas Palangka Raya; Jl. Yos Sudarso, Palangka, Kec. Jekan Raya, Kota Palangka Raya, Kalimantan Tengah 73111; (0536)

Received: 8 Maret 2025
Accepted: 27 Maret 2025
Published: 14 April 2025

Keywords:

Sistem Pakar;
Pengujian Perangkat Lunak;
White Box Testing;
Black box Testing;
Diagnosa Penyakit Tanaman.

Abstrak. Penelitian ini menganalisis pengujian *Sistem Pakar Selada* menggunakan metode *Black Box* dan *White Box Testing* untuk memastikan keakuratan diagnosis penyakit tanaman. *Unit Testing* dilakukan dengan PHPunit untuk menguji fitur utama sistem, mencakup skenario normal, tepi, dan kesalahan. *Integration Testing* mengevaluasi integrasi modul pada halaman diagnosa, sedangkan *System Testing* memastikan seluruh sistem berfungsi sesuai spesifikasi. *Validation Testing* diterapkan dengan *Alpha* dan *Beta Testing* untuk menilai penerimaan pengguna. Hasil pengujian menunjukkan bahwa sistem bekerja stabil, memberikan diagnosa akurat, serta dapat digunakan oleh petani secara efektif. Pengembangan lebih lanjut dapat mencakup integrasi IoT dan optimasi kecepatan pemrosesan data.

Correspondent Email:

nelapsptaf@gmail.com

Abstract. This study analyzes the testing of the *Lettuce Expert System* using *Black Box* and *White Box Testing* methods to ensure accurate plant disease diagnosis. *Unit Testing* with PHPunit evaluates the system's main features, covering normal, boundary, and error scenarios. *Integration Testing* assesses module integration on the diagnosis page, while *System Testing* ensures overall system functionality according to specifications. *Validation Testing* is conducted through *Alpha* and *Beta Testing* to evaluate user acceptance. The results indicate that the system operates stably, provides accurate diagnoses, and can be effectively used by farmers. Further development may include IoT integration and optimization of data processing speed.

1. PENDAHULUAN

Pertanian merupakan sektor krusial dalam perekonomian Indonesia, tetapi sering menghadapi tantangan besar seperti serangan hama dan penyakit tanaman yang dapat menurunkan hasil panen secara signifikan. Untuk mengatasi masalah ini, pemanfaatan teknologi informasi, seperti sistem pakar, menjadi solusi yang efektif dalam membantu petani mendiagnosis penyakit tanaman dengan lebih cepat dan akurat [1].

Sistem pakar sendiri adalah program komputer yang dirancang untuk meniru

kemampuan seorang ahli dalam menyelesaikan masalah spesifik, seperti diagnosis penyakit pada tanaman. Dalam dunia pertanian, teknologi ini sangat berguna karena memungkinkan petani untuk mengenali gejala penyakit sejak dini dan mengambil langkah-langkah pencegahan atau pengobatan yang sesuai [2].

Salah satu metode yang banyak digunakan dalam sistem pakar adalah forward chaining, yang bekerja dengan memulai dari data awal atau gejala yang diketahui, kemudian menerapkan aturan-aturan tertentu untuk

mencapai kesimpulan atau diagnosis. Beberapa penelitian sebelumnya menunjukkan bahwa metode ini telah berhasil diterapkan dalam diagnosis penyakit tanaman, seperti pada tanaman padi, dan mampu membantu petani dalam mengenali penyakit dengan lebih akurat berdasarkan gejala yang muncul [3].

Dalam pengembangannya, sistem pakar umumnya menggunakan pendekatan Software Development Life Cycle (SDLC) dengan model Waterfall, yang terdiri dari beberapa tahap seperti analisis kebutuhan, desain sistem, implementasi, pengujian, dan pemeliharaan. Pendekatan ini memastikan bahwa sistem dikembangkan dengan prosedur yang sistematis dan memenuhi standar kualitas yang tinggi. Salah satu tahapan penting dalam siklus ini adalah pengujian perangkat lunak untuk memastikan sistem berfungsi sesuai dengan spesifikasi yang diharapkan.

Salah satu metode yang sering digunakan dalam pengujian perangkat lunak adalah *Black Box Testing*, yang berfokus pada pemeriksaan fungsionalitas sistem tanpa memperhatikan struktur internalnya. Metode ini memungkinkan pengembang untuk memastikan bahwa sistem memberikan output yang benar sesuai dengan input yang dimasukkan serta mampu menangani berbagai kemungkinan kesalahan dengan baik [4].

Berdasarkan hal tersebut, penelitian ini bertujuan untuk mengembangkan dan menguji Sistem Pakar Selada dengan menggunakan algoritma forward chaining serta pendekatan SDLC model *Waterfall*. Pengujian akan dilakukan menggunakan metode *Black Box Testing* untuk memastikan sistem dapat mendiagnosis penyakit tanaman selada dengan akurat dan dapat digunakan sebagai alat bantu yang efektif bagi petani dalam meningkatkan hasil pertanian mereka.

2. TINJAUAN PUSTAKA

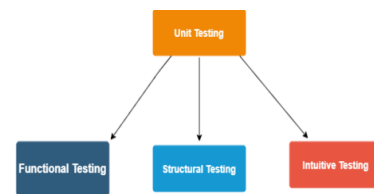
2.1. Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah fase krusial dalam proses pengembangan sistem untuk memastikan bahwa produk akhir memenuhi kriteria kualitas dan harapan pengguna[5]. Pengujian perangkat lunak membantu dalam mendeteksi serta memperbaiki kesalahan sebelum perangkat lunak digunakan secara luas [6]. Pengujian ini dapat dilakukan menggunakan berbagai

metode, seperti *Unit Testing*, *Integration Testing*, *Validation Testing*, dan *System Testing*.

2.2. Unit Testing

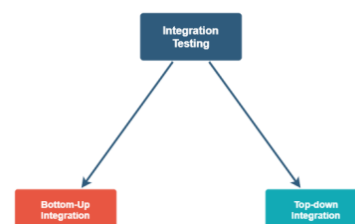
Unit adalah bagian terkecil dari kode yang dapat diuji secara independen dalam sebuah sistem. *Unit Testing* merupakan salah satu tahap pengujian perangkat lunak yang berfungsi untuk memastikan bahwa setiap unit atau modul dapat berjalan dengan baik sebelum diintegrasikan ke dalam sistem yang lebih besar. Pengujian ini memiliki tujuan untuk memverifikasi setiap komponen kode secara terpisah sebelum digabungkan ke dalam sistem secara menyeluruh. Pada studi ini, pengujian unit dilaksanakan dengan memanfaatkan framework PHPUnit melalui tiga skenario pengujian, yaitu kondisi normal, kasus batas, dan situasi error[7].



Gambar 1. Hirarki Unit Testing

2.3. Integration Testing

Integration Testing adalah proses pengujian yang berfungsi untuk memverifikasi bahwa seluruh komponen sistem dapat beroperasi secara optimal setelah dilakukan proses penggabungan[8]. Berbeda dengan *Unit Testing* yang berfokus pada pengujian komponen secara individu, *Integration Testing* memastikan bahwa setiap modul dapat bekerja sama dengan baik dan berinteraksi sesuai dengan yang diharapkan. Pengujian ini menitikberatkan pada bagaimana modul saling berkomunikasi dan apakah ada masalah dalam pertukaran data di antara mereka.



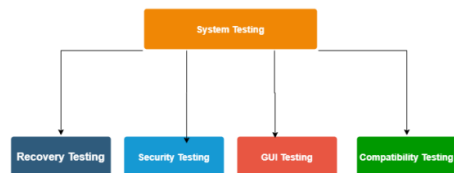
Gambar 2. Hirarki Integration Testing

2.4. Validation Testing

Validation Testing adalah tahap pengujian akhir dalam pengembangan perangkat lunak untuk memastikan bahwa sistem yang dibuat benar-benar sesuai dengan kebutuhan dan harapan pengguna. Pengujian ini bertujuan untuk memverifikasi apakah perangkat lunak dapat digunakan sebagaimana mestinya sebelum dirilis. Proses evaluasi ini mengimplementasikan pendekatan Alpha dan Beta Testing guna mengukur tingkat kepuasan dan penerimaan end-user terhadap sistem yang telah selesai dikembangkan[9].

2.5. System Testing

System Testing adalah tahap pengujian akhir dalam perangkat lunak atau perangkat keras, di mana seluruh sistem diuji secara menyeluruh untuk memastikan bahwa semuanya berfungsi sesuai dengan persyaratan yang telah ditentukan. Pengujian ini memusatkan perhatian pada evaluasi komprehensif terhadap sistem guna memverifikasi bahwa seluruh fungsi beroperasi sesuai dengan ketentuan yang telah ditetapkan[10].



Gambar 3. Hirarki System Testing

3. METODE PENELITIAN

3.1. Analisis Sistem

Penelitian ini diawali dengan menganalisis kebutuhan sistem untuk memastikan bahwa Sistem Pakar Selada yang dikembangkan dapat membantu petani dalam mendiagnosis penyakit tanaman dengan cepat dan akurat. Proses analisis ini mencakup identifikasi masalah utama yang sering dihadapi petani, seperti sulitnya mendeteksi penyakit tanaman sejak dini. Untuk memastikan sistem bekerja dengan baik, digunakan metode *Black Box Testing*, yang memungkinkan pengujian tanpa perlu melihat kode sumber. Pengujian dilakukan secara bertahap, mulai dari *Unit Testing* untuk menguji setiap bagian kecil sistem, *Integration Testing* untuk memastikan modul-modul dapat

bekerja sama, *Validation Testing* untuk mengecek apakah sistem sudah sesuai dengan kebutuhan pengguna, hingga *System Testing* untuk memastikan seluruh sistem dapat berjalan dengan baik dalam berbagai kondisi.

3.2. Arsitektur Sistem

Sistem ini dikembangkan menggunakan metode Waterfall, yaitu pendekatan berurutan yang dimulai dari analisis kebutuhan, dilanjutkan dengan perancangan sistem, implementasi, pengujian, dan terakhir pemeliharaan untuk memastikan sistem tetap optimal. Dalam sistem ini, diterapkan metode Forward Chaining, di mana sistem akan memproses data awal berupa gejala tanaman dan mencocokkannya dengan aturan diagnosis yang telah ditentukan hingga menghasilkan kesimpulan berupa jenis penyakit yang terdeteksi. Dengan arsitektur ini, sistem dapat memberikan rekomendasi diagnosis yang lebih akurat dan sistematis kepada pengguna.

3.3. Metode Penyelesaian Masalah

Untuk mempermudah petani dalam mendeteksi penyakit tanaman selada, sistem ini dirancang menggunakan pendekatan Sistem Pakar berbasis aturan. Sistem ini bekerja dengan cara menelusuri gejala yang dimasukkan oleh pengguna dan mencocokkannya dengan data penyakit yang sudah tersimpan dalam sistem. Dengan menerapkan metode Forward Chaining, sistem akan menganalisis data dari tahap awal hingga akhir untuk mendapatkan hasil diagnosis yang paling sesuai. Selain itu, agar sistem lebih handal, dilakukan pengujian menggunakan metode *Black Box Testing*, yang membantu mengidentifikasi kekurangan dalam sistem tanpa harus memeriksa kode pemrogramannya.

3.4. Implementasi Sistem

Pada tahap implementasi, sistem dikembangkan dalam bentuk aplikasi berbasis web menggunakan PHP sebagai bahasa pemrograman dan MySQL untuk menyimpan basis data. Pengembangan dilakukan dengan beberapa langkah, mulai dari pembuatan antarmuka pengguna yang mudah digunakan, penyusunan basis pengetahuan berisi aturan diagnosis penyakit, hingga integrasi data agar sistem dapat bekerja secara optimal. Setelah pengembangan selesai, sistem diuji

menggunakan berbagai metode seperti *Unit Testing*, *Integration Testing*, dan *System Testing* untuk memastikan fungsionalitasnya berjalan dengan baik. Jika ditemukan kekurangan, dilakukan evaluasi dan perbaikan agar sistem semakin optimal dan dapat digunakan dengan lebih efektif oleh petani dalam mendeteksi penyakit tanaman selada.

4. HASIL DAN PEMBAHASAN

4.1. Unit Testing

Bagian *Unit Testing* dilakukan dengan metode *White Box Testing* menggunakan *PHPUnit* untuk menguji fitur utama website Sistem Pakar Selada. Pengujian fokus pada halaman pertanyaan, di mana pengguna memberikan jawaban "IYA" atau "TIDAK" untuk menentukan kondisi selada. Sistem kemudian menampilkan hasil diagnosis berdasarkan gejala yang diinput. Skenario pengujian mencakup kasus normal, kasus tepi, dan kasus kesalahan.

A. Skenario Kasus Normal

Pada skenario kasus normal, fungsi *testFetchGejala()* diuji untuk melakukan verifikasi terhadap tampilan data gejala yang optimal serta memastikan proses eksekusi query SQL berjalan tanpa kendala[11]. Pengujian mencakup dua *assertions*: pertama, memastikan *Query* sukses (*True*), dan kedua, memastikan tabel gejala tidak kosong (*True*). Hasilnya menunjukkan pengujian berhasil dengan *Query* sukses dan data tersedia.

B. Skenario Kasus Tepi

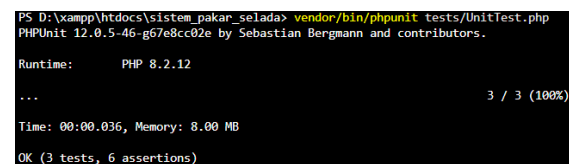
Pada skenario kasus tepi, fungsi *testGejalaDataFormat()* diuji dengan mengambil satu data pertama dari *database* untuk memverifikasi formatnya. Pengujian memastikan adanya kolom *id_gejala* dan *nama_gejala*, serta tipe data *nama_gejala* adalah *String*. Tiga *assertions* digunakan: pertama untuk memverifikasi *id_gejala*, kedua untuk *nama_gejala*, dan ketiga untuk tipe data *nama_gejala*. Hasil pengujian sukses dengan data sesuai format yang diharapkan.

C. Skenario Kasus Kesalahan

Pada skenario kasus kesalahan, melakukan evaluasi terhadap performa sistem dengan mengosongkan data pada tabel gejala guna memverifikasi bahwa sistem dapat beroperasi secara normal meskipun tanpa adanya data[12]. Pengujian menggunakan fungsi

testEmptyGejalaTable(), dengan satu *assertion* untuk memastikan tabel benar-benar kosong. Hasilnya menunjukkan *Query* tetap berjalan tanpa gagal, meskipun tidak ada data yang dikembalikan, sehingga pengujian berhasil.

Berdasarkan ketiga Skenario *Unit Testing* tersebut dapat disimpulkan bahwa pengujian ini telah berhasil keseluruhannya dengan semua pengujian berjalan dengan benar, tidak ada error dalam *Query* atau format data, serta *database* dapat menangani operasi CRUD dengan benar. Berikut adalah bukti hasil pengujian *Unit Testing* dengan Source Code pengujiannya.



```
PS D:\xampp\htdocs\sistem_pakar_selada> vendor/bin/phpunit tests/UnitTest.php
PHPUnit 12.0.5-46-g67e8cc02e by Sebastian Bergmann and contributors.

Runtime:       PHP 8.2.12
...
Time: 00:00.036, Memory: 8.00 MB

OK (3 tests, 6 assertions)
```

Gambar 4. Hasil *Unit Testing* Sistem Pakar Selada



```
1 <?php
2
3 use PHPUnit\Framework\TestCase;
4
5 class UnitTest extends TestCase {
6     private $koneksi;
7
8     protected function setUp(): void {
9         // Simpan koneksi database ke dalam variabel agar dapat digunakan kembali
10        $this->koneksi = new mysqli("localhost", "root", "", "db_sistem_pakar");
11
12        if ($this->koneksi->connect_error) {
13            die("Koneksi database gagal: " . $this->koneksi->connect_error);
14        }
15    }
16
17    // 1. Test Query Mengambil Data dari Database
18    public function testFetchGejala() {
19        $query = "SELECT * FROM tb_gejala";
20        $result = $this->koneksi->query($query);
21
22        $this->assertNotNull($result, "Query gagal dijalankan");
23        $this->assertGreaterThan(0, $result->num_rows, "Data gejala kosong");
24    }
25
26    // 2. Test Format Data yang Diambil
27    public function testGejalaDataFormat() {
28        $query = "SELECT * FROM tb_gejala LIMIT 1";
29        $result = $this->koneksi->query($query);
30        $gejala = $result->fetch_assoc();
31
32        $this->assertArrayHasKey('id_gejala', $gejala, "Kolom id_gejala tidak ditemukan");
33        $this->assertArrayHasKey('nama_gejala', $gejala, "Kolom nama_gejala tidak ditemukan");
34        $this->assertString($gejala['nama_gejala'], "Nama gejala harus berupa string");
35    }
36
37    // 3. Test Jika Tabel tb_gejala Kosong
38    public function testEmptyGejalaTable() {
39        // Buat kondisi tabel kosong dengan menghapus semua data (pastikan ini database pengujian!)
40        $this->koneksi->query("DELETE FROM tb_gejala");
41
42        $query = "SELECT * FROM tb_gejala";
43        $result = $this->koneksi->query($query);
44
45        $this->assertEquals(0, $result->num_rows, "Tabel tb_gejala tidak kosong setelah penghapusan");
46
47        // Restore data setelah pengujian
48        $this->koneksi->query("INSERT INTO tb_gejala (id_gejala, nama_gejala) VALUES (1, 'Daun menguning')");
49    }
50
51    protected function tearDown(): void {
52        $this->koneksi->close();
53    }
54 }
```

Gambar 5. Source Code *Unit Testing* Sistem Pakar Selada

4.2. Integration Testing

Integration Testing dengan metode *Black Box Testing* dilakukan untuk memastikan modul sistem, terutama pada halaman diagnosa penyakit, bekerja secara terintegrasi. Pengujian ini memverifikasi apakah sistem dapat menghasilkan diagnosis penyakit selada yang sesuai berdasarkan basis pengetahuan, tanpa

melihat kode program. Pengguna memberikan input berupa jawaban "Iya" atau "Tidak" untuk setiap gejala, yang kemudian diproses untuk mencocokkan pola gejala dengan data penyakit dalam knowledge base. Fokus utama pengujian adalah keakuratan hasil diagnosis tanpa memeriksa logika internal sistem.

Tabel 1. Tabel Integrarian Testing Sistem Pakar Selada

No	Test Case	Input (Jawaban dari User)	Expected Output	Status
1	Diagnosa penyakit Busuk Batang	Tidak, Iya, Iya, Iya, Tidak, Tidak, Tidak, Iya, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak	Busuk Batang	Berhasil
2	Diagnosa penyakit Busuk Lunak	Tidak, Iya, Tidak, Iya, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak	Busuk Lunak	Berhasil
3	Diagnosa penyakit Busuk Pangkal Daun	Tidak, Tidak, Tidak, Tidak, Iya, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak	Busuk Pangkal Daun	Berhasil
4	Diagnosa penyakit Mata Kodok	Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Iya, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak	Mata Kodok	Berhasil
5	Diagnosa penyakit Layu Fusarium	Iya, Tidak, Tidak, Tidak, Tidak, Iya, Tidak, Tidak, Tidak, Iya, Iya, Iya, Tidak, Tidak, Tidak, Tidak	Layu Fusarium	Berhasil
6	Diagnosa penyakit Busuk Akar	Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Iya, Tidak, Tidak, Tidak	Busuk Akar	Berhasil
7	Input tidak sesuai (Jawaban tidak cocok dengan	Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak, Tidak	Sistem menampilkan "Selada anda Sehat"	Berhasil

penyakit yang ada)	Tidak, Tidak, Tidak, Tidak		
--------------------	----------------------------	--	--

4.3. System Testing

System Testing memastikan sistem pakar selada berfungsi sesuai spesifikasi menggunakan metode *Black Box Testing*, yang berfokus pada fungsionalitas tanpa melihat kode sumber. Pengujian mencakup skenario fungsional dan non-fungsional untuk menguji seluruh alur kerja pengguna dari awal hingga akhir.

A. Skenario Pengujian Registrasi

Tabel 2. Tabel Pengujian Registrasi Sistem Pakar Selada

No	Deskripsi	Hasil yang di Harapkan	Status
1	Masukkan nama, alama, <i>password</i> , nomor telpon, dan <i>username</i> yang benar ke dalam sistem	Sistem berhasil registrasi, dan menampilkan pesan "Registrasi Berhasil."	Berhasil

B. Skenario Pengujian Login

Tabel 3. Tabel Pengujian Login Sistem Pakar Selada

No	Deskripsi	Hasil yang di Harapkan	Status
1	Masukkan <i>username</i> dan <i>password</i> yang benar ke dalam sistem	Sistem berhasil masuk ke halaman beranda	Berhasil
2	Masukkan <i>username</i> dan <i>password</i> yang salah ke dalam sistem	Sistem tidak berhasil masuk ke halaman beranda, dan menampilkan pesan, "Data Salah!!."	Berhasil

C. Skenario Pengujian Diagnosa

Tabel 4. Tabel Pengujian Diagnosa Sistem Pakar Selada

No	Deskripsi	Hasil yang di Harapkan	Status
1	Pengguna memilih gejala yang dialami tanaman selada	Sistem memberikan hasil diagnosa	Berhasil
2	Data hasil diagnosa dan solusi disimpan dalam sistem	Solusi tampil sesuai dengan hasil diagnosa	Berhasil
3	Pengguna melakukan logout dari sistem	Sistem kembali ke halaman login	Berhasil

D. Skenario Pengujian Non-Fungsional

Tabel 5. Tabel Pengujian Non-Fungsional Sistem Pakar Selada

No	Jenis Pengujian	Deskripsi	Hasil yang Diharapkan	Status
1	Kinerja	Sistem diuji dalam lingkungan lokal untuk melihat kecepatan respon dan stabilitasnya	Sistem tetap berjalan dengan baik dalam localhost	Berhasil
2	Keamanan	Uji validasi input terhadap SQL Injection dan XSS	Sistem tidak rentan terhadap serangan	Berhasil
3	Kegunaan (Usability)	Evaluasi kemudahan navigasi bagi pengguna	Antarmuka mudah digunakan	Berhasil

Setelah pengujian selesai, hasil dievaluasi untuk memastikan sistem berfungsi sesuai dengan spesifikasi. Jika sistem berhasil melewati semua pengujian, maka dinyatakan siap untuk digunakan. Namun, jika ditemukan bug atau kesalahan, perbaikan akan dilakukan sebelum pengujian ulang. Pengujian sistem pakar selada dengan metode *Black Box Testing* memastikan bahwa sistem dapat digunakan secara efektif oleh pengguna.

4.4. Validation Testing

Validation Testing pada *website* Sistem Pakar Selada memastikan bahwa sistem memenuhi spesifikasi dan dapat diterima pengguna. Pengujian dilakukan dengan metode *Black Box Testing* untuk menilai fungsionalitas tanpa melihat kode sumber. Selain itu, untuk mengidentifikasi error dari sisi pengguna, dilakukan Alpha Testing dan Beta Testing guna memastikan sistem berjalan sesuai kebutuhan dan ekspektasi.

A. Alpha Testing

Alpha Testing dilakukan di lingkungan *developer* dengan melibatkan pengguna (*customer*). Pengguna mencoba perangkat

lunak sementara *developer* memantau dan mencatat kesalahan atau permasalahan yang mungkin muncul.

Tabel 6. Tabel Alpha Testing Sistem Pakar Selada

No	Test Case	Skenario Pengguna	Expected Outcome	Actual Outcome	Status
1	Navigasi Sistem	Pengguna mencoba mengakses fitur diagnosa penyakit	Fitur diagnosa mudah diakses dan digunakan	Fitur dapat diakses dengan baik	Berhasil
2	Input Data Gejala	Pengguna memasukkan gejala tanaman	Sistem menerima input dengan benar tanpa error	Input diterima dengan baik	Berhasil
3	Akurasi Diagnosa	Pengguna memasukkan gejala penyakit tertentu	Sistem memberikan hasil diagnosa yang benar	Diagnosa sesuai dengan basis pengetahuan	Berhasil
4	Handling Error	Pengguna memasukkan data tidak valid	Sistem memberikan peringatan/error handling yang sesuai	Sistem menampilkan pesan yang tepat	Berhasil
5	Performansi Sistem	Pengguna menguji respons sistem dengan banyak data	Sistem tetap berjalan dengan baik tanpa lag	Sistem tetap responsif	Berhasil

B. Beta Testing

Beta Testing dilakukan oleh pengguna akhir (*end user*) tanpa kehadiran *developer*. Pengguna mencoba perangkat lunak dalam lingkungan operasional yang sebenarnya, lalu mencatat semua masalah dan memberikan laporan kepada *developer* untuk perbaikan lebih lanjut.

Tabel 7. Tabel Beta Testing Sistem Pakar Selada

No	Test Case	Skenario Pengguna	Feedback dari End User	Status
1	Akses Sistem	Pengguna mencoba mengakses sistem di berbagai perangkat	Sistem dapat digunakan di <i>mobile</i> dan PC dengan baik	Berhasil
2	Kemudahan Pengguna	Pengguna mencoba memahami cara kerja sistem tanpa panduan	Sebagian besar pengguna dapat memahami cara kerja sistem dengan mudah	Berhasil
3	Akurasi Diagnosa	Pengguna membandingkan hasil diagnosa dengan kondisi nyata tanaman	Hasil diagnosa sesuai dengan gejala yang di- <i>input</i>	Berhasil
4	Error Handling	Pengguna memasukkan <i>input</i> tidak valid atau kosong	Sistem menampilkan peringatan yang jelas	Berhasil
5	Performa Sistem	Pengguna menggunakan sistem dalam kondisi jaringan yang kurang stabil	Sistem tetap berjalan, meskipun sedikit lebih	Perlu optimasi untuk koneksi lambat

			lambat pada koneksi buruk	
--	--	--	---------------------------	--

Berdasarkan hasil pengujian *Alpha* dan *Beta Testing*, sistem pakar selada telah memenuhi kebutuhan pengguna berdasarkan spesifikasi

yang telah dirancang. Beberapa perbaikan terkait tampilan antarmuka dan optimasi kinerja disarankan sebelum sistem dirilis secara penuh kepada pengguna akhir.

5. KESIMPULAN

Berdasarkan hasil penelitian dan pengujian, Sistem Pakar Selada terbukti mampu memberikan diagnosa penyakit tanaman dengan akurasi yang tinggi. Pengujian menggunakan metode *Black Box Testing* memastikan bahwa sistem bekerja sesuai dengan spesifikasi yang telah dirancang. Selain itu, sistem ini juga menunjukkan performa yang stabil dan dapat digunakan dalam berbagai kondisi.

Keunggulan utama dari sistem ini terletak pada kemampuannya dalam menganalisis gejala tanaman dengan cepat serta memberikan hasil yang akurat. Selain itu, tampilan antarmuka yang sederhana membuat sistem ini mudah digunakan oleh petani tanpa memerlukan keahlian teknis khusus. Pengolahan data yang efisien juga memungkinkan sistem memberikan hasil diagnosa dalam waktu singkat, sehingga membantu petani dalam mengambil tindakan pencegahan lebih dini.

Meskipun sistem ini telah berfungsi dengan baik, masih ada peluang pengembangan lebih lanjut. Salah satunya adalah dengan mengintegrasikan teknologi Internet of Things (IoT) agar petani dapat memantau kondisi tanaman secara real-time. Selain itu, optimasi pada kecepatan pemrosesan data juga diperlukan agar sistem dapat bekerja lebih responsif dan efektif. Melalui proses pengembangan yang terus-menerus dilakukan, Sistem Pakar Selada diharapkan mampu menjadi sebuah inovasi teknologi yang membantu para petani dalam memantau dan memelihara kondisi tanaman secara lebih efektif [13].

DAFTAR PUSTAKA

- [1] K. Aryasa, "Sistem Pakar Diagnosa Penyakit Tanaman Agribisnis Menggunakan Metode Certainty Factor," *E-JURNAL JUSITI : Jurnal Sistem Informasi dan Teknologi Informasi*, vol. 7, no. 1, pp. 54–67, Aug. 2019, [Online]. Available: <https://ejurnal.undipa.ac.id/index.php/jusiti/article/view/64>
- [2] F. Mulyadi, "Sistem Pakar Dalam Diagnosa Penyakit Pada Tanaman," *JURNAL TIKA*, vol. 7, no. 1, pp. 39–47, Apr. 2022, doi: 10.51179/tika.v7i1.1084.
- [3] N. A. Muslimah, "SISTEM PAKAR TENTANG MENDIAGNOSA PENYAKIT PADA TANAMAN PADI MENGGUNAKAN METODE FORWARD CHAINING," UNIVERSITAS MUHAMMADIYAH MAKASSAR, Makassar, 2024.
- [4] E. Setiana, M. R. Ramadhan, Budiman, and R. Yadi Rakhman A4, "Pengujian Perangkat Lunak Metode Black Box Pada Aplikasi Sistem Pakar Pola Latihan dan Asupan Makanan," *NUANSA INFORMATIKA*, vol. 18, no. 1, pp. 68–74, Jan. 2024, doi: 10.25134/ilkom.v18i1.67.
- [5] R. S. Widodo, "Implementasi Sistem Pakar Untuk Diagnosa Penyakit Tanaman Hortikultura," *Jurnal Teknologi Informasi*, vol. 10, no. 2, pp. 34–45, 2021.
- [6] Uminingsih, M. Nur Ichsanudin, M. Yusuf, and S. Suraya, "PENGUJIAN FUNGSIONAL PERANGKAT LUNAK SISTEM INFORMASI PERPUSTAKAAN DENGAN METODE BLACK BOX TESTING BAGI PEMULA," *STORAGE: Jurnal Ilmiah Teknik dan Ilmu Komputer*, vol. 1, no. 2, pp. 1–8, May 2022, doi: 10.55123/storage.v1i2.270.
- [7] T. Kurniawan, "Perancangan Sistem Pakar Dengan Pendekatan Forward Chaining Untuk Diagnosa Penyakit Tanaman," *Jurnal Informatika Pertanian*, vol. 5, no. 1, pp. 22–30, 2020.
- [8] L. Prasetyo, "Evaluasi Pengujian Perangkat Lunak Menggunakan Metode White Box dan Black Box Testing," *Jurnal Teknologi dan Rekayasa Sistem*, vol. 12, no. 3, pp. 56–65, 2023.
- [9] Y. R. Saputra, "Pengembangan Sistem Pakar Berbasis Web Untuk Mendiagnosa Penyakit Tanaman Pangan," *Jurnal Ilmu Komputer Terapan*, vol. 6, no. 2, pp. 45–55, 2021.
- [10] Y. R. Saputra, "Pengembangan Sistem Pakar Berbasis Web Untuk Mendiagnosa Penyakit Tanaman Pangan," *Jurnal Ilmu Komputer Terapan*, vol. 6, no. 2, pp. 45–55, 2021.
- [11] D. Ramadhani, "Implementasi Machine Learning Pada Sistem Pakar Diagnosa Penyakit Tanaman," *Jurnal Kecerdasan Buatan dan Sistem Pakar*, vol. 4, no. 1, pp. 23–33, 2023.
- [12] B. Nurdin, "Analisis Pengujian Perangkat Lunak Menggunakan Metode Black Box Testing," *Jurnal Rekayasa Perangkat Lunak*, vol. 9, no. 2, pp. 12–21, 2020.
- [13] M. Hakim, "Sistem Pakar Untuk Identifikasi Penyakit Tanaman Sayuran Berbasis Web," *Jurnal Teknologi dan Inovasi Pertanian*, vol. 7, no. 1, pp. 67–76, 2022.