

CASE BASED REASONING UNTUK MENDIAGNOSA PENYAKIT DAN HAMA PADA TANAMAN MANGGA MENGGUNAKAN ALGORITMA SIMILARITAS SORGENFREI

Anton Setiawan¹, Setyawan Wibisono²

^{1,2}Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Stikubank Semarang
email: ¹ants.setiawan@gmail.com, ²setyawan@edu.unisbank.ac.id

Abstrak

Untuk mendapatkan suatu informasi tentang bagaimana mendiagnosa penyakit dan hama pada tanaman mangga secara digital maka dibutuhkan suatu aplikasi dalam bentuk sistem pakar yang dapat mewakili seorang pakar yang ahli di bidangnya untuk memberikan solusi terhadap permasalahan tanaman mangga. Case Based Reasoning untuk mendiagnosa penyakit dan hama tanaman mangga menggunakan algoritma similaritas Sorgenfrei adalah sebuah sistem pakar yang dapat sebagai alat bantu untuk melakukan konsultasi tentang penyakit dan hama tanaman mangga. Untuk mengimplementasikan Case Based Reasoning diperlukan empat tahapan proses yaitu retrieve, reuse, revise, dan retain. Sistem yang telah memberikan hasil dari perhitungan dengan nilai paling tinggi yang akan dijadikan solusi penyakit dan hama tanaman mangga. Pada proses revise, sistem akan meninjau kembali hasil perhitungan penyakit tanaman mangga. Jika hasil tersebut kurang atau sama dengan 20 persen, maka sistem tidak akan memberikan solusi penyakit tanaman mangga, dikarenakan informasi berupa penyakit yang tidak memenuhi syarat akan masuk ke dalam tabel revise yang selanjutnya akan diperbaiki kembali oleh pakar untuk menemukan solusi yang tepat. Setelah proses revise selesai dan sudah ditemukan solusi yang benar-benar tepat barulah pakar mulai menambah aturan dengan memasukan data kasus baru yang sudah ditemukan solusinya proses inilah yang dinamakan dengan proses retain.

Kata Kunci: algoritma similaritas Sorgenfrei, Case Based Reasoning, mangga, sistem pakar

1. PENDAHULUAN

Sistem pakar merupakan aplikasi kecerdasan buatan atau *Artificial Intelligence* (AI) yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Salah satu masalah dalam sistem pakar adalah dalam bidang pertanian, di antaranya untuk mendiagnosa penyakit dan hama pada tanaman mangga.

Mangga (*Mangifera indica*) di Indonesia merupakan komoditas ekspor. Mangga Probolinggo terutama varietas Arumanis sangat populer bahkan sudah dipasarkan sampai ke luar negeri seperti ke Singapura. Namun, tumbuhan tidak selamanya dapat terlepas dari serangan hama dan penyakit. Hama dan penyakit tanaman dapat disebut juga sebagai organisme pengganggu tanaman. Hewan dapat disebut hama karena mereka mengganggu tumbuhan dengan memakannya. Belalang, kumbang, ulat, wereng, tikus, walang sangat merupakan beberapa contoh binatang yang sering menjadi hama tanaman. Terdapat puluhan bahkan ratusan jenis hama dan penyakit yang menyerang tanaman. Berbagai hama dan penyakit pada tanaman mangga yang menyerang dapat ditandai dari gejala-gejala serangan yang timbul. Dengan adanya sistem pakar nanti diharapkan berguna bagi petani untuk mengetahui beberapa jenis hama atau penyakit yang paling umum atau yang paling sering menyerang sehingga dapat meminimalkan kerugian atau kerusakan tanaman mangga.

Untuk memberikan suatu informasi tentang bagaimana mendiagnosa penyakit dan hama pada tanaman mangga, dibutuhkan suatu aplikasi sistem pakar yang dapat mewakili seorang pakar yang ahli di bidangnya untuk memberikan solusi terhadap permasalahan tanaman mangga. Metode *Case Based Reasoning* ini adalah salah satu metode untuk membangun sistem pakar dengan pengambilan keputusan dari kasus yang baru dengan berdasarkan solusi dari kasus-kasus sebelumnya.

2. TINJAUAN PUSTAKA

Buah durian adalah salah satu buah tropik yang berasal dari Asia Tenggara yang cukup digemari oleh masyarakat. Penyakit pada buah durian sering dijumpai misalnya gala-gala/penggerek buah yang biasanya untuk mengetahui ciri-ciri durian yang terserang hama dan penyakit masih menggunakan diagnosa manual oleh para ahli tanaman secara langsung. Untuk itu diperlukan sistem pakar berbasis web untuk mengidentifikasi jenis hama dan penyakit durian serta penanggulangannya [1].

Pengembangan sistem pakar yang bergerak di bidang pertanian, yaitu sistem yang dapat diterapkan pada kasus penentuan penyakit pada tanaman melon serta cara penanggulangannya. Penentuan penyakit pada tanaman melon serta cara penanggulangannya sering menjadi permasalahan bagi petani melon. Untuk penentuan penyakit dibutuhkan seorang pakar yaitu seorang ahli dalam bidang pertanian khususnya tanaman melon. Untuk mencari informasi atau bertanya langsung pada ahlinya tentu saja memakan waktu yang tidak sedikit, bahkan ahli pertanian tidak selalu dapat membantu memecahkan permasalahan tersebut setiap waktu. Untuk alasan tersebut, maka perlu dikembangkan suatu sistem pakar untuk mempermudah user dalam menentukan penyakit pada tanaman melon berdasarkan gejala-gejala yang ada. Sistem pakar ini akan dibangun dengan menggunakan metode *Case Based Reasoning*. Metode ini akan mengambil solusi dari kasus-kasus sebelumnya yang akan digunakan untuk menyelesaikan kasus baru. Sistem ini bertujuan untuk meneliti apakah *Case Based Reasoning* dapat digunakan untuk menentukan penyakit pada tanaman melon [2].

Hama dan penyakit yang menyerang tanaman mangga menyebabkan petani mangga mengalami kerugian karena tanamannya rusak atau gagal panen. Untuk membantu kerja para penyuluh pertanian dalam mendiagnosa hama dan penyakit yang menyerang mangga apabila pakar dan ahli pertanian belum ada atau tidak ada, maka diperlukan sebuah sistem berbasis komputer yang dapat memberikan informasi hama dan penyakit dan tata cara pengendaliannya. Penelitian ini membangun sebuah sistem pakar berbasis aturan (*rule based*) dengan menggunakan metode *certainty factor* untuk mendiagnosa hama dan penyakit. Hasil yang diperoleh yang harus dilakukan, serta nilai *certainty factor* akan kebenaran dari hama atau penyakit tersebut [3].

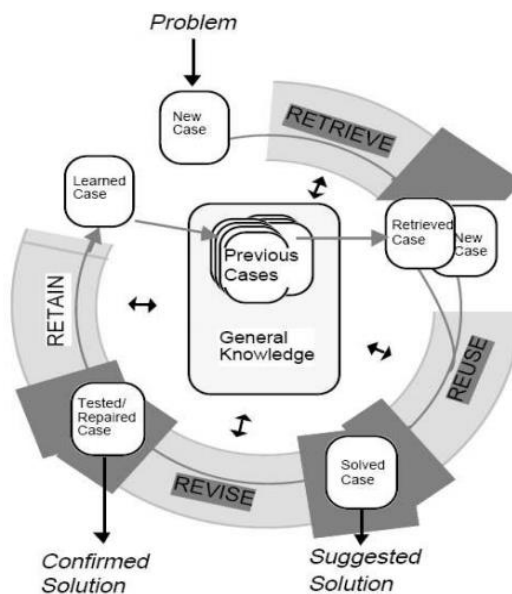
Hasil panen padi dari tahun ke tahun mengalami penurunan dikarenakan tanaman padi rentan terhadap hama dan penyakit. Maka diperlukan suatu aplikasi sistem pakar yang diharapkan mampu mengatasi masalah tersebut. Aplikasi yang dibangun menggunakan 3 tahapan proses identifikasi penyakit tanaman padi, yaitu tahapan pra-proses grayscale digunakan untuk ekstraksi ciri warna, ekstraksi ciri menggunakan local binary pattern dan metode *Jensen Shannon Divergence* untuk mengklasifikasikan dengan cara menghitung kedekatan vector dari citra sampel dan citra uji. Dalam penelitian ini terdapat 4 jenis penyakit utama yaitu tungro, hawar daun, *leafblas* dan bercak coklat. Dari hasil pengujian yang dilakukan tingkat akurasi data dalam proses identifikasi penyakit pada padi adalah 86% [4].

3. METODE PENELITIAN

3.1 *Case Based Reasoning* (CBR)

Case Based Reasoning (CBR) adalah salah satu metode untuk membangun sistem dengan pengambilan keputusan untuk memecahkan kasus atau masalah yang baru dengan cara mengingat solusi dari kasus lama/sebelumnya dengan menggunakan informasi dan pengetahuan pada situasi tersebut. Semakin mirip masalah, semakin mirip pula solusinya.

CBR dapat digunakan untuk menyimpan dan menggunakan kembali pengetahuan dari pengalaman sebelumnya. Dalam CBR pengalaman dimodelkan ke dalam bentuk problem dan solusi dan biasa disebut dengan kasus (*case*). Berdasarkan model deskripsi suatu masalah dan solusi yang berisi hasil penyelesaian masalah. Siklus *Case Based Reasoning* ditunjukkan pada gambar 1.



Gambar 1. Siklus CBR [5]

Untuk kasus – kasus yang tersimpan dalam tempat penyimpanan kasus (*case based*) disebut dengan kasus asal (*source case*), sedangkan untuk yang baru akan dicari solusinya disebut dengan kasus target (*target case*). Secara umum, metode ini terdiri dari 4 langkah antara lain:

1. *Retrieve* (penelusuran)

Proses akan dimulai dengan tahapan mengenali masalah, dan berakhir ketika kasus yang ingin dicari solusinya telah ditemukan serupa dengan kasus yang telah ada. Proses *retrieve* akan melakukan dua langkah pemrosesan, yaitu pencarian persamaan masalah (*similar case*) atau pencarian masalah baru (*new case*) pada *database*.

2. *Reuse* (menggunakan kembali)

Masalah/kasus untuk mencoba memecahkan masalah/kasus. *Reuse* suatu kasus dalam konteks kasus baru terfokus pada dua aspek yaitu perbedaan antara kasus yang ada dengan kasus yang baru, serta bagian mana dari penelusuran kasus yang dapat digunakan pada kasus yang baru.

3. *Revise* (meninjau kembali/memperbaiki) usulan solusi

Revise terdiri dari dua tugas, yaitu mengevaluasi solusi kasus yang dihasilkan oleh proses *reuse*. Jika berhasil, maka dilanjutkan dengan proses *retain*, serta jika tidak maka memperbaiki solusi kasus menggunakan domain spesifik pengetahuan.

4. *Retain* (menyimpan)

Proses *retain* akan menggunakan solusi baru sebagai bagian dari kasus baru. Selanjutnya, solusi baru itu akan disimpan ke dalam *knowledge-based* (basis pengetahuan) kemudian kasus baru di-*update* ke dalam basis kasus untuk menyelesaikan permasalahan yang akan datang. Tentunya permasalahan yang akan diselesaikan adalah permasalahan yang memiliki kesamaan. Pada tahap ini terjadi suatu proses penggabungan dari solusi kasus yang baru dan benar ke *knowledge* yang telah ada.

3.2 Algoritma *K-Nearest Neighbor* (K-NN)

K-Nearest Neighbor adalah sebuah algoritma untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Kasus khusus di mana klasifikasi diprediksikan berdasarkan data pembelajaran yang paling dekat dengan kata lain, k disebut algoritma *Nearest Neighbor* [6].

Rumus untuk menghitung nilai *K-Nearest Neighbor* adalah:

$$Similarity (problem, case) = \frac{(s_1 * w_1) + (s_2 * w_2) + \dots + (s_n * w_n)}{(w_1) + (w_2) + \dots + (w_n)} \quad (1)$$

Keterangan:

S = *similarity* (nilai kemiripan) yaitu 1 (sama) dan 0 (beda)
 W = *weight* (bobot yang diberikan)

Rumus *K-Nearest Neighbor* diatas digunakan untuk menyempurnakan tingkat nilai kemiripan, dengan metode *indexing* antara KNN dan *Sorgenfrei* dapat melakukan perhitungan tingkat kemiripan yang lebih sempurna dari dua buah objek yang diteliti.

3.3 Algoritma Similaritas Sorgenfrei

Similaritas merupakan ukuran kedekatan antara satu objek dengan objek lainnya. Kemudian yang dimaksud dengan similaritas *Sorgenfrei* adalah metode yang digunakan untuk menghitung kemiripan antara jarak dua distribusi probabilitas dengan rumus sebagai berikut:

$$S_{\text{Sorgenfrei}} = \frac{a^2}{(a+b) \times (a+c)} \quad (2)$$

Keterangan:

S = Nilai similaritas
 a = Jumlah atribut yang sama antara konsultasi dan data yang tersimpan dalam *database*
 b = Jumlah atribut yang dimiliki data yang tersimpan dalam *database*, namun tidak dimiliki data konsultasi
 c = Jumlah atribut yang dimiliki data konsultasi, namun tidak dimiliki data yang tersimpan dalam *database*

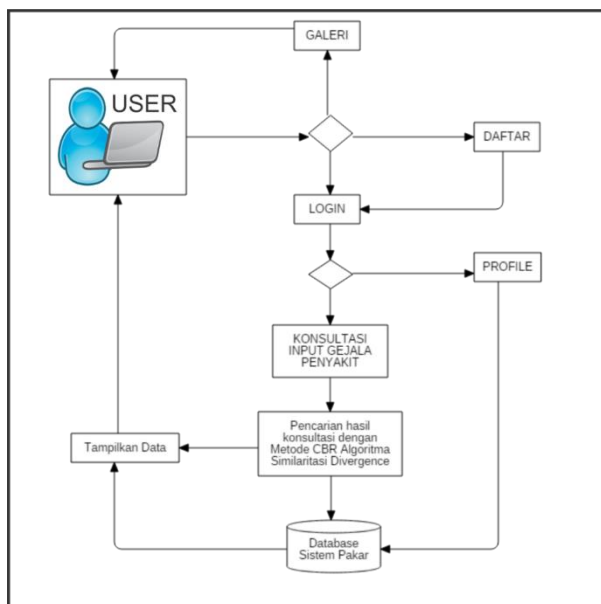
Dari rumus di atas *Sorgenfrei* merupakan konsep penting dalam teori informasi, karena jumlah informasi-teori lainnya termasuk entropi dan informasi timbal balik dapat diformulasikan sebagai kasus khusus. *Sorgenfrei* juga memainkan peranan penting dalam menghasilkan besarnya kemiripan termasuk asymptotic tingkat penurunan probabilitas kesalahan dalam pengujian hipotesis masalah biner. Selain itu, perbedaan telah terbukti berguna dalam aplikasi [7].

3.4 Deskripsi Sistem

Case Based Reasoning untuk mendiagnosa penyakit dan hama tanaman mangga menggunakan algoritma similaritas *Sorgenfrei* dapat digunakan untuk melihat informasi penyakit dan hama tanaman mangga. Dalam memulai proses pada aplikasi ini dibedakan menjadi dua yaitu pengguna dan administrator. Untuk pengguna yang belum mempunyai *username* dan *password* dapat melakukan registrasi pada form registrasi aplikasi. Baru pengguna dapat melakukan konsultasi dengan memilih gejala yang dialami tanaman mangga.

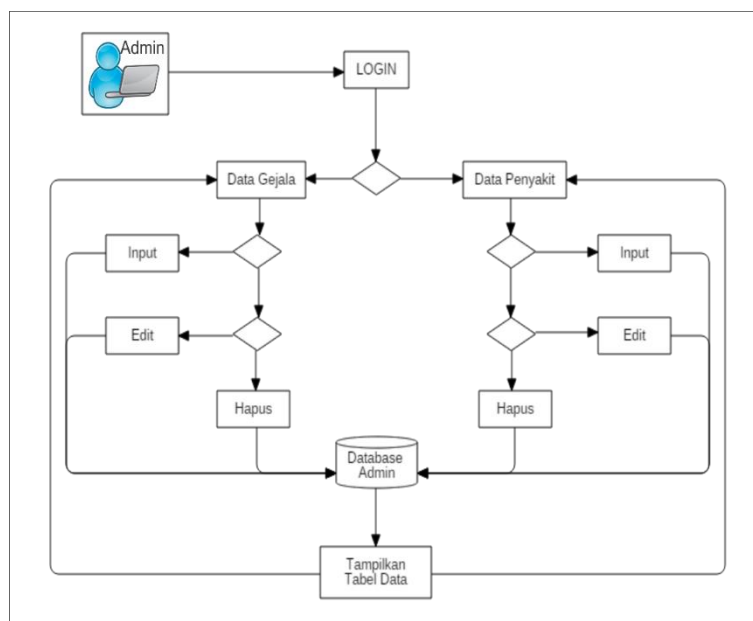
Untuk mengimplementasikan *Case Based Reasoning* diperlukan empat tahapan proses yaitu *retrieve*, *reuse*, *revise*, dan *retain*, sistem yang telah memberikan hasil dari perhitungan dengan nilai paling tinggi yang akan dijadikan solusi penyakit dan hama tanaman mangga. Pada proses *revise*, sistem akan meninjau kembali hasil perhitungan penyakit tanaman mangga. Jika hasil tersebut kurang dari atau sama dengan 20 persen, maka sistem tidak akan memberikan solusi penyakit tanaman mangga. Dikarenakan informasi berupa penyakit yang tidak memenuhi syarat akan masuk ke dalam tabel *revise* yang selanjutnya akan diperbaiki kembali oleh pakar untuk menemukan solusi yang tepat. Setelah proses *revise* selesai dan sudah ditemukan solusi yang benar-benar tepat barulah pakar mulai menambah aturan dengan memasukan data kasus baru yang sudah ditemukan solusinya proses inilah yang dinamakan dengan proses *retain*.

Pengguna dapat memilih gejala yang dialami oleh tanaman mangga, kemudian sistem akan menampilkan data penyakit dan hama tanaman mangga kepada *user* dengan perintah SQL. Sedangkan admin harus *login* terlebih dahulu agar dapat untuk memilih menu gejala penyakit, penyakit dan *revise*. Pada menu gejala, admin dapat melakukan inputan data gejala penyakit baru. Untuk mengedit data yang dipilih dari tabel data yang ditampilkan, setelah pengeditan selesai data-data tersebut akan ditransferkan ke variabel-variabel yang ditentukan dengan format kalimat SQL. Dalam proses hapus mengambil data yang berada di tabel dengan format kalimat SQL delete. Alur kerja user dapat dilihat pada gambar 2.



Gambar 2. Blok diagram proses kerja user

Pada menu penyakit, *input data*, *edit data*, dan *hapus data* proses kerjanya sama dengan proses menu gejala penyakit yaitu menambah data penyakit yaitu dengan *query input*, *edit* dengan *query update*, dan *hapus* dengan *query delete*. sedangkan alur kerja admin ditunjukkan pada gambar 3.



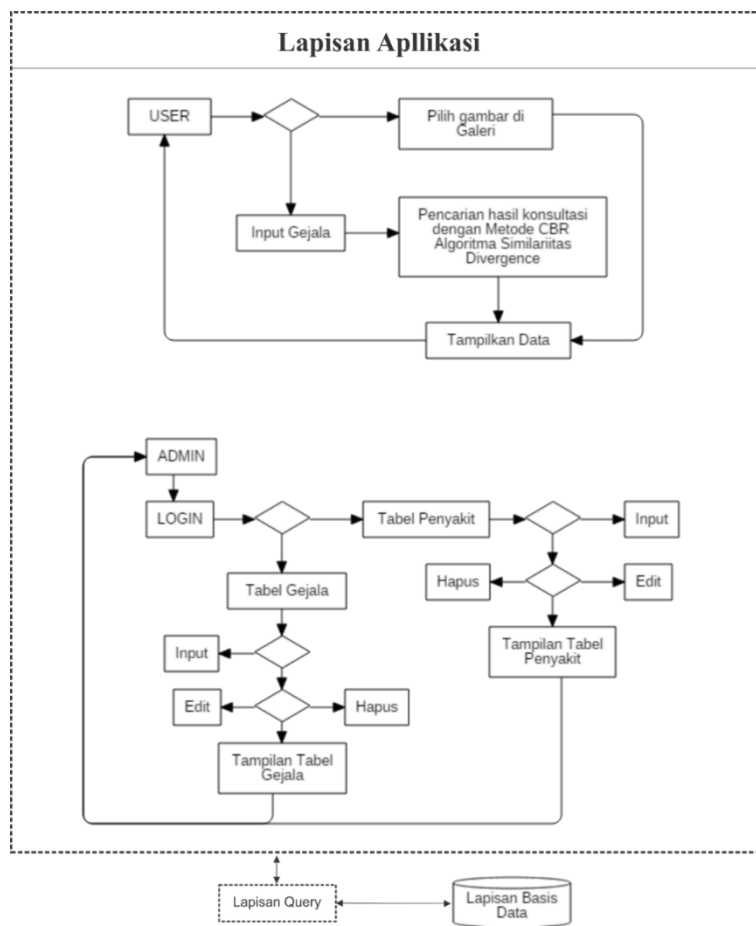
Gambar 3. Blok diagram proses kerja administrator

3.5 Arsitektur Sistem

Sistem ini dibagi menjadi 3 lapisan yaitu: lapisan *database*, lapisan bahasa *query* dan lapisan aplikasi. Lapisan *database* merupakan lapisan yang digunakan untuk menyimpan data gejala dan penyakit. Media penyimpanan data dalam aplikasi ini menggunakan AppServ 2.5.9 dengan format data SQL. Lapisan *query* merupakan lapisan PHP *query* yang digunakan untuk pemisahan variabel data yang akan diubah ke dalam format SQL sebelum melakukan penyimpanan data. Lapisan aplikasi adalah antarmuka dalam aplikasi yang memudahkan pengguna untuk melakukan interaksi terhadap sistem.

Pada antarmuka sistem pengguna yang akan melakukan pencarian gejala penyakit dapat menginputkan gejala. Sistem akan menggunakan lapisan *query* untuk mencari dengan menggunakan algoritma similaritas pada *database* dan menampilkan hasil similaritas terbesar.

Pada antarmuka admin terdapat form login untuk selanjutnya menampilkan pilihan gejala dan penyakit. Form gejala digunakan untuk melihat data gejala dalam bentuk tabel. Pada tabel gejala penyakit dapat melakukan *input*, *edit*, dan hapus data gejala. Form penyakit digunakan untuk melihat data penyakit dalam bentuk tabel. Pada tabel penyakit, admin dapat melakukan *input*, *edit*, dan hapus bahan. Untuk mengaplikasikan aplikasi antarmuka menggunakan bahasa pemrograman PHP. Arsitektur sistem diperlihatkan pada gambar 4.



Gambar 4. Arsitektur sistem

3.6 Analisa Data Penyakit dan Gejala

Dalam pembuatan sebuah sistem perlu adanya data penunjang seperti data gejala, data penyakit dan gambar. Data tersebut bisa diperoleh dari hasil wawancara, jurnal dan buku agar nantinya dapat dikonversikan ke dalam sistem guna mempermudah proses pencarian solusi. Data penyakit dan data gejala digunakan sebagai pola pencocokan informasi yang dimasukkan oleh *user* dan berbasis pengetahuan seperti terlihat pada tabel 1.

Tabel 1. Penyakit dan Solusi

Kode	Penyakit	Solusi
P001	Antraknose	Memotong dan memusnahkan bagian tanaman yang terserang, Membersihkan dan memusnahkan daun-daun kering, buah yang gugur di sekitar tanaman. Pada saat tampak gejala awal dilakukan penyemprotan fungisida DELSENE MX -200, DEROSAL 500 EC atau ALTO 100 SL.
P002	Penyakit	Ranting atau cabang yang terserang dipotong dan dibakar,

	Diplodia	batang yang terserang dikupas kulit luarnya sampai batas jaringan kayu, kemudian diolesi dengan fungisida BENLATE seminggu sekali. Apabila gejala awal muncul pada saat buah masih muda, disemprot dengan fungisida KOCIDE 10 hari sekali sebanyak 3 kali.
P003	Penyakit Gloesporium	Fungisida bubuk Bordeaux.
P004	Penyakit Cendawan Jelaga	Dengan memberantas serangga yang menghasilkan cairan manis dengan insektisida atau tepung belerang.
P005	Penyakit Kudis	Fungisida Dithane M-45, Manzate atau Pigone tiga kali seminggu. Memangkas tangkai bunga yang terserang.
P006	Bercak Karat Merah	Pemangkasan daun, cabang, ranting, menyemprotkan fungisida bubuk bordeaux atau sulfat tembaga.
P007	Penyakit Kulit/Blendok	Memotong bagian yang sakit, lubang ditutupi dengan kapas yang telah dicelupkan ke dalam insektisida dan menyemprot pohon dengan bubuk bordeaux.
P008	Gulma/Benalu	Potong bagian cabang dan ranting yang sudah terjangkiti benalu hingga benar-benar bersih tidak ada lagi akar benalu yang menempel.
P009	Wereng Mangga	Dengan pengasapan, memotong bagian bunga yang terserang, kemudian dimusnahkan. Menyuntik pohon mangga dengan insektisida sistemik sebanyak 10 - 20 cc per pohon. Menyemprotkan insektisida APLAUD 400 F atau DARMABASH 500.
P010	Ulat Penggerek Buah	Membungkus (membrongsong) buah dengan kertas semen. Menyemprotkan insektisida AGROTHION 50 EC atau AMCOTHENE 75 SP pada saat buah muda. Memusnahkan buah-buah yang gugur.
P011	Lalat Buah	Membungkus (membrongsong) buah muda sampai menjadi tua dengan kertas semen. Memasang umpan metyl eugenol. Menyemprotkan insektisida DECIS atau TALSTAR 25 EC pada saat buah muda. Memusnahkan buah-buah busuk.
P012	Bisul Daun	Penyemprotan buah dan daun dengan Ripcord, Cymbuth atau Phosdrin tiga kali dalam seminggu. Membakar daun yang terserang. Menggemburkan tanah untuk mengeluarkan kepompong dan memperbaiki aerasi.
P013	Thrips (Scirtothrips dorsalis)	Tunas muda terserang dipotong lalu dibakar. Tangkap dengan perangkap warna kuning. Pemangkasan teratur. Penyemprotan dengan menggunakan produk nasa yang berupa pestona + Aero-810 dengan dosis (5 + 1/3) tutup /tangki semprot, lakukan penyemprotan di sore hari.
P014	Codot/Kalong	Dengan membiarkan semut kerangkang hidup di sela daun mangga. Memasang kitiran angin berpeluit dan melindungi pohon dengan jaring atau di brongsong.

Tabel gejala adalah tabel yang akan digunakan sebagai basis pengetahuan. Data gejala yang dihimpun adalah data gejala yang nantinya akan dimasukkan dalam konsultasi maupun data gejala suatu penyakit dan hama tanaman mangga yang sudah dicatat dalam basisdata seperti terlihat pada tabel 2.

Tabel 2. Gejala

Kode	Gejala
G001	Bercak hitam kecoklatan/kemerahan pada daun
G002	Bunga menjadi mengering dan berguguran
G003	Bercak hitam pada buah dan berguguran
G004	Kulit batang/ranting pecah-pecah
G005	Mengelurkan cairan getah berwarna coklat kehitaman pada batang
G006	Kulit batang menglupas dan akhirnya bisa mati
G007	Daun berbintik hitam dan menggulung
G008	Bunga menjadi layu
G009	Buah busuk
G010	Daun berubah menjadi hitam
G011	Bercak kuning yang akan berubah menjadi abu-abu
G012	Mempengaruhi proses pembuahan
G013	Bercak coklat kemerahan pada daun dan bunga
G014	Pohon menjadi susah besar alias pertumbuhannya lambat
G015	Daun mulai berkurang sehingga pohon terlihat gersang
G016	Buah berlubang dan mengeluarkan getah
G017	Kulit buah terdapat titik noda hitam
G018	Buah jatuh/berguguran
G019	Permukaan daun berbintil-bintil menyerupai bisul
G020	Daun Memucat
G021	Daun berguguran
G022	Tunas daun layu
G023	Buah bekas gigitan hama
G024	Mati pucuk

4. HASIL DAN PEMBAHASAN

Rancangan *Case Based Reasoning* untuk diagnosa hama dan penyakit tanaman mangga dengan menggunakan algoritma similaritas Sorgenfrei yang telah dibuat akan dilakukan pengujian guna mengetahui kinerja aplikasi, dan berikut hasil pengujiannya.

Perhitungan kasus dengan ID_Kasus: K001 dibandingkan dengan penyakit dengan ID_Penyakit: P001, dengan nama penyakit: Antraknose. Diagnosa pertama dilakukan oleh *user* dengan memilih gejala tanaman mangga yang berada di dalam tabel seperti tabel 3.

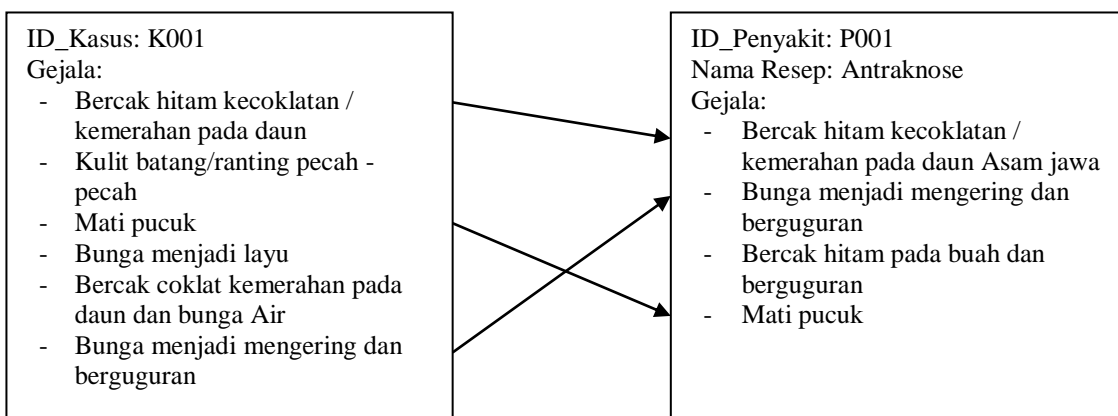
Tabel 3. Contoh gejala dalam konsultasi

Gejala yang dipilih:
Bercak hitam kecoklatan / kemerahan pada daun
Kulit batang/ranting pecah - pecah
Mati pucuk
Bunga menjadi layu
Bercak coklat kemerahan pada daun dan bunga
Bunga menjadi mengering dan berguguran

G 3. Contoh gejala dalam konsultasi

Setelah *user* memilih gejala, sistem akan melakukan proses pencarian kemiripan gejala yang dipilih dengan gejala pada tabel penyakit. Proses ini dilakukan dengan metode *Case Based Reasoning*

dengan algoritma Sorgenfrei. Untuk nilai bobot tiap parameter gejala hanya memilih 1, karena tiap gejala memiliki nilai bobot yang sama. Berikut proses perhitungan pada sistem.



Gambar 6. Ilustrasi pencocokan antara gejala pada data konsultasi dan data penyakit

Dalam tahapan ini akan dilakukan pencocokan antara gejala yang dimasukkan sebagai kasus baru yang dikonsultasikan dengan gejala yang berada dalam sebuah penyakit yang telah tersimpan dalam *database*. Dari proses pencocokan ini akan terdapat tiga hal penting yang menjadi dasar perhitungan nilai similaritas, yaitu:

1. Gejala yang sama-sama dimiliki oleh konsultasi dan penyakit yang telah tersimpan dalam *database*, yang dalam algoritma similaritas dinotasikan dengan simbol ‘a’.
2. Gejala yang dimiliki oleh penyakit yang telah tersimpan dalam *database*, namun tidak terdapat dalam konsultasi, yang dalam algoritma similaritas dinotasikan dengan simbol ‘b’.
3. Gejala yang dimiliki oleh konsultasi, namun tidak terdapat dalam penyakit yang telah tersimpan dalam *database*, yang dalam algoritma similaritas dinotasikan dengan simbol ‘c’.

Berikut ini adalah contoh proses perhitungan nilai similaritas menggunakan algoritma Sorgenfrei berdasarkan contoh konsultasi K001:

- ID_Penyakit: P001
- Nama penyakit: Antraknose
- Proses penghitungan nilai similaritas dengan pembobotan menggunakan algoritma Sorgenfrei, dengan terlebih dahulu mencari:
 $a = 3$
 $b = 1$
 $c = 3$
- Maka nilai kemiripan dengan menggunakan algoritma Sorgenfrei dihitung sebagai berikut:

$$S_{\text{Sorgenfrei}} = \frac{a^2}{(a+b) \times (a+c)}$$

$$S_{\text{Sorgenfrei}} = \frac{3^2}{(3+1) \times (3+3)}$$

$$S_{\text{Sorgenfrei}} = \frac{9}{4 \times 6}$$

$$S_{\text{Sorgenfrei}} = \frac{9}{24}$$

$$S_{\text{Sorgenfrei}} = 0.375$$

Dari hasil perhitungan algoritma similaritas Sorgenfrei diketahui bahwa pada kasus dengan ID_Kasus K001 jika dibandingkan penyakit P001 akan diperoleh nilai kemiripan sebesar 0,375, yang dapat diartikan bahwa kasus K001 mempunyai kemiripan yang relatif rendah, karena nilai kemiripan mempunyai rentang 0 sampai dengan 1. Nilai 0 adalah nilai kemiripan yang menyatakan tidak ada kemiripan sama sekali, sedangkan nilai kemiripan 1 adalah nilai kemiripan yang menyatakan bahwa kasus baru sama persis dengan resep masakan yang ada dalam basisdata.

5. KESIMPULAN

Case Based Reasoning untuk mendiagnosa penyakit dan hama pada tanaman mangga menggunakan algoritma similaritas Sorgenfrei dapat digunakan untuk mencari data penyakit yang mengidentifikasi suatu gejala yang di masukan *user*. Semua gejala yang ada dalam *database* mempunyai nilai bobot 1 (satu), karena semua penyakit memungkinkan memiliki gejala yang sama dengan yang lainnya. Sistem tidak akan memberikan solusi hama dan penyakit apabila hasil perhitungan similaritas sama dengan atau kurang dari 20 persen. Kemudian sistem akan menampung kedalam tabel *revise* untuk dicari solusi yang tepat. Proses revise dilakukan oleh admin untuk dilakukan input sebagai data hama penyakit mangga yang baru.

DAFTAR PUSTAKA

- [1] Mundhorie, 2015, Sistem Pakar untuk Mengidentifikasi Hama dan Penyakit Durian Menggunakan Metode Cased Based Reasoning (CBR), *Skripsi*, Universitas Stikubank Semarang
- [2] Setiyanto, Y, 2013, Case Based Reasoning untuk Diagnosa Penyakit pada Tanaman Melon, *Skripsi*, Universitas Kristen Duta Wacana Yogyakarta
- [3] Bumi, K.H.C., Azhari, S., 2013, Sistem Pakar Diagnosa Hama dan Penyakit pada Tanaman Mangga, *Skripsi*, Program Studi Ilmu Komputer, Universitas Gadjah Mada Yogyakarta
- [4] Dengen, N., 2015, Identifikasi Penyakit Tanaman Padi Menggunakan Jensen Shannon Divergence Berbasis Android, *Skripsi*, Jurusan S-1 Teknik Informatika, Universitas Halu Oleo, Kendari
- [5] Aamodt, A., Plaza, E., 1994, Case Based Reasoning: Foundation Issues Methodological Variation, and System Approaches, *AI Communication* Vol 7, No 1, March 1994, Pp39-59.
- [6] Cover, T., Hart, P., 1967, Nearest Neighbor Pattern Classification, *IEEE Transactions on Information Theory*, Vol. 13, Ed. 1.
- [7] Choi, S. S., Cha, S. H., & Tappert, C. C. (2010). A Survey of Binary Similarity and Distance Measures, *Journal of Systemics, Cybernetics and Informatics*, 8 (1), 43-48.