

IMPLEMENTASI CASE BASED REASONING UNTUK SISTEM DIAGNOSIS HAMA DAN PENYAKIT TANAMAN CABE MERAH MENGGUNAKAN ALGORITMA SIMILARITAS NEYMAN

Arno Reza Pahlawan¹, Setyawan Wibisono²

^{1,2}Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Stikubank
e-mail: ¹arno.reza@gmail.com, ²setyawan@edu.unisbank.ac.id

ABSTRAK

Cabai merah adalah tanaman dengan intensitas gangguan yang cukup tinggi, berupa hama ataupun penyakit yang dapat membuat tanaman rusak maupun gagal panen. Hama dan penyakit adalah faktor yang dapat menurunkan hasil produksi. Minimnya pengetahuan masyarakat dan petani cabai merah tentang hama dan penyakit serta mahalnya biaya untuk penyuluhan membuat penanganan gangguan menjadi keliru. Untuk itu dibuatlah program yang mengadopsi pola berfikir manusia yang sering dinamakan *artificial intelligence*.

Sistem yang dibuat mampu mengidentifikasi hama dan penyakit tanaman cabai merah dengan memasukkan gejala, kemudian menghitung nilai similaritas menggunakan algoritma similaritas Neyman berdasarkan perbandingan kesamaan gejala yang dikonsultasikan dengan data gejala yang sudah ada dalam basisdata. Solusi penanganan penyakit yang diderita diberikan setelah mengetahui jenis hama dan penyakit dari hasil identifikasi menggunakan algoritma similaritas Neyman, dengan menampilkan hasil identifikasi hama dan penyakit tanaman cabai merah. Apabila hasil similaritas lebih dari sama dengan 50 persen, maka akan ditampilkan hasil diagnosis, namun jika hasil similaritas kurang dari 50 persen, maka akan dimasukkan dalam tahap *revise* agar dapat dijadikan sebagai sebuah kasus baru.

Kata Kunci: Hama dan Penyakit Tanaman Cabai Merah, Case Based Reasoning, Algoritma Similaritas Neyman

1. PENDAHULUAN

Tanaman cabai merah dalam proses budidayanya seringkali mengalami gangguan berupa hama ataupun penyakit yang dapat membuat tanaman rusak ataupun mati. Dalam mengatasi penyakit diperlukan langkah khusus berupa pengobatan dan terapi terhadap tanaman yang terjangkit penyakit. Kurangnya pemahaman dalam penanganan gangguan pada tanaman cabai merah seringkali membuat petani merugi. Untuk itu diperlukan sebuah penyuluhan dari seorang pakar pertanian untuk membantu memberikan informasi dan pemahaman kepada petani yang membutuhkan. Hal inilah yang mendorong dibangunnya sistem pakar yang bertujuan mempermudah penyampaian informasi untuk dapat mengatasi permasalahan penyakit pada tanaman cabai merah dengan menerapkan sistem pakar menggunakan metode *Case Based Reasoning* (CBR) dengan algoritma similaritas Neyman.

Metode CBR adalah metode yang digunakan untuk menyelesaikan suatu masalah dengan menggunakan pengalaman sebelumnya guna menyelesaikan suatu masalah baru dengan tetap memperhatikan aspek kesamaan antara satu dengan beberapa penyelesaian dari permasalahan-permasalahan sebelumnya serta dibangunnya sebuah sistem basisdata untuk menyimpan revisi terhadap suatu solusi permasalahan baru [1].

Dalam banyak pendekatan klasifikasi biner, pengklasifikasi dirancang untuk meminimalkan kesalahan dengan pengujian hipotesis biner, dan seseorang harus memutuskan antara hipotesis nol dan alternatif hipotesis. Sebuah tingkat pengujian dikenakan pada probabilitas *false alarm* dan tetap mencari tes yang memenuhi hambatan ini sambil memperkecil kehilangan probabilitas serta memaksimalkan ekuivalen deteksi probabilitas. Untuk itu kami memilih satu pendekatan untuk klasifikasi yaitu algoritma similaritas Neyman. Pendekatan yang dominan untuk desain klasifikasi dalam studi telah meminimalkan kemungkinan terjadinya kesalahan. Namun jelas bahwa kegagalan untuk mendeteksi memiliki konsekuensi yang drastis. Dalam pendekatan Neyman, prosedur diagnostik melibatkan pengaturan tingkat pengujian guna membangun klasifikasi untuk meminimalkan jumlah kasus tidak terjawab benar.

Sistem pakar untuk hama dan penyakit tanaman cabai merah menggunakan metode *Case Based Reasoning* dengan algoritma similaritas Neyman dapat menjadi program bantu untuk mendukung keputusan menentukan hama dan penyakit tanaman cabai merah.

2. TINJAUAN PUSTAKA

Penelitian pemanfaatan CBR dalam diagnosa penyakit telah dilakukan untuk penyakit mata. Metode CBR dapat membantu dalam mengambil keputusan terkait dengan penyakit mata pada manusia. Perlu dipahami bahwa sistem CBR tidak menjamin solusi yang direkomendasikan adalah solusi yang terbaik, karena karena CBR hanya memberikan solusi berdasarkan kasus-kasus yang disimpan. Kelengkapan dan kekompleksan kasus yang tersimpan dalam basis kasus dapat menjadikan CBR suatu sistem yang dapat menghasilkan solusi optimal dan tepat [2].

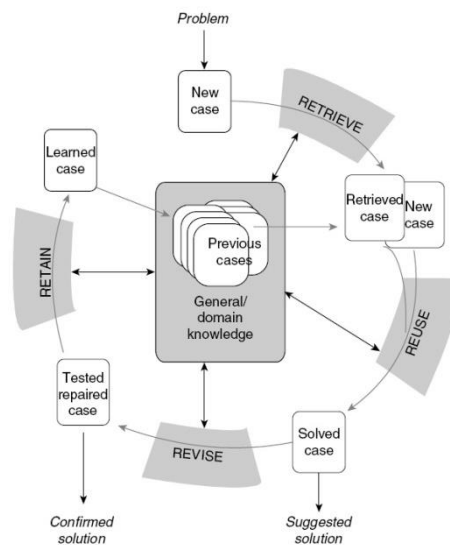
Dengan memanfaatkan metode *Case Based Reasoning*, dapat dihasilkan suatu aplikasi untuk mengidentifikasi pertumbuhan balita. Dengan sistem ini nantinya dapat digunakan sebagai sarana atau sebagai

pengetahuan dalam menjaga kestabilan pertumbuhan balita dan membantu anda untuk mendapatkan hasil yang optimal dalam menjaga pertumbuhan setiap balita. Metode *Case Based Reasoning* digunakan dalam aplikasi pertumbuhan balita dengan menggunakan perhitungan *Nearest Neighbor*, dimana data kasus baru akan dibandingkan perhitungannya dengan data kasus lama yang ada di basisdata, dan kemudian dihitung kriteria kemiripannya berdasarkan rumus atau ketentuan yang berlaku [3].

Salah satu pendekatan untuk klasifikasi dalam konteks ini terinspirasi pengujian hipotesis klasik Neyman-Pearson dengan pendekatan klasifikasi biner, pengklasifikasi dirancang untuk meminimalkan kesalahan dengan pengujian hipotesis biner. Untuk memotivasi pendekatan Neyman guna belajar klasifikasi, mempertimbangkan masalah pengelompokan tumor menggunakan *microarray* ekspresi gen yang biasanya digunakan adalah sebagai berikut: pertama mengidentifikasi status beberapa pasien guna mengetahui bentuk status dari kanker. Selanjutnya mengumpulkan sampel jaringan sel darah dari setiap pasien. Kemudian melakukan eksperimen *microarray* untuk menilai banyaknya relatif dari berbagai gen dalam setiap subyek. Akhirnya, data pelatihan ini digunakan untuk membangun klasifikasi yang pada prinsipnya dapat digunakan untuk mendiagnosa profile ekspresi gen para pasien dimasa depan. Neyman secara eksklusif hanya fokus pada klasifikasi biner, meskipun ekstensi untuk multi-kelas mungkin saja terjadi. Misalkan ditetapkan variabel X dan membiarkan $Z=(X,Y)$ mengacak variabel menggunakan pengambilan nilai variabel $Z=(X \times \{0,1\})$. Variabel X sesuai dengan tanda yang diamati (pola, vektor fitur) dan Y adalah label kelas yang terkait dengan X . Dalam pengujian klasik Neyman $Y=0$ sesuai dengan hipotesis nol [4].

2.1. Case Based Reasoning (CBR)

Cara kerja CBR adalah dengan membandingkan kasus baru dengan kasus lama. Jika kasus baru tersebut mempunyai kemiripan dengan kasus lama maka CBR akan memberikan solusi kasus lama untuk kasus baru tersebut [5]. Jika tidak ada yang cocok, CBR akan melakukan proses adaptasi, yakni memperbaiki pengetahuan lama agar sesuai untuk menyelesaikan kasus baru. Kemudian pengetahuan baru akan disimpan sebagai salah satu basis kasus [6]. CBR menggunakan pendekatan kecerdasan buatan yang menitikberatkan pemecahan masalah berdasarkan pengetahuan dari kasus-kasus sebelumnya, seperti terlihat pada gambar 1.



Gambar 1. Siklus Case Based Reasoning [1]

Secara umum metode ini memiliki 4 langkah yaitu [6]:

- e. *Retrieve*, memperoleh kembali kasus-kasus terdahulu yang pernah terjadi atau paling mirip.
- f. *Reuse*, menggunakan informasi dan pengetahuan dari sebuah kasus sebagai dasar untuk memecahkan masalah.
- g. *Revise*, meninjau kembali atau mengevaluasi pemecahan masalah yang dihasilkan proses reuse, jika berhasil maka dilanjutkan ke proses *retain*.
- h. *Retain*, menyimpan bagian-bagian pengalaman tersebut guna memecahkan masalah di masa mendatang dan selanjutnya solusi baru akan disimpan ke dalam basis pengetahuan.

2.2. Indexing

Indexing adalah proses pengelompokan kasus untuk mengefisienkan waktu dalam melakukan proses pencarian kasus yang ada pada sistem. Sistem *Case Based Reasoning* hanya menghitung nilai kemiripan kasus

yang berada pada kelompok yang sama. Pada penelitian ini metode yang digunakan untuk *indexing* adalah metode *K-Nearest Neighbor* [5].

2.3. Algoritma Similaritas Neyman

Pendekatan Neyman digunakan terutama untuk masalah deteksi biner. Dalam banyak pendekatan klasifikasi biner, pengklasifikasi dirancang untuk meminimalkan kesalahan dengan pengujian hipotesis biner, dan seseorang harus memutuskan antara hipotesis nol (H_0) dan alternatif hipotesis (H_1). Hanya salah satu hipotesis ini menguraikan benar. Ruang keputusan (H_0), (H_1), masing-masing mewakili menerima H_0 dan menolak H_0 . Fokusnya adalah pada probabilitas kondisional. Untuk memaksimalkan probabilitas memilih H_1 padahal sebenarnya H_1 adalah benar, meskipun tidak melebihi probabilitas tetap memilih H_1 ketika itu tidak benar. Sebuah tingkat pengujian dikenakan pada probabilitas *false alarm*. Probabilitas menolak hipotesis nol (H_0) ketika itu benar disebut dengan *false alarm*, dan tetap mencari tes yang memenuhi hambatan ini sambil memperkecil kehilangan probabilitas serta memaksimalkan ekuivalen deteksi probabilitas. Kekuatan deteksi probabilitas, dari aturan pengambilan keputusan adalah probabilitas benar menerima hipotesis alternatif (H_1) ketika itu benar [7].

Probabilitas untuk menerima (H_0) ketika (H_1) benar menghasilkan deteksi dilewatkan. Pendekatan yang dominan untuk desain klasifikasi dalam studi telah meminimalkan kemungkinan terjadinya kesalahan. Namun jelas bahwa kegagalan untuk mendeteksi memiliki konsekuensi yang drastis. Dalam pendekatan Neyman, prosedur diagnostik melibatkan pengaturan tingkat pengujian guna membangun klasifikasi untuk meminimalkan jumlah kasus tidak terjawab benar. Fungsi rumus algoritma jarak Neyman diformulasikan dalam persamaan 1, kemudian dihitung menggunakan persamaan seperti terlihat pada persamaan 2. [7].

$$dN(P, Q) = \sum_{i=1}^d \frac{(P_i - Q_i)^2}{P_i} \quad (1)$$

$$S = 1 - \sum_{i=1}^d \frac{(P_i - Q_i)^2}{P_i} \quad (2)$$

Keterangan :

P = Nilai *distance* / jarak proses *indexing*

Q = Nilai *distance* / jarak kasus target

d = jumlah atribut dalam setiap kasus

i = atribut individu antara 1 s.d. n

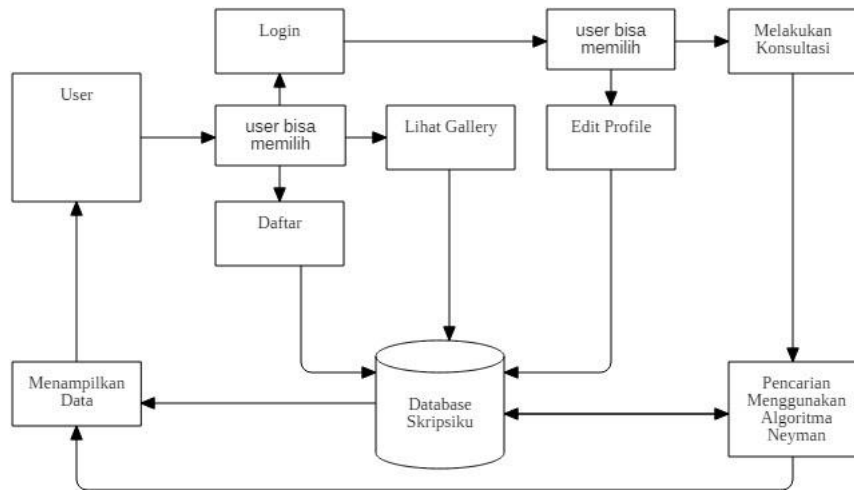
S = Nilai similaritas

3. METODE PENELITIAN

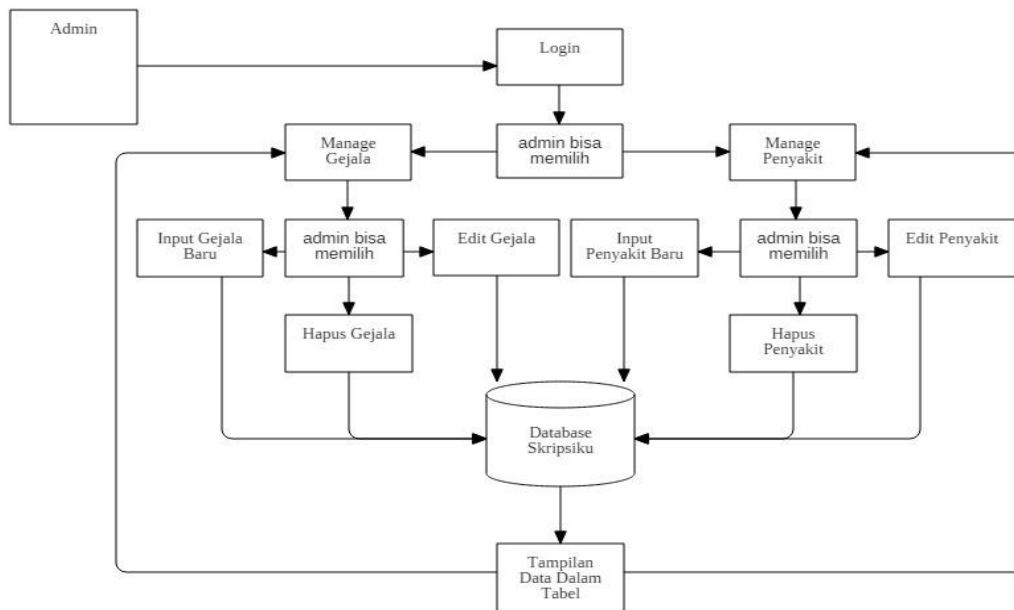
Sistem pakar untuk hama dan penyakit tanaman cabai merah menggunakan metode *Case Based Reasoning* dengan algoritma similaritas Neyman merupakan aplikasi berbasis web yang berisi informasi tentang hama dan penyakit tanaman cabai merah. Proses kerja sistem web ini dibedakan menjadi 2 bagian antara pengguna dan admin. Untuk pengguna yang tidak melakukan *login* hanya memiliki akses sebatas melihat informasi tentang hama dan penyakit tanaman cabai merah. Untuk dapat memiliki hak akses penuh sebagai pengguna haruslah melakukan proses *login* terlebih dahulu, apabila belum memiliki akun diharuskan melakukan registrasi agar dapat melakukan proses *login*. Bagi pengguna yang telah *login* tidak hanya memiliki hak akses melihat informasi saja tetapi juga dapat melakukan pencarian hama dan penyakit tanaman cabai merah dengan melakukan *input* gejala.

Pada proses *revise* sistem akan memperbaiki/meninjau kembali solusi yang telah dihasilkan pada proses *reuse*, jika hasil perhitungan nilai similaritas kurang dari 50% maka sistem tidak akan memberikan saran hama dan penyakit tanaman cabai merah. Informasi berupa gejala yang tidak memenuhi syarat minimal akan disimpan ke dalam tabel *revise* untuk diperbaiki kembali oleh pakar untuk ditemukan solusinya. Solusi yang telah ditemukan oleh pakar pada proses *revise* inilah yang akan digunakan untuk kasus berikutnya untuk permasalahan yang sama. Proses inilah yang disebut *retain*. pengguna dapat memperoleh informasi tentang hama dan penyakit tanaman cabai merah dengan cara memilih gambar pada halaman galeri kemudian sistem akan mencari dan menampilkan informasi tersebut.

Untuk admin proses yang harus dilakukan adalah melakukan *login* pada halaman admin untuk dapat memiliki hak akses sebagai administrator. Apabila telah melakukan *login* maka akan muncul halaman *home* admin yang berisikan menu gejala, penyakit, dan *revise*. Untuk halaman gejala dan penyakit admin dapat melakukan aktivitas *create*, *read*, *update*, *delete* data gejala dan penyakit. Untuk menyimpan data dari *form input* dilanjutkan dengan kalimat *query insert*. Untuk mengedit data yang telah dipilih dari tabel data dilanjutkan dengan kalimat *query update*. Untuk proses hapus data dilanjutkan dengan kalimat *query delete*. Dan untuk menampilkan data dilanjutkan dengan kalimat *query select*. Alur kerja pengguna dan admin adalah sebagai berikut:



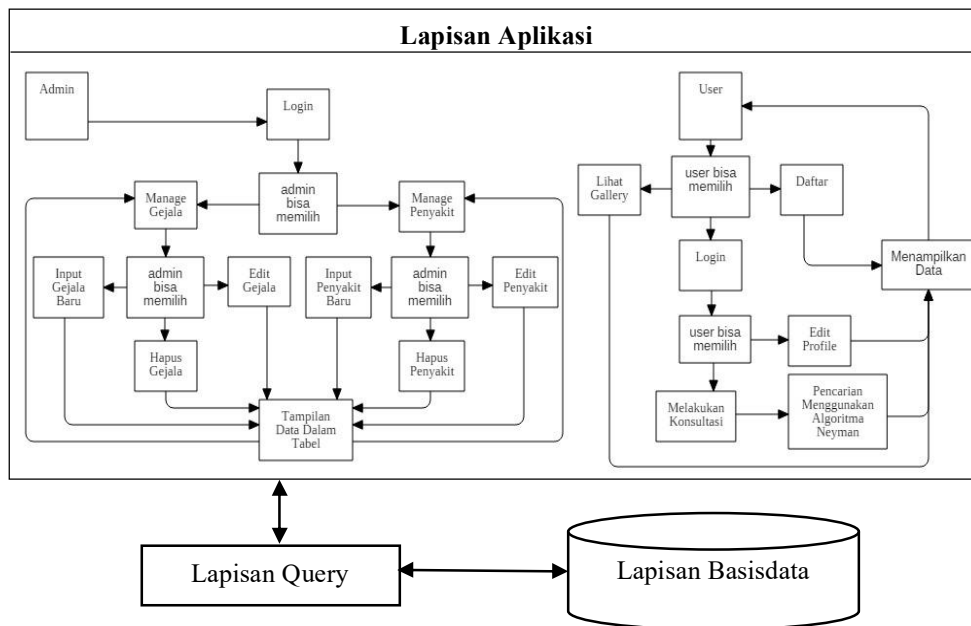
Gambar 2. Alur Kerja Pengguna



Gambar 3. Alur Kerja Admin

a. Arsitektur Sistem

Sistem pakar untuk hama dan penyakit tanaman cabai merah menggunakan metode *Case Based Reasoning* dengan algoritma similaritas Neyman berbasis Web ini memiliki lapisan basisdata yang berfungsi sebagai tempat penyimpanan data gejala dan penyakit, aplikasi sebagai media penyimpanan ini menggunakan XAMPP v3.2.1 dengan format data SQL. Untuk memisahkan *variabel* data yang akan dirubah dalam format SQL digunakan bahasa pemrograman PHP sebagai lapisan *query*. Lapisan aplikasi antarmuka juga merupakan hal yang sangat penting karena berfungsi untuk memudahkan pengguna untuk melakukan hubungan interaksi dengan sistem.



Gambar 4. Arsitektur Sistem

b. Analisis Data Penyakit dan Gejala

Basis pengetahuan berupa data gejala, data penyakit, gambar tentang tanaman yang terserang penyakit, dan solusi penanganan dari penyakit tersebut. Data-data tersebut dapat diperoleh dari hasil wawancara, jurnal, buku, dan berbagai macam sumber lainnya yang berkaitan tentang penyakit tanaman cabai merah. Dari data tersebut dapat direpresentasikan menjadi bentuk yang sistematis dengan pengkodean dan digambarkan dalam bentuk rancangan berupa tabel jenis hama dan penyakit seperti terlihat pada tabel 1.

Tabel 1. Jenis Hama dan Penyakit

Kode	Jenis Hama dan Penyakit
P01	Hama tungau kuning (Polyphagotarsonemus latus banks)
P02	Hama trips (thrips parvispinus karny)
P03	Hama kutu daun persik (myzus persicae sulz)
P04	Hama kutu kebul (bemisia tabaci)
P05	Hama lalat buah (dacus dorsalis)
P06	Hama tikus (rattus argentiventer)
P07	Penyakit layu fusarium (fusarium oxysporum)
P08	Penyakit layu bakteri (ralstonia pseudomonas solanacearum)
P09	Penyakit bercak daun (cercospora capsici)
P10	Penyakit antraknosa / penyakit patek (colletotrichum capsici)
P11	Penyakit virus kerupuk (luteo virus)
P12	Penyakit virus mosaik (tobacco mosaic virus)
P13	Penyakit virus kuning (gemini virus)
P14	Penyakit fisiologis
P15	Penyakit rebah semai
P16	Penyakit embun tepung

Tabel gejala adalah tabel yang akan digunakan sebagai basis pengetahuan yang dicatat dalam basisdata. Tabel gejala ini dibandingkan dengan gejala-gejala, baik gejala yang sudah terdapat dalam basisdata maupun gejala yang belum ada di dalam basisdata. Jumlah gejala penyakit lama akan dibandingkan dengan jumlah gejala penyakit baru yang dikonsultasikan oleh pengguna, jumlah gejala yang sama akan dihitung menggunakan algoritma similarias Neyman untuk menentukan jenis hama dan penyakit yang diderita oleh tanaman cabai merah. Pada tabel gejala terdapat dua buah data yaitu data kode gejala yang menunjukkan urutan kode gejala dan data gejala hama dan penyakit tanaman cabai merah seperti terlihat pada tabel 2.

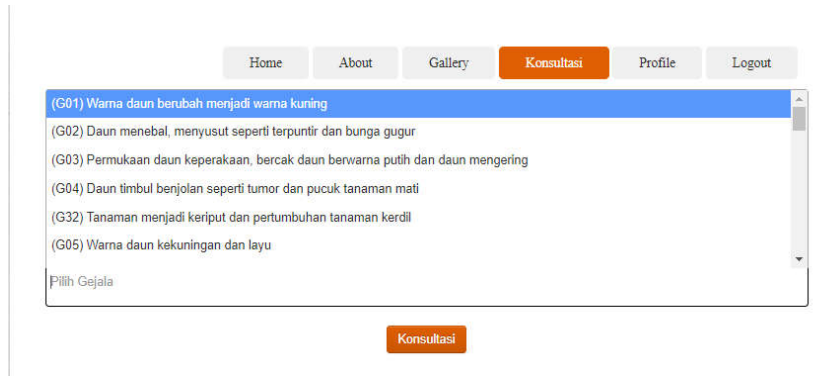
Tabel 2. Gejala

Kode	Gejala
G01	Warna daun berubah menjadi warna kuning
G02	Daun menebal, menyusut seperti terpuntir dan bunga gugur

G03	Permukaan daun keperakaan, bercak daun berwarna putih dan daun mengering
G04	Daun timbul benjolan seperti tumor dan pucuk tanaman mati
G05	Warna daun kekuningan dan layu
G06	Daun muda layu secara mendadak
G07	Sel-sel dan jaringan daun rusak
G08	Terdapat bercak bulat dan kering berwarna pucat sampai putih pada daun
G09	Terdapat bercak pada daun menyerupai mata kodok
G10	Daun menguning dan terdapat bercak pada batang
G11	Daun melengkung kebawah, berwarna hijau pekat, permukaanya tidak rata
G12	Daun terkesan regas seperti kerupuk
G13	Warna daun belang, daun cekung, daun mengering
G14	Bentuk daun menyempit dan memanjang
G15	Daun berwarna kuning jelas, tulang daun menebal
G16	Daun menggulung keatas dan menyusut, tanaman kerdil
G17	Pada sisi bawah daun terdapat lapisan tepung berwarna putih
G18	Daun menjadi pucat dan cepat rontok
G19	Buah berlubang dan membusuk
G20	Terdapat titik coklat kehitaman pada pangkal buah, buah busuk jatuh ke tanah
G21	Buah dan daun cabai rusak di pesemaian masa vegetatif dan generatif panen
G22	Buah berserakan di permukaan lubang dan selokan sekitar tanaman
G23	Buah cabai kecil dan buah gugur
G24	Bercak kecil coklat kehitaman pada buah, buah membusuk
G25	Bercak hitam menjadi busuk lunak dan warna kulit buah seperti jerami
G26	Buah busuk banyak yang berjatuhan
G27	Terdapat bercak kecil kebasah-basahan pada buah, buah mongering
G28	Biji buah yang terserang menjadi coklat dan keriput
G29	Terbakarnya buah cabai akibat sengatan sinar matahari
G30	Buah berwarna hijau gelap, kemudian menjadi lekukan basah kecoklatan
G31	Biji membusuk di dalam tanah
G32	Tanaman menjadi keriput dan pertumbuhan tanaman kerdil
G33	Akar tanaman membusuk
G34	Pertumbuhan tanaman terhambat
G35	Semai mati sebelum muncul ke permukaan tanah
G36	Warna jaringan akar, batang dan daun kuning kecoklatan
G37	Jaringan vaskuler bawah batang kecoklataan
G38	Adanya lendir berwarna putih jika batang dipotong

4. HASIL DAN PEMBAHASAN

Halaman konsultasi digunakan oleh pengguna untuk mencari hama dan penyakit tanaman cabai merah dengan memilih gejala yang diderita di dalam form yang telah disediakan. Pengguna hanya dapat memilih gejala yang ada yang ada di dalam *form* sebagai masukan untuk dihitung similaritasnya dengan data gejala pada penyakit yang ada di dalam basisdata. Berikut tampilan *form* konsultasi dengan nama pengguna yang telah melakukan proses *login* terlebih dahulu. Pengguna memberikan inputan gejala warna daun berubah menjadi warna kuning, daun menebal menyusut seperti terpuntir dan bunga gugur, warna jaringan akar batang dan daun kuning kecoklatan, daun muda layu secara mendadak, jaringan *vaskuler* bawah batang kecoklatan, adanya lendir berwarna putih jika batang dipotong, terdapat bercak pada daun menyerupai mata kodok, daun menguning dan terdapat bercak pada batang, daun menggulung keatas dan menyusut tanaman kerdil, seperti terlihat pada form konsultasi pada gambar 5.



Gambar 5. Tampilan Form Konsultasi

Setelah pengguna memilih gejala pada form konsultasi, sistem akan melakukan proses *retrieve* atau pencarian kemiripan gejala masukan dengan gejala pada penyakit tanaman cabai merah yang ada di dalam basisdata. Berikut adalah proses *retrieve*:

Perhitungan kasus dengan ID 1 (Hama Tungau Kuning)

Kasus Baru	Data Penyakit: ID 1
Gejala : 1. Warna daun berubah menjadi warna kuning 2. Daun menebal, menyusut seperti terpuntir dan bunga gugur 3. Warna jaringan akar, batang dan daun kuning kecoklatan 4. Daun muda layu secara mendadak 5. Jaringan vaskuler bawah batang kecoklataan 6. Adanya lendir berwarna putih jika batang dipotong 7. Terdapat bercak pada daun menyerupai mata kodok 8. Daun menguning dan terdapat bercak pada batang 9. Daun menggulung keatas dan menyusut, tanaman kerdil	Gejala : 1. Warna daun berubah menjadi warna kuning 2. Daun menebal, menyusut seperti terpuntir dan bunga gugur 3. Daun berwarna kuning jelas, tulang daun menebal

Gambar 6. Proses retrieve pada kasus baru

Pada proses ini sistem melakukan proses *retrieve* dengan mengambil kasus-kasus yang disimpan di dalam *case base* kemudian membandingkan setiap gejala *source case* dengan gejala-gejala pada *target case*. Jika perbandingan nilai *source case* dan nilai *target case* menghasilkan nilai yang hampir sama maka *source case* akan diusulkan sebagai solusi. Berikut ini adalah proses perhitungan:

Nilai bobot parameter pada Gejala Penyakit = 1

Setelah mendapatkan kasus-kasus yang tersimpan dalam *case base* maka proses pertama yang dilakukan adalah *indexing*. Proses ini akan menghitung jarak (*distance*) untuk mengetahui kedekatan setiap gejala *source case* dengan gejala-gejala pada *target case*. Proses *indexing* bertujuan agar pada saat proses perhitungan nilai similaritas tidak terlalu lama. Proses *indexing* menggunakan metode k-nearest neighbor.

$$\text{Similaritas (problem, case)} = \frac{\text{Jumlah nilai source case gejala}}{\text{Jumlah nilai target case gejala}}$$

Berikut adalah contoh proses perhitungan *indexing* dengan ID kasus 1.

Perhitungan K-nearest neighbor :

$$\text{Similaritas (x, 1)} = \frac{1 + 1}{1 + 1 + 1} = \frac{2}{3} = 0.6667$$

Proses kedua setelah proses *indexing* adalah proses penghitungan nilai similaritas menggunakan algoritma Neyman.

$$d_N(P,Q) = \sum_{i=1}^d \frac{(P_i - Q_i)^2}{P_i}$$

$$S = 1 - \sum_{i=1}^d \frac{(P_i - Q_i)^2}{P_i}$$

P = Nilai *distance* / jarak proses *indexing*

Q = Nilai *distance* / jarak *target case*

Nilai *distance*/jarak *target case* d disini selalu bernilai selalu satu karena gejala-gejala pada *target case* sama semua dengan *target case* pada sistem. Perhitungan algoritma similaritas Neyman :

$$D_p(x,1) = 1 - (P_i - Q_i)^2 / P_i$$

$$D_p(x,1) = 1 - (0.67 - 1)^2 / 0.67$$

$$D_p(x,1) = 1 - (-0.33)^2 / 0.67$$

$$D_p(x,1) = 1 - (-0.33 * -0.33) / 0.67$$

$$D_p(x,1) = 1 - (0.1089 / 0.67)$$

$$D_p(x,1) = 1 - 0.1625$$

$$D_p(x,1) = 0.83$$

Dari hasil perhitungan similaritas kasus di atas yang memiliki nilai similaritas paling kecil yaitu kasus ID 9 (Penyakit Bercak Daun) dengan nilai similaritas 0.1. Di ikuti dengan kasus ID 7 (Penyakit Layu Fusarium) dengan nilai similaritas 0.5. Kemudian diikuti dengan kasus ID 1 (Hama Tungau Kuning) dengan nilai similaritas 0.83. Kemudian diikuti dengan kasus ID 13 (Penyakit Virus Kuning) dengan nilai similaritas 0.91, dan kasus dengan nilai similaritas paling tinggi adalah kasus ID 8 (Penyakit Layu Bakteri) dengan nilai similaritas 1.

Kasus dengan nilai similaritas tertinggi inilah yang akan dijadikan solusi proses *reuse*), yaitu kasus dengan ID 8 dan penyakit yang memiliki gejala-gejala tersebut adalah Penyakit Layu Bakteri.

5. KESIMPULAN

Sistem pakar untuk hama dan penyakit tanaman cabai merah menggunakan metode *Case Based Reasoning* dengan algoritma similaritas Neyman dapat digunakan masyarakat luas khususnya petani cabai merah untuk membantu mengetahui jenis hama dan penyakit tanaman cabai merah yang diderita. Sistem yang dibuat mampu mengidentifikasi hama dan penyakit tanaman cabai merah yang diinputkan oleh pengguna dengan cara menghitung nilai similaritasnya dengan yang ada pada sistem menggunakan algoritma similaritas Neyman. Solusi penanganan penyakit yang diderita diberikan setelah mengetahui jenis hama dan penyakit dari hasil identifikasi menggunakan algoritma similaritas Neyman akan menampilkan hasil identifikasi hama dan penyakit tanaman cabai merah apabila hasil similaritasnya lebih dari sama dengan 50 persen, namun apabila nilai similaritas kurang dari 50 persen maka akan dilakukan proses *revise* untuk menjadi temuan hama dan penyakit baru.

DAFTAR PUSTAKA

- [1] Aamodt. A., dan Plaza E., 1994, *Case Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches*, *IA Com-Artificial Intelligence Communication, IOS Press*, Vol. 7. Ed. 1.
- [2] Faizal, E., 2012, *Case Based Reasoning Diagnosis Penyakit Mata*, *FAHMA – Jurnal Teknologi Informasi dan Ilmu Komputer*, Vol. 10, Ed. 2.
- [3] Shaid, M., Laksito, W., dan Utami, Y. R. W., 2015, *Sistem Pakar Pertumbuhan Balita Berbasis Web dengan Metode Case Based Reasoning*, *Jurnal Teknologi Informasi dan Komunikasi (TIKoSIN)*, Vol 3, Ed. 1.
- [4] Scott, C., dan Nowak, R., 2005, *A Neyman-Pearson Approach to Statistical Learning*, *IEEE Transactions on Information Theory*, Vol. 51, Ed. 11.
- [5] Rismawan, T., dan Hartati, S., 2012, *Case-Based Reasoning untuk Diagnosa Penyakit THT (Telinga Hidung dan Tenggorokan)*, *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, Vol. 6, Ed. 2.
- [6] Prakoso, I.M., Anggraeni, W., dan Mukhlason, A., *Penerapan Case-Based Reasoning pada Sistem Cerdas untuk Pendeteksian dan Penanganan Dini Penyakit Sapi*, *Jurnal Teknik ITS*, Vol. 1, Ed. 1.
- [7] Cha, S.H., 2007, *Comprehensive Survey On Distance/Similarity Measures Between Probability Density Functions*, *International Journal Of Mathematical Models And Methods In Applied Sciences*, Issue 4, Vol. 1.