

**LAPORAN IMPLEMENTASI ULANG SISTEM PAKAR BERBASIS FORWARD  
CHAINING**

**Sistem Pakar Untuk Mendiagnosis Penyakit Pada Tanaman Cabai Merah**

*Dosen pengampu : Liptia Venica, S.T., M.T.*



Dibuat oleh:

Fathir Raihan	2310051
Muhamad Reyvanatha Hernawan	2303965

Tanggal: 27 Oktober 2025

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

## DAFTAR ISI

1. Judul dan Sumber Penelitian Acuan .....	3
1.1 Latar Belakang Penelitian Asli .....	3
1.2 Tujuan Penelitian Asli.....	3
1.3 Alasan Pemilihan Penelitian Ini .....	4
2. Deskripsi Sistem Pakar pada Penelitian Asli .....	4
2.1 Konsep Umum Sistem Pakar .....	4
2.2 Komponen Utama Sistem .....	5
3. Penjelasan Implementasi Ulang .....	6
3.1 Tujuan Implementasi Ulang.....	6
3.2 Struktur dan Komponen Program .....	6
3.3 Antarmuka Pengguna (UI).....	7
4. Tabel Aturan (Rule Base) dan Representasi Json .....	7
4.1 Tabel Aturan (Rule Base) .....	7
4.3 Representasi Json .....	10
4.3 Penjelasan Representasi JSON .....	12
5. Proses Inferensi (Forward Chaining + Certainty Factor).....	12
5.1 Mekanisme Forward Chaining.....	12
5.2 Perhitungan Certainty Factor (CF).....	13
5.3 Ringkasan Alur Kerja Sistem.....	14
6. Hasil Uji Coba.....	14
7. Kesimpulan .....	17

## 1. Judul dan Sumber Penelitian Acuan

Penelitian yang dijadikan acuan dalam implementasi ulang sistem pakar ini berjudul:

“Expert System with Certainty Factor for Early Diagnosis of Red Chili Peppers Diseases”

Penelitian tersebut ditulis oleh Fahrul Agus, Achmad Solichin, dan Riza Alfita, dan diterbitkan dalam Jurnal Telematika MKOM, Volume 9 Nomor 2 Tahun 2017 oleh Universitas AMIKOM Yogyakarta.

Sebagai referensi tambahan, digunakan pula literatur dari Universitas Lampung berjudul: “Pengendalian Virus Kuning Keriting Cabai (Gemini Virus)”, Repository LPPM Unila (2019). Sumber:

<http://repository.lppm.unila.ac.id/15839/1/PFI%20BANJARMASIN%20FINAL%20Pengendalian%20%20Virus%20Kuning%20Keriting%20Cabai.pdf>

### 1.1 Latar Belakang Penelitian Asli

Tanaman cabai merah merupakan salah satu komoditas pertanian penting di Indonesia, namun sangat rentan terhadap berbagai penyakit seperti antraknosa, layu fusarium, busuk buah, dan infeksi virus kuning. Penyakit-penyakit tersebut dapat menyebabkan penurunan hasil panen secara signifikan.

Permasalahan yang sering muncul adalah sulitnya petani dalam mengenali jenis penyakit berdasarkan gejala awal, karena gejala antarpenyakit sering kali mirip. Sementara itu, ketersediaan tenaga ahli pertanian terbatas, terutama di daerah produksi cabai yang terpencil.

Untuk menjawab masalah tersebut, Fahrul Agus dkk. mengembangkan sistem pakar (expert system) berbasis web yang mampu membantu petani mendiagnosis penyakit cabai merah berdasarkan gejala yang diamati. Sistem ini menggunakan metode Forward Chaining untuk melakukan proses penalaran dan Certainty Factor (CF) untuk menghitung tingkat keyakinan hasil diagnosis.

Dengan adanya sistem pakar ini, petani dapat memperoleh diagnosis awal secara cepat, mudah, dan akurat tanpa perlu bergantung langsung pada pakar lapangan.

### 1.2 Tujuan Penelitian Asli

Tujuan utama penelitian acuan adalah:

1. Membangun sistem pakar berbasis web untuk diagnosis penyakit tanaman cabai merah.
2. Menerapkan metode Certainty Factor (CF) dalam menentukan tingkat keyakinan diagnosis berdasarkan kombinasi nilai dari pakar dan pengguna.
3. Menerapkan metode Forward Chaining untuk melakukan penalaran logis dari fakta (gejala) menuju kesimpulan (penyakit).
4. Memberikan antarmuka yang mudah digunakan bagi petani untuk memilih gejala dan melihat hasil diagnosis penyakit tanaman.

### 1.3 Alasan Pemilihan Penelitian Ini

Penelitian ini dipilih sebagai acuan karena memenuhi seluruh kriteria tugas, yaitu:

- Menggunakan pendekatan rule-based reasoning (berbasis aturan).
- Mengimplementasikan metode forward chaining dalam proses inferensi.
- Menggunakan Certainty Factor (CF) untuk mengukur tingkat keyakinan hasil diagnosis.
- Mempunyai topik yang jelas dan aplikatif, yaitu diagnosis penyakit tanaman cabai merah, yang relevan dengan bidang sistem pakar pertanian.

Selain itu, penelitian ini memiliki struktur pengetahuan yang mudah diadaptasi ke format JSON, sehingga cocok untuk diimplementasikan ulang menggunakan JavaScript dan antarmuka web modern.

## 2. Deskripsi Sistem Pakar pada Penelitian Asli

### 2.1 Konsep Umum Sistem Pakar

Sistem pakar (*expert system*) merupakan salah satu cabang dari kecerdasan buatan (*Artificial Intelligence*) yang dirancang untuk meniru kemampuan dan cara berpikir seorang pakar manusia dalam menyelesaikan masalah di bidang tertentu.

Penelitian acuan yang digunakan berjudul “Expert System with Certainty Factor for Early Diagnosis of Red Chili Peppers Diseases” oleh Fahrul Agus dkk. (2017). Penelitian tersebut membangun sistem pakar untuk membantu petani mendiagnosis penyakit tanaman

cabai merah berdasarkan gejala yang terlihat. Sistem ini menggunakan metode rule-based reasoning, dengan proses forward chaining sebagai mekanisme penalaran dan certainty factor (CF) untuk menghitung tingkat keyakinan terhadap hasil diagnosis.

Dalam penelitian asli, sistem pakar terdiri atas tiga komponen utama, yaitu:

1. Basis Pengetahuan, berisi aturan *if-then* yang menghubungkan gejala dengan penyakit berdasarkan konsultasi dengan pakar pertanian.
2. Mesin Inferensi, yang melakukan penalaran maju dari gejala menuju penyakit dengan menggunakan rumus perhitungan *certainty factor*.
3. Antarmuka Pengguna, yang berfungsi sebagai media interaksi antara pengguna dengan sistem untuk memilih gejala dan menampilkan hasil diagnosis.

Penelitian ini menjadi acuan karena telah menerapkan metode forward chaining dan certainty factor secara lengkap pada kasus nyata, serta memberikan kerangka dasar yang mudah direplikasi dan dikembangkan lebih lanjut.

## 2.2 Komponen Utama Sistem

Penelitian asli menyebutkan tiga komponen pokok dalam perancangan sistem pakar:

1. Basis Pengetahuan (Knowledge Base). Menyimpan fakta dan aturan yang digunakan untuk melakukan penalaran. Fakta berupa daftar gejala yang mungkin terjadi pada tanaman cabai merah, sedangkan aturan menggambarkan hubungan antara gejala dan penyakit.
2. Mesin Inferensi (Inference Engine). Bertugas melakukan proses *penalaran maju (forward chaining)*, yaitu mencari kesimpulan (penyakit) berdasarkan fakta (gejala) yang dimasukkan pengguna. Mesin ini akan memeriksa setiap aturan dalam basis pengetahuan dan menentukan aturan mana yang terpenuhi oleh fakta.
3. Antarmuka Pengguna (User Interface). Menyediakan media bagi pengguna untuk berinteraksi dengan sistem. Pengguna dapat memilih gejala-gejala yang muncul dan melihat hasil diagnosis berupa nama penyakit dan tingkat keyakinannya.

### **3. Penjelasan Implementasi Ulang**

#### **3.1 Tujuan Implementasi Ulang**

Implementasi ulang dilakukan dengan tujuan mereplikasi mekanisme sistem pakar dari penelitian acuan menggunakan format JSON dan bahasa pemrograman JavaScript. Sistem dirancang untuk tetap mempertahankan konsep utama penelitian asli, namun dengan beberapa penyesuaian agar sesuai dengan ketentuan tugas dan struktur proyek modern.

Penyesuaian yang dilakukan meliputi:

1. Struktur Basis Pengetahuan disimpan dalam tiga file JSON, yaitu symptoms.json, rules.json, dan diseases.json, agar data lebih terpisah dan mudah dipelihara.
2. Nilai CF Pakar diletakkan pada gejala di symptoms.json, sedangkan rules.json hanya berisi hubungan antar gejala dan penyakit.
3. Penambahan Aturan Paralel dan Sekuensial, sesuai ketentuan tugas, berdasarkan literatur tambahan yang relevan agar sistem mampu menangani hubungan penyakit yang saling berkaitan.
4. Mesin Inferensi dikembangkan dalam file engine.js untuk menjalankan proses penalaran otomatis dan perhitungan CF, sedangkan antarmuka pengguna dibuat menggunakan HTML, CSS, dan JavaScript sederhana.

Dengan implementasi ulang ini, sistem mampu melakukan inferensi secara otomatis, menghitung nilai keyakinan penyakit berdasarkan gejala yang dipilih pengguna, dan menampilkan hasil diagnosis dengan antarmuka yang mudah digunakan.

#### **3.2 Struktur dan Komponen Program**

Aplikasi sistem pakar yang dikembangkan memiliki tiga komponen utama sebagaimana pada penelitian asli:

1. Basis Pengetahuan (Knowledge Base). Terdiri dari tiga file JSON:
  - symptoms.json → berisi daftar gejala beserta nilai CF pakar.
  - rules.json → berisi kumpulan aturan (rule base) yang menghubungkan gejala dan penyakit.

- diseases.json → berisi daftar penyakit dengan kode dan nama lengkapnya.
2. Inference Engine (engine.js). Berisi logika *forward chaining* untuk mencocokkan fakta (gejala) yang dipilih dengan aturan, sekaligus menghitung nilai CF final berdasarkan kombinasi CF pakar dan CF pengguna.
  3. User Interface (ui folder). Berisi file index.html, app.js, dan style.css yang memungkinkan pengguna memilih gejala, memberikan tingkat keyakinan (CF user), dan melihat hasil diagnosis penyakit.

### 3.3 Antarmuka Pengguna (UI)

Antarmuka sistem dibangun menggunakan HTML, CSS, dan JavaScript, dengan tiga tahapan utama:

1. Pemilihan Gejala – pengguna mencentang gejala yang muncul.
2. Penentuan Keyakinan (CF User) – pengguna memilih tingkat keyakinan untuk setiap gejala.
3. Hasil Diagnosis – sistem menampilkan daftar penyakit dengan nilai CF masing-masing.

UI ini sepenuhnya berjalan di sisi klien (*client-side*), sehingga mudah dijalankan hanya dengan browser tanpa instalasi tambahan.

## 4. Tabel Aturan (Rule Base) dan Representasi Json

### 4.1 Tabel Aturan (Rule Base)

Berikut adalah tabel aturan yang digunakan dalam sistem:

Table 1. Symptoms of Antraknosa (Framboesia)

Code		Name	CF
K001	G09	Buah memiliki bintik hitam dan membusuk	0,8
	G08	Buah mengering dan keriput	0,6
	G07	Bercak kering pada daun	0,6
	G14	Daun menguning	0,4
	G10	Daun mati/rontok	0,2

Table 2. Symptoms of Cercospora Leaves Patches

Code		Name	CF
K002	G06	Bercak bulat di daun dan sobek	0,8
	G07	Bercak kering pada daun	0,6
	G05	Bercak abu-abu kecoklatan pada daun	0,4
	G12	Daun hijau dan mengkilat	0,2
	G10	Daun mati/rontok	0,2

Table 3. Symptoms of Fusarium Withered

Code		Name	CF
K003	G17	Tanaman layu mendadak	0,8
	G01	Akar membusuk dan berwarna kecoklatan	0,6
	G14	Daun menguning	0,4
	G20	Daun memutih dan menebal	0,2

Table 4. Symptoms of Gemini Virus / Yellow Virus

Code		Name	CF
K004	G04	Batang menguning	0,8
	G13	Daun menggulung ke atas	0,6
	G14	Daun menguning	0,4
	G20	Daun memutih dan menebal	0,2
	G16	Tanaman kerdil	0,2
	G18	Tanaman tidak berbuah	0,2

Table 5. Symptoms of Phytophthora Foul

Code		Name	CF
K005	G03	Batang dan daun berwarna kusam/gelap	0,8
	G15	Kulit batang mudah terkelupas	0,8
	G01	Akar membusuk dan berwarna kecoklatan	0,6



	G19	Tangkai berwarna coklat kehitaman	0,4
	G02	Buah dan bunga rontok/mati	0,2

Table 6. Symptoms of Mosaic

Code		Name	CF
K006	G11	Daun belang (hijau tua & hijau muda/mosaik)	0,8
	G22	Ukuran daun lebih kecil	0,6
	G21	Beberapa Daun menguning	0,4
	G16	Tanaman kerdil	0,2

Namun, dalam versi implementasi ulang ini, ditambahkan dua aturan tambahan agar sistem mengandung *hubungan paralel dan sekuensial*, sesuai ketentuan tugas:

1. Aturan Paralel (Tambahan). Menggambarkan bahwa penyakit yang sama (K001) dapat disebabkan oleh kombinasi gejala berbeda.

```
{
  "id": "R_K001_B (Tambahan & Paralel - Didukung Ref.)",
  "if": ["G09", "G08"],
  "then": "K001"
}
```

2. Aturan Sekuensial (Tambahan). Menggambarkan bahwa hasil dari aturan sebelumnya (K006) dapat menjadi premis pada aturan berikut untuk menghasilkan penyakit lain (K004).

```
{
  "id": "R_SEQ_K004_B (Tambahan & Sekuensial & Paralel - Didukung Ref.)",
  "if": ["K006", "G14"],
  "then": "K004"
}
```

Kedua aturan tambahan ini disusun berdasarkan logika biologis dan literatur pendukung:

*Sumber: "Pengendalian Virus Kuning Keriting Cabai", Repository Universitas Lampung (UNILA), 2018.*

<http://repository.lppm.unila.ac.id/15839/1/PFI%20BANJARMASIN%20FINAL%20Pengendalian%20%20Virus%20Kuning%20Keriting%20Cabai.pdf>

Aturan tersebut menunjukkan bahwa:

- Kombinasi gejala G08 dan G09 sering muncul bersama pada kasus penyakit antraknosa daun cabai (K001).
- Penyakit K006 (mosaik) dapat berkembang menjadi K004 (virus kuning keriting) jika disertai gejala G14 (daun menguning).

#### 4.3 Representasi Json

Basis pengetahuan sistem pakar disimpan dalam format JSON untuk memudahkan pembacaan oleh program JavaScript.

File JSON dibagi menjadi tiga bagian utama:

1. symptoms.json: berisi daftar gejala dan nilai *Certainty Factor (CF)* pakar.

```
[
  {"id": "G01", "name": "Akar membusuk dan berwarna kecoklatan",
  "cf_pakar": 0.6},
  {"id": "G02", "name": "Buah dan bunga rontok/mati", "cf_pakar": 0.2},
  {"id": "G03", "name": "Batang dan daun berwarna kusam/gelap", "cf_pakar":
0.8},
  {"id": "G04", "name": "Batang menguning", "cf_pakar": 0.8},
  {"id": "G05", "name": "Bercak abu-abu kecoklatan pada daun", "cf_pakar":
0.4},
  {"id": "G06", "name": "Bercak bulat di daun dan sobek", "cf_pakar": 0.8},
  {"id": "G07", "name": "Bercak kering pada daun", "cf_pakar": 0.6},
  {"id": "G08", "name": "Buah mengering dan keriput", "cf_pakar": 0.6},
  {"id": "G09", "name": "Buah memiliki bintik hitam dan membusuk",
"cf_pakar": 0.8},
  {"id": "G10", "name": "Daun mati/rontok", "cf_pakar": 0.2},
  {"id": "G11", "name": "Daun belang (hijau tua & hijau muda/mosaik)",
"cf_pakar": 0.8},
  {"id": "G12", "name": "Daun hijau dan mengkilat", "cf_pakar": 0.2},
  {"id": "G13", "name": "Daun menggulung ke atas", "cf_pakar": 0.6},
  {"id": "G14", "name": "Daun menguning", "cf_pakar": 0.4},
```

```
[
  {"id": "G15", "name": "Kulit batang mudah terkelupas", "cf_pakar": 0.8},
  {"id": "G16", "name": "Tanaman kerdil", "cf_pakar": 0.2},
  {"id": "G17", "name": "Tanaman layu mendadak", "cf_pakar": 0.8},
  {"id": "G18", "name": "Tanaman tidak berbuah", "cf_pakar": 0.2},
  {"id": "G19", "name": "Tangkai berwarna coklat kehitaman", "cf_pakar":
0.4},
  {"id": "G20", "name": "Daun memutih dan menebal", "cf_pakar": 0.2},
  {"id": "G21", "name": "Beberapa Daun menguning", "cf_pakar": 0.4},
  {"id": "G22", "name": "Ukuran daun lebih kecil", "cf_pakar": 0.6}
]
```

2. diseases.json – berisi daftar penyakit beserta penjelasan singkatnya.

```
[
  {"id": "K001", "name": "Antraknosa (Framboesia)"},
  {"id": "K002", "name": "Cercospora Leaves Patches"},
  {"id": "K003", "name": "Fusarium Withered"},
  {"id": "K004", "name": "Gemini Virus / Yellow Virus"},
  {"id": "K005", "name": "Phytophthora Foul"},
  {"id": "K006", "name": "Mosaic"}
]
```

3. rules.json – berisi aturan inferensi (*rule base*) yang menghubungkan gejala dengan penyakit.

```
[
  {
    "id": "R_K001_A (Asli & Paralel)",
    "if": ["G09", "G08", "G07", "G14", "G10"],
    "then": "K001"
  },
  {
    "id": "R_K002 (Asli)",
    "if": ["G06", "G07", "G05", "G12", "G10"],
    "then": "K002"
  },
  {
    "id": "R_K003 (Asli)",
    "if": ["G17", "G01", "G14", "G20"],
    "then": "K003"
  },
  {
    "id": "R_K004_A (Asli & Paralel)",
    "if": ["G04", "G13", "G14", "G20", "G16", "G18"],
    "then": "K004"
  },
  {

```

```

    "id": "R_K005 (Asli)",
    "if": ["G03", "G15", "G01", "G19", "G02"],
    "then": "K005"
  },
  {
    "id": "R_K006 (Asli & Pemicu Sekuensial)",
    "if": ["G11", "G22", "G21", "G16"],
    "then": "K006"
  },
  {
    "id": "R_K001_B (Tambahan & Paralel - Didukung Ref.)",
    "if": ["G09", "G08"],
    "then": "K001"
  },
  {
    "id": "R_SEQ_K004_B (Tambahan & Sekuensial & Paralel - Didukung Ref.)",
    "if": ["K006", "G14"],
    "then": "K004"
  }
]

```

### 4.3 Penjelasan Representasi JSON

Setiap aturan disusun dalam bentuk objek yang memiliki tiga elemen:

- id : identitas unik dari aturan.
- if : daftar gejala (atau hasil aturan sebelumnya) yang menjadi premis.
- then : hasil inferensi, berupa penyakit yang didiagnosis.

Dengan format JSON ini, sistem dapat dengan mudah membaca file menggunakan JavaScript, kemudian melakukan proses inferensi dengan metode forward chaining. Aturan tambahan (paralel dan sekuensial) disisipkan untuk memastikan sistem memenuhi spesifikasi tugas dan lebih menyerupai kondisi nyata diagnosis penyakit tanaman.

## 5. Proses Inferensi (Forward Chaining + Certainty Factor)

### 5.1 Mekanisme Forward Chaining

Metode forward chaining bekerja berdasarkan prinsip “*data-driven inference*”, yaitu proses penalaran yang dimulai dari fakta atau data yang diberikan pengguna menuju kesimpulan.

Tahapan penalaran dalam penelitian asli adalah sebagai berikut:

1. Pengguna memilih gejala yang sesuai dengan kondisi tanaman cabai merah.
2. Sistem mencocokkan gejala yang dimasukkan dengan aturan di basis pengetahuan.
3. Jika semua gejala dalam satu aturan terpenuhi, maka aturan tersebut diaktifkan (fired) dan menghasilkan dugaan penyakit.
4. Bila terdapat lebih dari satu aturan yang aktif, sistem akan menghitung nilai Certainty Factor (CF) untuk menentukan penyakit yang paling mungkin.

Sistem menampilkan hasil diagnosis akhir berupa nama penyakit beserta nilai keyakinan dalam bentuk persentase

## 5.2 Perhitungan Certainty Factor (CF)

Metode Certainty Factor (CF) digunakan untuk mengukur tingkat keyakinan sistem terhadap hasil diagnosis berdasarkan kombinasi keyakinan pakar dan pengguna.

Tahap perhitungannya adalah sebagai berikut:

1. Menentukan CF Gejala:

$$CF_{gejala} = CF_{pakar} \times CF_{user}$$

Di mana:

1.  $CF_{pakar}$  = tingkat keyakinan pakar terhadap hubungan gejala–penyakit.
2.  $CF_{user}$  = tingkat keyakinan pengguna terhadap kemunculan gejala.
2. Menggabungkan CF antar Gejala: Jika suatu penyakit terdiri dari beberapa gejala, nilai CF gabungan dihitung menggunakan rumus:

$$CF_{combine} = CF_1 + CF_2(1 - CF_1)$$

Rumus ini diterapkan berulang untuk seluruh gejala dalam satu aturan.

3. Menentukan CF Akhir (CF Final): Jika terdapat beberapa aturan berbeda yang menghasilkan penyakit yang sama (aturan paralel), maka nilai CF-nya juga digabungkan dengan rumus yang sama. Hasil akhirnya berupa nilai CF final yang menunjukkan tingkat keyakinan sistem terhadap diagnosis penyakit tersebut.

### 5.3 Ringkasan Alur Kerja Sistem

Secara umum, sistem pakar dari penelitian Fahrul Agus dkk. (2017) bekerja melalui langkah-langkah berikut:

1. Input Gejala: pengguna memilih gejala yang diamati pada tanaman.
2. Perhitungan Awal CF: sistem mengalikan CF pakar dan CF user untuk setiap gejala.
3. Penerapan Aturan Forward Chaining: sistem memeriksa aturan dalam basis pengetahuan dan menelusuri fakta yang sesuai.
4. Perhitungan CF Gabungan: semua gejala dalam aturan yang aktif digabung untuk menentukan nilai keyakinan penyakit.
5. Output Diagnosis: sistem menampilkan nama penyakit dengan nilai CF dalam persentase (%). Penyakit dengan CF tertinggi dianggap paling mungkin diderita tanaman.

### 6. Hasil Uji Coba

Pada tampilan awal pengguna (*user*) akan diarahkan pada halaman pembuka yang berisi penjelasan fungsi aplikasi.




Gambar 1.1 Tampilan Awal

Selanjutnya pengguna menekan tombol ‘Mulai Diagnosis’ untuk menggunakan aplikasi. Pada halaman selanjutnya pengguna (*user*) akan diminta untuk memilih beberapa gejala yang ada pada penyakit cabai merah. Total gejala yang bisa dipilih ada 22 gejala.

 semua gejala yang Anda amati pada tanaman cabai.' (Check ☒ all symptoms you observe on the chili plant). There is a scrollable list of five symptoms, each with a checkbox and a label: 'Akar membusuk dan berwarna kecoklatan (G01)', 'Buah dan bunga rontok/mati (G02)', 'Batang dan daun berwarna kusam/gelap (G03)', 'Batang menguning (G04)', and 'Bercak abu-abu kecoklatan pada daun (G05)'. At the bottom left, there is a red button with the text 'Lanjut →' (Next →)." data-bbox="233 166 752 444"/>

Gambar 1.2 Memilih Gejala

Setelah memilih gejala apa saja yang ada pada tanaman cabai merah, pengguna menekan tombol ‘Lanjut’. Pada halaman selanjutnya pengguna diminta untuk memilih tingkat keyakinan dari gejala yang diderita tanaman cabai merah. Pada halaman ‘Menentukan Tingkat Keyakinan’, pengguna bisa memilih 5 tingkat keyakinan yakni; Tidak Yakin, Cukup Yakin, Yakin, Sangat Yakin, dan Pasti.



### Langkah 2: Tentukan Tingkat Keyakinan

🤔

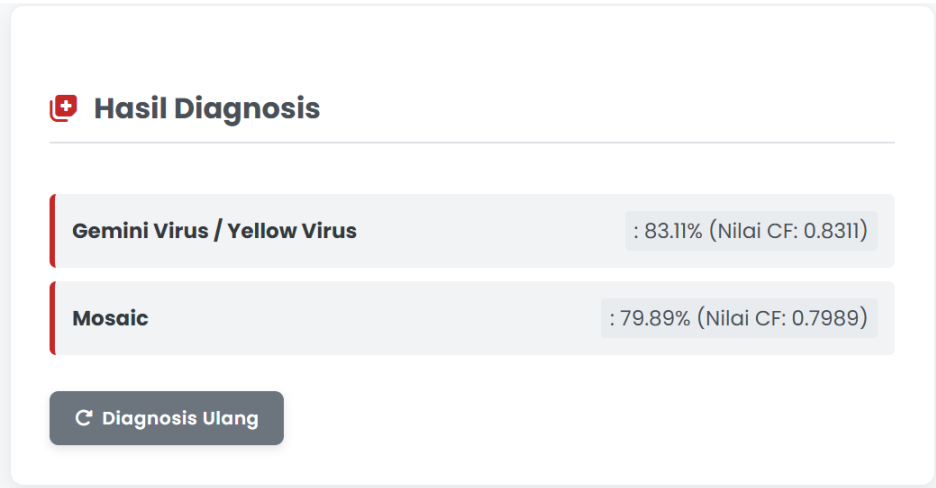
Untuk setiap gejala yang Anda pilih, seberapa yakin Anda gejala tersebut benar-benar ada?

Daun belang (hijau tua & hijau muda/mosaik) (G11)	Tidak Yakin (20%)
Daun menguning (G14)	Cukup Yakin (40%)
Tanaman kerdil (G16)	Yakin (60%)
Beberapa Daun menguning (G21)	Sangat Yakin (80%)

← Kembali   **Lihat Hasil Diagnosis**

Gambar 1.3 Menentukan Tingkat Keyakinan

Setelah pengguna sudah menentukan tingkat keyakinan, selanjutnya pengguna menekan 'Lihat Hasil Diagnosis' untuk melihat hasilnya. Pada halaman 'Hasil Diagnosis' pengguna ditampilkan hasil berupa nama penyakit dan nilai CF nya. Pengguna juga bisa menekan tombol 'Diagnosis Ulang' untuk melakukan diagnosis lagi.



### Hasil Diagnosis

<b>Gemini Virus / Yellow Virus</b>	: 83.11% (Nilai CF: 0.8311)
<b>Mosaic</b>	: 79.89% (Nilai CF: 0.7989)

**Diagnosis Ulang**

Gambar 1.4 Hasil Diagnosis

Pada pengujian yang dilakukan, sistem berhasil menampilkan hasil diagnosis dengan nilai keyakinan yang konsisten terhadap kombinasi gejala yang diberikan. Hal ini menunjukkan bahwa mekanisme inferensi dan perhitungan CF yang diterapkan pada *engine* bekerja sesuai dengan teori dan metode pada penelitian acuan. Selain itu, antarmuka pengguna juga berjalan



dengan baik dan responsif, memudahkan pengguna dalam memilih gejala serta memahami hasil yang ditampilkan.

## 7. Kesimpulan

Sistem pakar diagnosis penyakit tanaman cabai merah yang telah diimplementasikan ulang ini berhasil mereplikasi mekanisme penelitian acuan dengan baik menggunakan metode rule-based reasoning, forward chaining, dan certainty factor. Sistem dirancang menggunakan format JSON sebagai basis pengetahuan dan diimplementasikan dengan bahasa JavaScript, sehingga memiliki struktur yang sederhana namun efektif untuk melakukan proses inferensi dan perhitungan nilai keyakinan (*certainty factor*).

Proses inferensi berjalan sesuai harapan, di mana sistem mampu menelusuri gejala yang dimasukkan pengguna hingga menghasilkan diagnosis penyakit dengan nilai keyakinan yang akurat. Kombinasi nilai CF pakar dan CF pengguna memberikan hasil yang lebih realistis serta dapat mencerminkan tingkat keyakinan terhadap kemungkinan penyakit yang terjadi.

Penambahan aturan paralel dan sekuensial berdasarkan literatur relevan juga berhasil memperkaya basis pengetahuan sistem, menjadikannya lebih fleksibel dalam menangani hubungan antarpenyakit yang saling berkaitan. Antarmuka pengguna yang dibuat secara sederhana turut mendukung kemudahan penggunaan sistem ini, baik oleh pengguna umum maupun pelajar yang ingin memahami cara kerja sistem pakar berbasis *certainty factor*.

Secara keseluruhan, sistem pakar hasil implementasi ulang ini tidak hanya berfungsi sebagaimana penelitian aslinya, tetapi juga telah disesuaikan dengan format dan ketentuan tugas yang menekankan penggunaan JSON, modularitas kode, serta penerapan logika inferensi yang transparan. Sistem ini dapat dijadikan dasar untuk pengembangan lanjutan, seperti penambahan data penyakit baru, integrasi dengan sistem berbasis IoT, maupun penerapan dalam bentuk aplikasi mobile untuk mendukung pertanian cerdas (*smart farming*).