

LAPORAN PRAKTIKUM
IMPLEMENTASI SISTEM PAKAR BERBASIS *FORWARD CHAINING* DAN
***CERTAINTY FACTOR* UNTUK MENDIAGNOSA PENYAKIT PADA TANAMAN**
PADI

Dosen Pengampu: Liptia Venica, S.T., M.T.



Oleh :

Zaidan Ahmad (2305172)
Adzka Dzikri Imanullah (2301706)

PROGRAM STUDI MEKATRONIKA DAN KECERDASAN BUATAN
UNIVERSITAS PENDIDIKAN INDONESIA
KAMPUS DAERAH PURWAKARTA
2025

1.1 Acuan Penelitian

Judul	<i>Rice crop management expert system with forwarding chaining method and certainty factor</i>
Penulis	Kharisma and Adi, K and Isnanto, R R
Tahun Terbit	2020
Jurnal	Journal of Physics: Conference Series
DOI	https://doi.org/10.1088/1742-6596/1524/1/012037

1.1.1 Deskripsi Sistem Pakar

Sistem pakar yang dikembangkan oleh Kharisma dkk. (2020) merupakan aplikasi berbasis web yang dirancang untuk membantu petani dalam pengelolaan tanaman padi, khususnya dalam pemilihan varietas benih dan diagnosis hama serta penyakit. Sistem ini menggunakan metode *forward chaining* untuk melakukan penalaran berdasarkan gejala yang diberikan pengguna, serta metode *certainty factor* untuk menghitung tingkat keyakinan hasil diagnosis karena gejala tanaman seringkali memiliki lebih dari satu kemungkinan penyebab. Basis pengetahuan sistem diperoleh dari wawancara dengan pakar pertanian dan literatur Dinas Pertanian. Sistem terdiri atas dua subsistem utama, yaitu pemilihan varietas benih berdasarkan kondisi lahan dan iklim, serta diagnosis penyakit berdasarkan gejala pada daun, batang, atau bulir padi. Hasilnya, sistem mampu memberikan rekomendasi varietas benih dan mendeteksi penyakit dengan tingkat keyakinan tertentu, sehingga dapat membantu petani mengambil keputusan secara cepat dan tepat meskipun tanpa kehadiran langsung seorang pakar.

2.1 Implementasi Ulang

Implementasi memanfaatkan Next.js untuk membangun UI input gejala yang intuitif, termasuk komponen form dan validasi sederhana agar pengguna memilih gejala yang relevan. Integrasi dengan mesin inferensi dilakukan melalui pemanggilan fungsi sinkron/asinkron yang menerima daftar gejala sebagai masukan dan mengembalikan daftar kandidat diagnosa serta rekomendasi. Struktur data dibuat eksplisit agar mudah diuji dan ditelusuri, misalnya objek hasil yang menyertakan aturan pemicu untuk transparansi.

Dari sisi kode, modul inferensi diekspor sebagai fungsi stateless dengan dependensi tunggal pada basis pengetahuan. Pendekatan ini memudahkan pengujian unit terhadap berbagai skenario: tidak ada kecocokan, kecocokan tunggal, dan kecocokan ganda. Antarmuka menampilkan hasil secara prioritas, memberi label tingkat keyakinan sederhana (misalnya berdasarkan jumlah gejala cocok), dan menyediakan tautan atau saran tindakan lanjutan. Pengembangan diarahkan pada kemudahan pemeliharaan rules dan pengalaman pengguna yang lancar.

2.2 Tabel Perbandingan dengan Acuan Penelitian

Aspek	Acuan Penelitian	Implementasi Ulang pada Proyek (Next.js)
Metode Inti	Mengusulkan sistem pakar menggunakan Forward Chaining untuk inferensi dan Certainty Factor (CF) untuk menangani ketidakpastian.	Mengimplementasikan Forward Chaining dan Certainty Factor (CF) sebagai inti dari inference engine.
Cakupan Sistem	Menjelaskan dua subsistem: 1. Diagnosis Hama dan Penyakit 2. Pemilihan Varietas Benih.	Hanya mengimplementasikan 1 subsistem: 1. Diagnosis Hama dan Penyakit. Subsistem pemilihan varietas benih tidak diimplementasikan.
Basis Pengetahuan (Rules)	Menyediakan 9 aturan (R1-R9) untuk diagnosis penyakit, yang memetakan kombinasi fakta (Gejala A dan B) ke kesimpulan (Penyakit P).	Menggunakan 9 aturan (R1-R9) yang identik dengan yang ada di jurnal, beserta nilai CF aturan yang sama persis, yang dimuat dari <code>lib/rules.ts</code> .
Basis Pengetahuan (Facts)	Menyediakan daftar pertanyaan (gejala) yang dipetakan ke ID Fakta (misal A1, A2, B1, B2, dst.) beserta nilai CF pakar.	Menggunakan struktur fakta yang identik (A1, A2, B1, B2, dst.) dengan pertanyaan yang telah diterjemahkan ke Bahasa Indonesia dan nilai CF pakar yang sama.
Kalkulasi CF (Paralel)	Untuk aturan dengan kondisi AND (misal A1 AND B2...), nilai CF paralel dihitung menggunakan MIN dari semua CF fakta. <code>CF_Paralel = MIN(CF(A1), CF(B2), ...).</code>	Mengimplementasikan logika yang sama persis di <code>lib/inference-engine.ts</code> : <code>cfParallel = cfValues.reduce((acc, v) => Math.min(acc, v), 1).</code>
Kalkulasi CF (Sequential)	CF sekuensial (CF aturan yang diterapkan) dihitung dengan mengalikan CF Paralel dengan CF Aturan itu sendiri. <code>CF_Sekuensial =</code>	Mengimplementasikan logika yang sama persis di <code>lib/inference-engine.ts</code> : <code>cfSequential = cfParallel * cfRule.</code>

	CF_Paralel * CF_Aturan.	
Kalkulasi CF (Kombinasi)	Jurnal ini tidak secara eksplisit menunjukkan formula kombinasi MYCIN untuk menggabungkan beberapa aturan yang menghasilkan kesimpulan yang sama. Jurnal hanya menunjukkan perhitungan untuk satu aturan tunggal.	Implementasi <code>lib/inference-engine.ts</code> lebih lengkap dengan menyertakan fungsi <code>calculateCF</code> yang menggunakan formula kombinasi MYCIN untuk menggabungkan beberapa nilai CF (jika ada lebih dari satu aturan yang mendukung diagnosis yang sama).
Teknologi & UI	Dideskripsikan sebagai aplikasi berbasis web. Gambar screenshot (Gambar. 2, 3, 4) menunjukkan UI web yang fungsional namun bergaya tradisional.	Dibangun ulang (dimigrasi) menggunakan tumpukan teknologi modern: Next.js 14, React, dan TypeScript.
Interaksi Pengguna (Input)	Jurnal tidak merinci metode input, tetapi UI menyiratkan pemilihan gejala.	Menggunakan slider interaktif (Tidak/Ragu/Ya) yang memetakan jawaban ke nilai CF 0%, 50%, atau 100% dari CF fakta.
Interaksi Pengguna (Navigasi)	UI Jurnal tampak menampilkan semua pertanyaan atau hasil dalam satu halaman.	Menggunakan manajemen sesi (pagination), di mana pertanyaan dibagi menjadi beberapa halaman (6 pertanyaan per sesi) untuk UX yang lebih baik.

2.3 Tabel Rule Base Sistem Pakar Tanaman Padi

Kode Rule	Fakta (IF)	Kesimpulan (THEN)	Nilai CF Rule	Nama Penyakit (note)
R1	A1, B2, B8, B11, B17	P01	0.70	Hawar daun bakteri (<i>Bacterial leaf blight</i>)
R2	A2, B4, B8, B17	P02	0.80	Penyakit BLAS (<i>Blast disease</i>)
R3	A3, B5, B8, B21	P03	0.70	Bercak cokelat sempit (<i>Narrow brown leaf spot</i>)

Kode Rule	Fakta (IF)	Kesimpulan (THEN)	Nilai CF Rule	Nama Penyakit (note)
R4	A5, B8, B12, B14, B15	P04	0.90	Wereng hijau & tungro (<i>Green leafhopper and tungro virus</i>)
R5	A6, B1, B8, B12	P05	0.90	Wereng cokelat (<i>Brown planthopper</i>)
R6	A7, B8, B9, B16	P06	0.90	Penggerek batang padi (<i>Rice stem borer</i>)
R7	A8, B7, B13, B15, B18	P07	0.70	Kepik hitam padi (<i>Black rice bug</i>)
R8	A9, B19, B20	P08	0.80	Tikus sawah (<i>Rice field rat</i>)
R9	A10, B10	P09	0.80	Kepik busuk padi (<i>Stinky rice bug</i>)

Kode Penyakit	Nama Penyakit	Penyebab / Organisme
P01	Hawar daun bakteri	<i>Xanthomonas oryzae</i>
P02	Penyakit BLAS	<i>Pyricularia grisea</i>
P03	Bercak cokelat sempit	<i>Cercospora oryzae Miyake</i>
P04	Wereng hijau & tungro	<i>Rice tungro bacilliform virus</i>
P05	Wereng cokelat	<i>Nilaparvata lugens</i>
P06	Penggerek batang padi	<i>Scirphophaga incertulas</i>
P07	Kepik hitam padi	<i>Scotinophara coarctata</i>
P08	Tikus sawah	<i>Rattus argentiventer</i>
P09	Kepik busuk padi	<i>Leptocoris acuta</i>

3.1 Proses Inference Engine

3.1.1 Pengumpulan Fakta Awal

Proses dimulai saat pengguna memberikan input melalui antarmuka (UI).

1. Penerjemahan Input: Pengguna tidak memasukkan angka CF secara langsung. Mereka memilih salah satu dari tiga opsi pada *slider* untuk setiap pertanyaan (gejala).

2. Konversi ke CF: Jawaban pengguna tersebut kemudian dikonversi menjadi nilai CF numerik oleh sistem (dapat dilihat di `pages/index.tsx`):
 - o "Ya" (1): Menggunakan 100% dari nilai CF pakar.
 - Contoh: Jika CF pakar untuk fakta A1 adalah 0.95, jawaban "Ya" akan disimpan sebagai $A1 = 0.95$.
 - o "Ragu" (0.5): Menggunakan 50% dari nilai CF pakar.
 - Contoh: $A1 = 0.95 * 0.5 = 0.475$.
 - o "Tidak" (0): Menggunakan nilai CF 0.
 - Contoh: $A1 = 0$.
3. Working Memory: Kumpulan semua jawaban pengguna ($\{A1: 0.95, A2: 0, A3: 0.475, \dots\}$) membentuk "fakta awal" atau yang disebut sebagai Working Memory.

3.1.2 Proses Inferensi Forward Chaining

Setelah fakta awal terkumpul, mesin inferensi (`lib/inference-engine.ts`) mulai bekerja dalam sebuah siklus (loop).

1. Mulai Siklus: Mesin inferensi memulai *loop* (di kode `while (newFacts)`). Tujuannya adalah untuk menemukan aturan (rules) mana yang dapat "dieksekusi" (fired) berdasarkan fakta yang ada di Working Memory.
2. Pencocokan Aturan (Rule Matching): Sistem memeriksa *setiap* aturan (R1 hingga R9) satu per satu.
3. Evaluasi Kondisi (IF): Untuk setiap aturan, sistem memeriksa apakah *semua* kondisi (fakta if) yang diperlukan oleh aturan tersebut sudah ada di dalam Working Memory.
 - o Contoh: Untuk Rule R1: IF [A1, B2, B8, B11, B17] THEN P01, sistem akan bertanya: "Apakah saya memiliki nilai untuk A1, B2, B8, B11, dan B17 di Working Memory saya?"
4. Eksekusi Aturan: Jika *semua* kondisi terpenuhi, aturan tersebut "dieksekusi". Di sinilah perhitungan Certainty Factor (CF) dimulai.

3.1.3 Kalkulasi Certainty Factor

Saat sebuah aturan dieksekusi, sistem menghitung seberapa yakin kesimpulan (penyakit) tersebut berdasarkan fakta-fakta pendukungnya. Ini sesuai dengan contoh perhitungan di jurnal. Proses ini memiliki dua langkah utama:

A. Kalkulasi CF Paralel (Menggabungkan Fakta "AND")

Aturan R1 mengharuskan 5 fakta terpenuhi (A1 AND B2 AND B8...). Untuk menggabungkan CF dari beberapa fakta dalam satu aturan (operasi AND), metode yang digunakan adalah mengambil nilai terkecil dari semua CF fakta tersebut.

- Contoh (mengikuti Jurnal):

- Asumsikan pengguna menjawab "Ya" untuk semua 5 gejala R1.
- CF Fakta (dari Working Memory) adalah:
 - $CF(A1) = 0.95$
 - $CF(B2) = 0.85$
 - $CF(B8) = 0.75$
 - $CF(B11) = 0.70$
 - $CF(B17) = 0.95$
- $CF_{Paralel} = \text{MIN}(0.95, 0.85, 0.75, 0.70, 0.95)$
- $CF_{Paralel} = 0.70$

B. Kalkulasi CF Sekuensial (Menggabungkan Fakta dengan Aturan)

Nilai $CF_{Paralel}$ (0.70) tadi menunjukkan keyakinan gabungan dari gejalanya. Sekarang, nilai ini harus dikombinasikan dengan nilai keyakinan dari aturan itu sendiri (CF_{Rule}). Ini dilakukan dengan perkalian.

- Contoh (lanjutan):

- Nilai CF_{Rule} untuk R1 (Hawar daun bakteri) adalah 0.70.
- $CF_{Sekuensial} = CF_{Paralel} * CF_{Rule}$
- $CF_{Sekuensial} = 0.70 * 0.70$
- $CF_{Sekuensial} = 0.49$

3.1.4 Pembaruan Working Memory dan Siklus Lanjutan

1. Fakta Baru: Nilai $CF_{Sekuensial}$ (0.49) ini adalah fakta baru. Sistem sekarang menambahkan kesimpulan $P01$ (Hawar daun bakteri) dengan keyakinan 0.49 ke dalam Working Memory.
2. Pengecekan Ulang: Karena ada fakta baru ($P01 = 0.49$), sistem menandai bahwa Working Memory telah diperbarui ($newFacts = true$). Ini menyebabkan siklus Forward Chaining (Tahap 2) diulangi kembali dari awal.

3. Siklus Berikutnya: Sistem kembali memeriksa semua aturan (R1-R9).
4. Kombinasi MYCIN (jika ada): Jika ada aturan lain (misal, R10) yang juga menghasilkan kesimpulan P01, fungsi `calculateCF` di `lib/inference-engine.ts` akan menggabungkan CF lama (0.49) dengan CF baru dari R10 menggunakan formula MYCIN untuk mendapatkan nilai keyakinan yang lebih kuat.
5. Berhenti: Siklus akan terus berulang sampai tidak ada lagi aturan baru yang dapat dieksekusi, yaitu ketika satu putaran penuh pemeriksaan aturan tidak menghasilkan fakta baru (`newFacts = false`).

3.1.5 Diagnosis

Setelah siklus Forward Chaining berhenti, proses inferensi selesai.

1. Pengumpulan Hasil: Sistem mengumpulkan semua fakta kesimpulan (P01, P02, dst.) yang ada di Working Memory.
2. Pelabelan: ID Penyakit (misal P01) dipetakan ke nama lengkapnya ("Hawar daun bakteri (*Xanthomonas oryzae*)") menggunakan `rulesData.labels`.
3. Pengurutan: Hasil diagnosis diurutkan dari nilai CF tertinggi ke terendah.
4. Output: Hasil akhir (misal: "Hawar daun bakteri: 49%") ditampilkan kepada pengguna.

Gambar 1 Form Pengisian Fakta Awal



Gambar 2 Hasil Diagnosa

3.2 Kesimpulan

Berdasarkan praktikum yang telah dilaksanakan, sistem pakar untuk diagnosis penyakit padi telah berhasil diimplementasikan dengan mengadopsi metode *forward chaining* sebagai mesin inferensi dan *certainty factor* (CF) untuk mengelola ketidakpastian. Sistem terbukti mampu melakukan penalaran dari sekumpulan fakta (gejala) yang diberikan untuk menghasilkan kesimpulan (diagnosis penyakit) beserta persentase keyakinan yang terukur. Implementasi basis pengetahuan dan alur kalkulasi CF pada sistem ini telah sesuai dengan metodologi yang diuraikan dalam jurnal acuan.