

MENDETEKSI HAMA TANAMAN BUAH MANGGA DENGAN METODE CERTAINTY FACTOR

Penulis : Gatot Fitriyanto¹⁾, Nur Ahmad Azhar²⁾, Muhammad Kurniawan³⁾

^{1), 2), 3)}Teknik Informatika STMIK AMIKOM Yogyakarta
Jl Ring road Utara, Condongcatur, Sleman, Yogyakarta 55281
Email : gatot.f@students.amikom.ac.id¹⁾, nur.az@students.amikom.ac.id²⁾,
muhammad.7071@students.amikom.ac.id³⁾

Abstrak

Hama penyakit yang menyerang tanaman mangga tersebut menyebabkan para pembudidaya tanaman mangga mengalami kerugian akibat kerusakan tanaman mangga tersebut. Untuk membantu mendeteksi hama dan penyakit yang menyerang tanaman mangga apabila pakar atau ahli pertanian belum ada, maka diperlukan system berbasis komputer yang dapat menginformasikan tentang hama dan penyakit dan pengendaliannya.

Penelitian terhadap tanaman tersebut bertujuan untuk membangun sebuah system pakar berbasis aturan (rule based) dengan menggunakan metode certainty factor untuk mendeteksi hama penyakit pada tanaman mangga berdasarkan gejala –gejala tersebut. Metode certainty factor tidak dapat berdiri sendiri, maka dari itu didukung dengan metode forward chaining yang melakukan penalaran dari sekumpulan fakta-fakta yang ada menuju kesimpulan.

Untuk mendeteksi hama dan penyakit pada tanaman mangga yang berdasarkan gejala-gejala tersebut, maka diperlukan seorang ahli untuk menganalisis hama penyakit yang menyerang tanaman mangga dan memberikan solusi cara pengendaliannya. System pakar ini dapat bermanfaat bagi para pembudidaya atau Pengamat Organisme Pengganggu Tanaman (POPT) sebagai pembelajaran dan memahami hama penyakit tanaman mangga.

Kata kunci : POPT, Sistem Pakar, Certainty Factor.

1. Pendahuluan

Latar Belakang Masalah

Hampir sebagian besar masyarakat di Indonesia menanam tanaman mangga di pekarangan mereka. Untuk pembudidayaan dengan skala agrobisnis tanaman tersebut, hanya terdapat pada daerah tertentu seperti Probolinggo, Indramayu dan Cirebon. Dalam pembudidayaan diperlukan pemahaman mengenai factor-faktor yang menjadi pengganggu budidaya tanaman mangga seperti hama dan penyakit. Dengan pengetahuan yang cukup, di harapkan para pembudidaya dapat

meminimalisir kerugian akibat serangan hama dan penyakit.

Rumusan Masalah

Masalah yang diangkat dalam penelitian ini adalah membangun atau merancang aplikasi untuk mendeteksi hama dan penyakit pada tanaman mangga dengan metode certainty factor dan forward chaining.

Batasan Masalah

Masalah yang diakibatkan hama dan penyakit sangat beragam karena banyak factor-faktor yang mempengaruhi, maka dibentuk batasan masalah sebagai berikut:

- Hama dan penyakit pada tanaman mangga.
- Sistem yang dibangun dapat melakukan proses diagnosa hama dan penyakit berdasarkan gejala-gejala yang ada dan memberikan tata cara pengendaliannya.
- Jenis hama dan penyakit, gejala dan tata cara pengendaliannya disesuaikan dengan keterangan pakar, buku tentang hama dan penyakit tanaman, terutama mengenai hama dan penyakit pada tanaman mangga dan artikel pertanian di internet.

Tinjauan Pustaka

Hama yang sering mengganggu tanaman buah mangga antara lain penggerek pucuk yang disebabkan oleh larva Sternochetus Geniocnemis. Hama ini menyebabkan ranting tanaman mati kering karena jaringan pembuluh kayu rusak akibat gerakan larvanya. Sedangkan pengendaliannya dapat dilakukan penyemprotan dengan menggunakan insektisida kontak pada bagian ranting [1]. Tetapi, berdasarkan pengamatan kami dilapangan para petani belum mengetahui secara pasti hama penyakit yang menyerang tanaman mereka. Oleh sebab itu, kami berinisiatif membuat system pakar dengan menggunakan data yang dimiliki Puslitbang Hortikultura. Supaya para petani dapat menentukan secara pasti hama penyakit yang menyerang tanaman mereka.

Landasan Teori

Sistem pakar adalah program komputer yang didesain untuk meniru kemampuan memecahkan masalah dari seorang pakar. Pakar adalah orang yang memiliki kemampuan atau mengerti dalam menghadapi suatu masalah. Lewat pengalaman, seorang pakar mengembangkan kemampuan yang membuatnya dapat memecahkan permasalahan dengan hasil yang baik dan efisien [2].

Dalam aplikasi sistem pakar terdapat suatu metode untuk menyelesaikan masalah ketidakpastian data. Salah satu metode yang dapat digunakan adalah faktor kepastian (certainty factor). Faktor ketidakpastian diperkenalkan oleh Shortliffe Buchanan dalam pembuatan MYCIN (Wesley, 1984) [3].

Metode Forward Chaining adalah metode pencarian atau teknik pelacakan ke depan yang di mulai dengan informasi yang ada dan penggabungan rule untuk menghasilkan suatu kesimpulan atau tujuan [4].

2. Pembahasan

Dalam menghadapi suatu masalah sering ditemukan jawaban yang tidak memiliki kepastian penuh. Hal ini sangat mudah dilihat pada system pendeteksi penyakit, dimana pakar tidak dapat mendefinisikan tentang hubungan antara gejala dengan penyebabnya secara pasti. Certainty factor merupakan nilai parameter klinis yang diberikan MYCIN untuk menunjukkan besarnya kepastian dari suatu masalah.

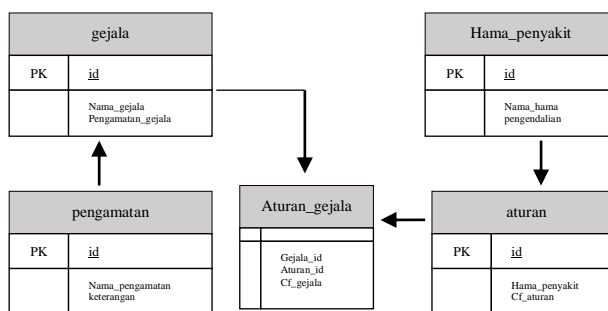
Metode inferensi tersebut dipengaruhi oleh tiga macam penelusuran, yaitu Depth-first search, Breadth-first search dan Best-first search.

- Depth-first search, melakukan penelusuran kaidah secara mendalam dari simpul akar bergerak menurun ke tingkat dalam yang berurutan.
- Breadth-first search, bergerak dari simpul akar, simpul yang ada pada setiap tingkat diuji sebelum pindah ke tingkat selanjutnya.
- Best-first search, bekerja berdasarkan kombinasi kedua metode sebelumnya. Dalam memilih apakah akan menggunakan pelacakan ke depan atau pelacakan ke belakang, semuanya bergantung masalah yang akan dibuat sistem pakar dan belum dapat dibuktikan mana yang lebih baik di antara kedua metode inferensi ini.

Dengan sistem ini, pengguna akan dapat mengetahui organisme pengganggu tanaman yang menyerang tanaman mangga dan tata cara pengendaliannya yang perlu dilakukan untuk menangani hama dan penyakit tersebut. Oleh karena itu, sistem ini harus memiliki spesifikasi sebagai berikut:

1. Mampu memberikan kesimpulan hama atau penyakit yang menyerang tanaman mangga .
2. Memberikan saran tata cara pengendalian yang harus dijalani oleh pengguna sistem ini berdasarkan hama dan penyakit yang menyerang.
3. Menggunakan mesin inferensi dengan metode penelusuran runut maju (forward chaining) dan faktor kepastian (certainty factor) untuk perhitungannya.
4. Mempunyai fasilitas akuisisi pengetahuan yang hanya dapat diakses oleh pakar.

Kebutuhan data ini direpresentasikan dalam ERD. Perancangan berbasis data tersebut diperlukan dalam merancang suatu system yang melibatkan basis data. Dapat dilihat pada gambar 2 diagram ERD tersebut.



Gambar 2 ERD

Pada penelitian ini, konsep basis data digunakan untuk mempresentasikan pengetahuan yang secara garis besar berisi gejala dan hama penyakit. Sistem yang dibangun memiliki tabel, yaitu:

1. Table Pengamatan

Table pengamatan digunakan untuk menyimpan data bagian tanaman yang diamati gejala-gejalanya akibat serangan hama dan penyakit.

Table 1. Table Pengamatan

No	Nama Field	Tipe Data	Ket
1	Id	Varchar	Primary Key
2	Nama_Pengamatan	Varchar	Nama Pengamatan

2. Table Gejala

Table ini berisi data-data mengenai gejala-gejala serangan hama dan penyakit dapat dilihat pada table 1.2.

Table 2. Table Gejala

Nama Field	Tipe data	Ket
Id	Integer	Primary key
Nama_gejala	Varchar	Nama gejala
Pengamatan_gejala	Varchar	Daerah pengamatan gejala

3. Table Hama dan Penyakit

Table ini berisi data-data Hama dan Penyakit dan cara pengendaliannya, dapat dilihat pada table 1.3.

Table 3. Table Hama dan Penyakit

Nama Field	Tipe Data	Ket
id	Varchar	Primary Key
Nama_Hama	Varchar	Nama Hama
Pengendalian	Varchar	Cara Pengendalian

4. Table Aturan

Table aturan terbentuk dari hama penyakit dan tingkat keyakinannya dapat dilihat pada table 1.4

Table 4. Table Aturan Nama Field

Tipe Data		Ket
Id	Varchar	Primary Key
id_Hama	Varchar	Foreign Key dari Tabel Hama
Cf_aturan	Varchar	

5. Tabel Aturan Gejala

Tabel ini berisi detail dari tabel aturan yaitu gejala-gejala tiap hama dan penyakit beserta tingkat keyakinannya, yang terdiri dari 3 field, yaitu: id_aturan, id_gejala dan cf_gejala seperti pada tabel 1.5 berikut

Table 5. Table Aturan Gejala Nama

Field	Tipe data	Ket
Aturan_id	Integer	Foreign key dari table_aturan
Gejala_id	Integer	Foreign key dari table_gejala
Cf_aturan	double	

6. Table Threshold

Table ini berisi bata nilai hama penyakit yang akan dimunculkan dalam syste, dapat dilihat pada table 1.6

Table 6. Table Threshold

Nama field	Tipe data	Ket
Id	Integer	
value	integer	Batas nilai

Proses Input Gejala

Input gejala oleh POPT merupakan proses awal sebelum dilakukan proses perhitungan nilai faktor kepastian terhadap suatu hama atau penyakit pada tanaman mangga untuk mendapatkan hasil diagnosa dan tata cara pengendaliannya. Dalam proses input gejala, semua gejala akan ditampilkan pada antarmuka diagnosa. Potongan kode untuk pengambilan data dari tabel Gejala untuk ditampilkan dapat dilihat pada gambar 3

```
public void getDataGejala() {
    //koneksi ke database
    java.sql.Connection conn = new Koneksi().connect();
    Object[] Data = {"ID", "Nama Gejala", "ID Pengamatan", "Daerah Pengamatan"};
    gejalaTableModel = new DefaultTableModel(conn, Data);

    try {
        String sql = "SELECT * FROM gejala";
        java.sql.Statement stmt = conn.createStatement();
        java.sql.ResultSet rset = stmt.executeQuery(sql);
        while (rset.next()) {
            String idGejala = rset.getString("ID");
            String nama = rset.getString("nama_gejala");
            String idPengamatan = rset.getString("pengamatan_gejala");
            String pengamatan = rset.getString("pengamatan_gejala");
            String[] dataField = {idGejala, nama, idPengamatan, pengamatan};
            gejalaTableModel.addRow(dataField);
        }
    } catch (Exception e) {}
}
```

Gambar 3 Kode Data Gejala

Setelah menampilkan data gejala pada antarmuka, POPT dapat memilih gejala-gejala pada antarmuka berdasarkan gejala yang timbul pada tanaman mangga berdasarkan daerah pengamatannya. Gejala-gejala yang telah dipilih POPT akan disimpan ke dalam larik untuk kemudian dijadikan sebagai masukan. Potongan kode untuk proses pemilihan gejala dapat dilihat pada gambar 4

```
public void pilihGejalaDiagnose() {
    boolean telahKda = false;
    for (int i = 0; i < diagnoseTableModel.getRowCount(); i++) {
        if (diagnosaTableModel.getValueAt(i, 0) == idGejalaLabel.getText()) {
            telahKda = true;
        }
    }
    if (!telahKda) {
        String[] dataField = {idGejalaLabel.getText(), namaGejalaLabel.getText(), cfField.getText()};
        diagnoseTableModel.addRow(dataField);
        diagnoseTableRef.setModel(diagnosaTableModel);
    }
}
```

Gambar 4 Kode Pemilihan Gejala

Proses Input Aturan

Input aturan oleh pakar merupakan proses manipulasi pengetahuan apabila ada pengetahuan baru mengenai suatu hama dan penyakit pada tanaman mangga. Antarmuka menampilkan data hama penyakit dan gejala. Apabila hama penyakit dan gejala yang akan ditambahkan tidak terdapat dalam basis data, maka pakar dapat menambahkan input aturan melalui antarmuka. Potongan kode untuk proses input aturan baru dapat dilihat pada gambar 5

```
String sql = "INSERT INTO aturan (nama_penyakit_id, cf_aturan) VALUES (?, ?)";
java.sql.PreparedStatement stmt = null;
try {
    . . .
}

String idAturan = "";
try {
    String sql = "SELECT * FROM 'aturan' order by id desc limit 1";
    java.sql.Statement stmt1 = conn.createStatement();
    java.sql.ResultSet rset1 = stmt1.executeQuery(sql);
    while (rset1.next()) {}
}
```

Gambar 5 Potongan Kode Tambah Aturan

Pakar dapat menentukan tingkat keyakinan tiap gejala yang berbeda-beda tiap aturannya, tergantung pada pengaruh gejala tersebut terhadap hama penyakit tersebut. Tingkat keyakinan tiap aturan ditentukan oleh pakar berdasarkan pemahaman pakar terhadap gejala-gejala yang terjadi dengan hama atau penyakitnya. Masukan-masukan dari aturan baru akan disimpan dalam

tabel aturan_gejala. Potongan kode untuk proses penyimpanan masukkan aturan baru dapat dilihat pada gambar 6

```
for (int i = 0; i < tabung1.tabung1.getRuleCount(); i++) {
    String idGejala = tabung1.tabung1.getValueAt(i, 0).toString();
    String aturanGejala = tabung1.tabung1.getValueAt(i, 1).toString();

    String sql2 = "INSERT INTO aturan_gejala (gejala_id,aturan_id, cf_gejala) VALUES (?, ?, ?)";
    java.sql.PreparedStatement stmt2 = null;
    try {
        stmt2 = conn.prepareStatement(sql2);
    } catch (SQLException ex) {}
    try {
        stmt2.setString(1, idGejala);
        stmt2.setString(2, idAturan);
        stmt2.setString(3, cfGejala);
        stmt2.executeUpdate();
    } catch (SQLException ex) {}
}
```

Gambar 6 Proses Memasukkan Data Table

Proses Input Hama Penyakit

Input hama penyakit dilakukan oleh pakar, ketika sebuah penyakit baru ditemukan. Antarmuka menampilkan data hama penyakit dan tata cara pengendalian. Apabila hama penyakit dan tata cara pengendalian tidak ada dalam basis data, maka pakar akan menambahkan hama penyakit baru melalui antarmuka dan akan tersimpan dalam tabel 1.4. Potongan kode untuk proses proses input hama penyakit baru dapat dilihat pada gambar 7

```
public void tambahPenyakit() {
    //panggil koneksi
    java.sql.Connection conn = new Koneksi().connect();

    String sql = "INSERT INTO hama_penyakit (nama_hamaPenyakit, pengendalian) VALUES (?, ?)";
    java.sql.PreparedStatement stmt = null;
    try {
        stmt = conn.prepareStatement(sql);
    } catch (SQLException ex) {}
}
```

Gambar 7 Kode Tambah Hama Penyakit

Proses Diagnosa

Proses diagnose merupakan proses mesin inferensi bekerja dalam system pakar ini. Pada proses ini dilakukan perhitungan *certainty factor* setelah POPT memberikan masukkan gejala-gejala yang dialami tanaman mangga. Hasil masukkan POPT diproses dengan mencocokkan aturan yang mencukupi, dapat dilihat pada table 1.5

```
public static String[][] aturanGejalaSetuArray(String a) {
    java.sql.Connection conn = new Koneksi().connect();
    int j = 0;
    try {
        String sql = "SELECT * FROM 'aturan_gejala' WHERE 'aturan_id' = ?";
        java.sql.Statement stmt = conn.createStatement();
        java.sql.ResultSet rs = stmt.executeQuery(sql);
        while (rs.next()) {
            j = j + 1;
        }
    } catch (Exception ex) {}

    String[][] aturanGejalaSetu = new String[j][2];
    try {
        String sql = "SELECT * FROM 'aturan_gejala' WHERE 'aturan_id' = ?";
        java.sql.Statement stmt = conn.createStatement();
        java.sql.ResultSet rs = stmt.executeQuery(sql);
        j = 0;
        while (rs.next()) {
            String idGejala = rs.getString("gejala_id");
            String cfGejala = rs.getString("cf_gejala");
            aturanGejalaSetu[j][0] = idGejala;
            aturanGejalaSetu[j][1] = cfGejala;
            aturanGejalaSetu[j][2] = "?";
            j = j + 1;
        }
    }
}
```

Gambar 8 kode Proses pencocokan atruan gejala

Proses *certainty factor* setelah gejala dimasukkan oleh POPT dapat dilihat pada gambar 8. Hasil hama atau penyakit yang muncul adalah hasil tertinggi pada perhitungan *certainty factor*.

```
for (int i = 0; i < aturanGejalaSetuArray(a).length; i++) {
    String[] hama = aturanGejalaSetuArray(a).get(i);
    hama[i][2] = Integer.parseInt(hama[i][2]);
    hama[i][3] = 100;

    if (checkKekru(temp, tabung1.tabung1)) {
        for (int j = 0; j < temp.tabung1.tabung1; j++) {
            for (int k = 0; k < tabung1.tabung1.tabung1; k++) {
                if (temp[i][0].equals(tabung1.tabung1[k][1]) && hama[i][1].equals(tabung1.tabung1[k][2])) {
                    hama[i][3] = Integer.parseInt(hama[i][3]) * Integer.parseInt(temp[i][1]) / 100;
                }
            }
        }
        hama[i][3] = hama[i][3] * Integer.parseInt(hama[i][3]) / 100;
        hama[i][3] = Integer.parseInt(hama[i][3]);
        System.out.println("gejala" + hama[i][1]);
        System.out.println("gejala" + hama[i][2]);
        System.out.println("gejala" + hama[i][3]);
    }

    for (int k = 0; k < temp.tabung1.tabung1; k++) {
        if (Integer.parseInt(hama[i][3]) > k) {
            k = Integer.parseInt(hama[i][3]);
            System.out.println("hasil" + k + " * " + k);
        }
    }
}
```

Gambar 9 Code perhitungan certainty factor

Gambar 9 memperlihatkan perhitungan *certainty factor* jika tidak ada hama penyakit yang sama dalam aturan yang diproses. Jika terdapat aturan yang memuat hama dan penyakit yang sama maka akan dihitung dengan CF combine seperti pada gambar 10

```
int cf = Integer.parseInt(hama[i][3]) * Integer.parseInt(hama[i][3]);
if (cf > 100) {
    cf = 100;
}
System.out.println("cf" + cf);
hama[i][3] = cf / 100;

System.out.println("hama" + hama[i][1] + " * " + hama[i][2] + " * " + cf);
hama[i][3] = cf;
```

Gambar 10 Code perhitungan CF Combine

Setelah melakukan proses perhitungan *certainty factor*, sistem akan menampilkan hama penyakit yang memiliki nilai *certainty factor* tertinggi di atas *threshold* dan jika nilai *certainty factor* lebih rendah dari *threshold* maka tidak ada hama penyakit yang muncul dalam sistem. Nilai *threshold* diambil dari table 1.6 Gambar 11 merupakan potongan code perhitungan *certainty factor* terhadap *threshold*.

```
String threshold = "?";
try {
    String sql = "SELECT * FROM 'threshold'";
    java.sql.Statement stmt = conn.createStatement();
    java.sql.ResultSet rs = stmt.executeQuery(sql);
    while (rs.next()) {
        threshold = rs.getString("threshold");
    }
} catch (SQLException ex) {}

String hamaPenyakit = "Hama Hama";
String hasilKalkulasi = "Hasil Kalkulasi";
String hamaPenyakit = "Hama Hama";

for (int i = 0; i < hamaPenyakit.length; i++) {
    System.out.println("hamaPenyakit" + hamaPenyakit[i]);
    for (int j = 0; j < hamaPenyakit.length; j++) {
        System.out.println("hamaPenyakit" + hamaPenyakit[j]);
        System.out.println("hamaPenyakit" + hamaPenyakit[i]);
    }
}
```

Gambar 11 Perhitungan Threshold

Proses Pengisian Basis Pengetahuan

Tahap awal dari penggunaan sistem adalah proses pengisian basis pengetahuan. Nilai faktor kepastian diberikan antara 0 sampai dengan 1. Pengisian faktor kepastian gejala terhadap hama atau penyakit dan dilakukan oleh seorang pakar ketika melakukan proses manipulasi pengetahuan dalam bentuk persen.

Proses Diagnosa

Diagnosa hama dan penyakit dilakukan dengan memberikan masukkan berupa gejala-gejala yang timbul pada tanaman mangga. Ketika gejala dimasukkan, sistem akan secara otomatis mencari hama atau penyakit yang memiliki faktor kepastian dengan nilai tertinggi dan lebih dari *threshold* untuk kemudian menampilkannya pada bagian hasil diagnosa.

Penjelasan Perhitungan Mesin Inferensi

Dalam proses perhitungan mesin inferensi, perhitungan untuk menentukan hasil diagnosa menggunakan persamaan.

Perhitungan Mesin Inferensi

Pengujian sistem dilakukan dengan data hama penyakit yang terdapat dalam basis data pada tabel 1.1 dan dengan data gejala yang terdapat pada tabel 1.3. Uji coba sistem dilakukan dengan mendiagnosa suatu penyakit dengan gejala yang telah terdapat dalam basis data. Rincian tabel keputusan terhadap suatu penyakit dapat dilihat pada tabel 1.4.

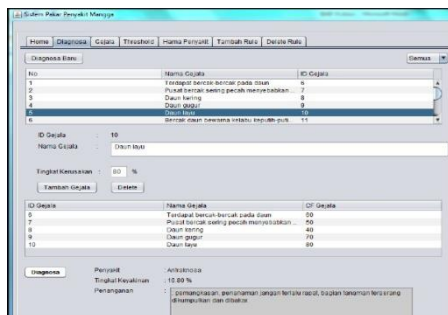
Uji coba sistem dilakukan dengan mendiagnosa suatu hama penyakit dengan terdapat bercak-bercak pada daun, pusat bercak sering pecah menyebabkan bercak berlubang, daun kering daun gugur dan daun layu. Untuk perhitungan manual mesin inferensi dalam pencarian solusi dapat dilihat pada tabel 1.1

Tabel 7. Pengujian Perhitungan

H P	G ej	CFp akar	CF pop t	CFa tura n	CFpakar *CFpop t	Min[CFpak ar*CFpop t] *CFaturan
1	1	0.7	0.6	0.7	0.42	Min[0.42;0.35;0.25;0.42;0.4]* CFaturan =0.24*0.7 =0.168
1	2	0.7	0.5		0.35	
1	3	0.6	0.4		0.24	
1	4	0.6	0.7		0.42	
1	5	0.5	0.8		0.4	

Hasil Uji Coba

Pada gambar 12 menunjukkan hasil diagnosa dari pengujian sistem. Sistem hanya akan menampilkan solusi dengan *certainty factor* yang paling besar serta melebihi atau memiliki selisih yang minimum dari *threshold*. Hasil pengujian sistem yang memiliki solusi yaitu Antraknosa dengan nilai *certainty factor* 0,168. Hal ini cocok dengan perhitungan manual pada tabel 1.1.



Gambar 12 Hasil Pengujian Diagnosa.

Penambahan gejala baru

Pada gambar 13 menunjukkan bahwa pakar dapat menambah gejala baru, menghapus dan mengedit gejala.



Gambar 13 Proses Penambahan Gejala Baru

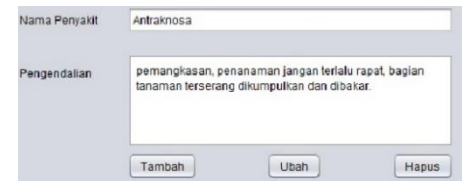
Setelah di klik *button* tambah maka data akan disimpan dan dapat dilihat pada antarmuka daftar gejala yang ada, seperti pada gambar 14

ID	Nama Gejala	ID Pengamatan	Bagian Pengam...
46	Akar busuk	3	Akar
47	Tunas Kerdil	6	Lain-lain
48	Terbentuknya mi...	6	Lain-lain
49	Tumbuhan terha...	6	Lain-lain
50	Tanaman layu	6	Lain-lain
51	Tanaman mati	6	Lain-lain

Gambar 14 Tampilan Gejala Update

Penambahan hama penyakit baru

Pada gambar 15 menunjukkan bahwa pakar dapat menambah hama penyakit, *certainty factor* dan tata cara pengendalian baru, dapat melakukan perubahan terhadap data hama dan penyakit, *certainty factor* dan tata cara pengendalian, serta dapat melakukan penghapusan data yang ada.



Gambar 15 Proses Penambahan Hama Penyakit Baru

Setelah di klik *button* tambah maka data akan disimpan dan dapat dilihat pada antarmuka daftar hama penyakit yang ada, seperti pada gambar 16

ID	Nama Penyakit/Hama	Pengendalian
4	Antraknosa	pemangkasan, penan...
5	Bercak Daun Kelabu	Pemeliharaan tanam...
6	Lalat Buah	Memusnahkan buah y...
7	Penyakit kulit Belendok	Cara kultur teknis yaitu...
8	Wereng Mangga	Pengendalian kimiawi ...
9	Penggerek Cabang	Memanfaatkan parasit...

Gambar 16 Tampilan Hama Penyakit.

Penambahan aturan baru

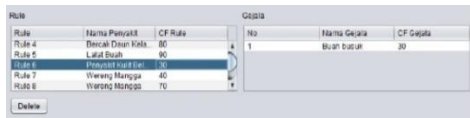
Pada gambar 17 menunjukkan bahwa pakar dapat menambah aturan baru.



Gambar 17 Proses Penambahan Aturan Baru

Penghapusan aturan

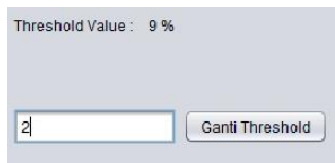
Pada gambar 18 menunjukkan bahwa pakar dapat melakukan aksi penghapusan yang telah ada



Gambar 18 Proses Penghapusan Aturan

Nilai threshold

Threshold merupakan nilai batas yang parameter kerjanya ditentukan untuk mengetahui kepastian suatu gejala penyakit. Untuk kecocokan hama penyakit jika parameternya melebihi nilai threshold yang sudah ditentukan Pada gambar 2.8 menunjukkan bahwa pakar dapat menetapkan *threshold* untuk menampilkan hasil diagnosa.



Gambar 19 Penetapan Nilai Threshold

Setelah di klik *button* update maka nilai *threshold* yang lama akan digantikan dengan nilai yang baru dan dapat dilihat pada antarmuka *threshold*, seperti pada gambar 20.



Gambar 20 Update Threshold

Tampilan pada POPT

POPT dapat memberikan masukkan berupa gejala-gejala yang timbul pada tanaman mangga yang akan diproses oleh sistem untuk menghasilkan diagnosa berdasarkan perhitungan *certainty factor*.

Input gejala

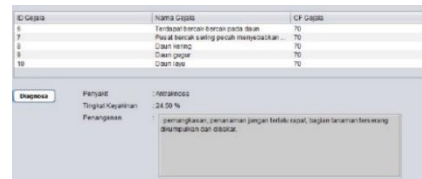
Pada gambar 21 menunjukkan bahwa POPT memberikan masukkan berupa gejala dan *certainty factor* pada system.



Gambar 21 Input Gejala

Hasil Diagnosa

Pada gambar 22 menunjukkan hasil diagnosa dan perhitungan *certainty factor* berdasarkan inputan gejala POPT.



Gambar 22 Hasil Diagnosa

Berdasarkan hasil pengujian Sistem Pakar Diagnosa Hama dan Penyakit pada Tanaman Mangga dapat ditarik kesimpulan.

3. Kesimpulan

1. Setiap *rule* yang ada dalam sistem memiliki tingkat keyakinan yang berbeda-beda, tergantung tingkat keyakinan pakar terhadap gejala-gejala yang mempengaruhi suatu penyakit.
2. Nilai tingkat keyakinan gejala pada proses diagnosa dapat ditentukan secara berbeda-beda oleh *popt*, sehingga dalam proses pendagnosaan menjadi lebih teliti dalam menentukan hasil diagnosa..
3. Sistem dapat memberikan hasil diagnosa berupa hama atau penyakit yang kemungkinan dialami tanaman dan tata cara pengendaliannya.

Daftar Pustaka

- [1] Puslitbang Hortikultura, “Hama dan Penyakit Penting Tanaman Mangga”, 2009.
- [2] Durkin, John; “Expert Systems Design and Development”, Prentice Hall, 1994.
- [3] Kusrini, M.Kom, “APLIKASI SISTEM PAKAR Menentukan Faktor Kepastian Pengguna dengan Metode Kuantifikasi Pertanyaan”, 2008.
- [4] Hayes-Roth, Frederick; Donald Waterman; Douglas Lenat, “Building Expert Systems”, 1983.

Biodata Penulis

Gatot Fitriyanto, saat ini sedang menempuh pendidikan Sarjana pada Jurusan Teknik Informatika di STMIK AMIKOM Yogyakarta.

Nur Ahmad Azhar, saat ini sedang menempuh pendidikan Sarjana pada Jurusan Teknik Informatika di STMIK AMIKOM Yogyakarta.

Muhammad Kurniawan, saat ini sedang menempuh pendidikan Sarjana pada Jurusan Teknik Informatika di STMIK AMIKOM Yogyakarta