

**IMPLEMENTASI ULANG SISTEM PAKAR DIAGNOSA DEPRESI
MENGUNAKAN FORWARD CHAINING DAN CERTAINTY FACTOR**

Disusun untuk memenuhi tugas pada mata kuliah Machine Learning

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



Disusun oleh:

Ahmad Yasin Firmansyah 2304483

Muhammad Ghailan Dena 2300561

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

2025

A. Judul dan Sumber Penelitian Acuan

Judul Asli: Expert System to Diagnose Diseases of Mental Health with Forward Chaining and Certainty Factor

Sumber (Jurnal): Wahana Matematika dan Sains: Jurnal Matematika, Sains, dan Pembelajarannya, Vol. 14 No 1, April 2020

Penulis: NK Ariasih dan I GAAD Indradewi

B. Deskripsi Sistem Pakar pada Penelitian Asli

Penelitian asli bertujuan untuk mengembangkan sebuah sistem pakar yang dapat membantu pengguna (pasien) mendiagnosis masalah kesehatan mental, dengan fokus khusus pada kategori depresi.

Tujuan: Membantu diagnosis dini depresi berdasarkan gejala yang dirasakan pasien.

Metodologi:

- a) **Representasi Pengetahuan:** Menggunakan aturan produksi (Production Rule) dalam bentuk IF-THEN.
- b) **Metode Inferensi:** Menggunakan Forward Chaining (penalaran berbasis data) untuk menarik kesimpulan. Proses dimulai dari fakta (gejala yang dimasukkan) untuk mencapai kesimpulan (diagnosis penyakit)
- c) **Penanganan Ketidakpastian:** Menerapkan metode Certainty Factor (CF) untuk mengatasi ketidakpastian, baik dari pakar maupun dari pengguna.

Lingkup Bahasan: Penelitian membatasi diagnosis pada empat jenis depresi: Depresi Vegetatif (D1), Depresi Agitasi (D2), Depresi Distimik (D3), dan Depresi Psikotik (D4)

Basis Pengetahuan: Basis pengetahuan sistem terdiri dari 23 gejala (G1 s/d G23) dan aturan-aturan yang berbeda untuk pasien Pria dan Wanita. Perbedaan ini didasarkan pada fakta bahwa depresi pada wanita dapat dipengaruhi oleh siklus hormonal (misal G1: Sebelum siklus menstruasi, G11: Kehilangan siklus menstruasi).

Platform: Sistem asli dikembangkan sebagai sistem berbasis web (web-based)

C. Penjelasan Implementasi Ulang (Perubahan dan Penyesuaian)

Implementasi ulang ini mereplikasi fungsionalitas inti dari penelitian acuan, namun dengan beberapa penyesuaian teknis dan modifikasi untuk memenuhi spesifikasi tugas.

Platform dan Bahasa: Sistem diimplementasikan ulang menggunakan bahasa pemrograman Python. Antarmuka pengguna (UI) dibangun menggunakan library Tkinter, berbeda dari penelitian asli yang berbasis web.

Basis Pengetahuan (rules.json): Sesuai arahan tugas, basis pengetahuan (gejala, penyakit, dan aturan) disimpan dalam format file JSON (rules.json). Struktur JSON ini mencakup:

- 1 gejala: Daftar 23 gejala beserta “cf_pakar” (CF dari pakar).
- 2 penyakit: Daftar 4 jenis depresi (D1-D4).
- 3 aturan_pria dan aturan_wanita: Aturan dipisahkan berdasarkan gender.

Input CF Pengguna: Penelitian asli menyebutkan pengguna memasukkan nilai keyakinan. Implementasi UI menyederhanakan ini menjadi 5 pilihan level keyakinan (Pasti Ada, Mungkin Ada, Tidak Tahu, Mungkin Tidak, Pasti Tidak). Pilihan ini kemudian dipetakan ke nilai CF pengguna (misal: "Pasti Ada" = 0.8, "Mungkin Tidak" = -0.4).

Modifikasi Aturan (Penyesuaian Tugas): Arahan tugas mewajibkan adanya aturan sekuensial dan/atau paralel

- A. **Aturan Sekuensial:** Penelitian asli tidak secara eksplisit menunjukkan aturan sekuensial (aturan yang menggunakan kesimpulan aturan lain). Dalam implementasi ulang ini, ditambahkan aturan baru (dalam rules.json di bawah key "aturan") untuk mendiagnosis komplikasi (K1, K2). Aturan ini (misal R8: JIKA D1 AND D3 MAKA K1) bersifat sekuensial karena premisnya (D1, D3) adalah hasil (kesimpulan) dari aturan diagnosis sebelumnya.
- B. **Aturan Paralel:** Arahan tugas dan penelitian asli membahas penanganan aturan paralel (dua aturan berbeda yang menghasilkan kesimpulan sama). Inference engine (engine.py) telah dilengkapi dengan fungsi

combine_cf_parallel untuk menangani skenario ini, sesuai dengan Persamaan pada jurnal.

D. Tabel Aturan (Rule Base) dan Representasi JSON

Basis pengetahuan yang digunakan merupakan adaptasi langsung dari penelitian acuan.

1. Rule Base

Group	Id	Jika	Maka	Cf_Rule
Pria	R1_P	G2; G3; G4	D1	0.8
Pria	R2_P	G2; G5; G6; G7; G8; G9; G10	D2	0.9
Pria	R3_P	G12; G13; G14; G15; G16; G17; G18	D3	0.8
Pria	R4_P	G19; G20; G21; G22; G23	D4	0.8
Wanita	R1_W	G1; G2; G3; G4	D1	0.8
Wanita	R2_W	G2; G5; G6; G7; G8; G9; G10; G11	D2	0.9
Wanita	R3_W	G1; G11; G12; G13; G14; G15; G16; G17; G18	D3	0.8

Wanita	R4_W	G19; G20; G21; G22; G23	D4	0.8
Komplikasi	R8	D1; D3	K1	0.6
Komplikasi	R9	D2; D4	K2	0.7

a) Daftar Gejala (G)

- **G1:** Sebelum siklus menstruasi
- **G2:** Tidak ada nafsu makan
- **G3:** Cenderung menyendiri
- **G4:** Insomnia (susah tidur)
- **G5:** Merasa gelisah
- **G6:** Memutar atau meremas tangan
- **G7:** Berbicara tidak nyambung
- **G8:** Tidak dapat merasakan emosi kesedihan
- **G9:** Merasa sangat bersalah
- **G10:** Ingin bunah diri
- **G11:** Kehilangan siklus menstruasi
- **G12:** Perubahan kepribadian
- **G13:** Wajah terlihat murung
- **G14:** Pasif
- **G15:** Merasa curiga
- **G16:** Suka mengkritik
- **G17:** Penuh pikiran negatif
- **G18:** Ketakutan berlebihan akan menjadi gila

- **G19:** Mengalami delusi (khayalan)
- **G20:** Mengalami halusinasi
- **G21:** Merasa tidak aman
- **G22:** Merasa sedang diawasi
- **G23:** Merasa tidak berharga

b) Daftar Penyakit (D)

- **D1:** Depresi Vegetatif
- **D2:** Depresi Agitasi
- **D3:** Depresi Distimia
- **D4:** Depresi Psikotik

c) Daftar Komplikasi (K)

- **K1:** Gangguan Kecemasan
- **K2:** Gangguan Tidur Kronis

2. Representasi JSON (rules.json):

Struktur JSON digunakan untuk menyimpan keseluruhan basis pengetahuan agar dapat dibaca oleh inference engine.

```
{
  "gejala": [
    {"kode": "G1", "nama": "Sebelum siklus menstruasi", "cf_pakar": -0.8},
    {"kode": "G2", "nama": "Tidak ada nafsu makan", "cf_pakar": 0.8},
    ...
    {"kode": "G23", "nama": "Merasa tidak berharga", "cf_pakar": 0.8}
  ],
  "penyakit": [
```

```

    {"kode": "D1", "nama": "Depresi Vegetatif"},
    {"kode": "D2", "nama": "Depresi Agitasi"},
    {"kode": "D3", "nama": "Depresi Distimia"},
    {"kode": "D4", "nama": "Depresi Psikotik"}
  ],
  "komplikasi": [
    {"kode": "K1", "nama": "Gangguan Kecemasan"},
    {"kode": "K2", "nama": "Gangguan Tidur Kronis"}
  ],
  "aturan_pria": [
    {
      "id": "R1_P",
      "jika": ["G2", "G3", "G4"],
      "maka": "D1",
      "cf_rule": 0.8
    },
    ...
  ],
  "aturan_wanita": [
    {
      "id": "R1_W",
      "jika": ["G1", "G2", "G3", "G4"],
      "maka": "D1",

```

```

    "cf_rule": 0.8
  },
  ...
],
"aturan": [
  {
    "id": "R8",
    "jika": ["D1", "D3"],
    "maka": "K1",
    "cf_rule": 0.6
  },
  ...
]
}

```

E. Penjelasan Proses Inferensi (Forward Chaining + CF)

Proses inferensi diatur oleh engine.py berdasarkan metode yang dijelaskan dalam jurnal.

1. Proses Forward Chaining

Proses penalaran maju diimplementasikan dalam fungsi forward_chaining:

- a) Inisialisasi: Sistem mengumpulkan fakta awal, yaitu gejala yang dipilih pengguna (disimpan dalam self.facts) dan gender pasien.
- b) Seleksi Aturan: Berdasarkan gender, sistem memuat set aturan yang relevan (aturan_pria atau aturan_wanita). Aturan umum (untuk komplikasi) juga ditambahkan.
- c) Iterasi: Sistem melakukan iterasi (loop while changed) untuk mengevaluasi semua aturan.

- d) Pencocokan (Matching): Untuk setiap aturan, sistem memeriksa apakah semua kondisi premis (bagian jika) sudah terpenuhi dalam `self.facts` atau `self.conclusions`.
- e) Eksekusi (Firing): Jika semua premis terpenuhi, aturan dieksekusi. Sistem menghitung nilai CF untuk kesimpulan aturan tersebut (dijelaskan di bawah).
- f) Penambahan Fakta Baru: Jika CF yang dihitung ≥ 0 , kesimpulan (bagian maka, misal "D1") ditambahkan ke `self.conclusions`.
- g) Siklus Lanjutan: Fakta baru "D1" ini kemudian dapat digunakan untuk memicu aturan sekuensial (misal R8: JIKA D1...) pada iterasi berikutnya.
- h) Terminasi: Proses berhenti ketika tidak ada lagi aturan yang dapat dieksekusi (tidak ada fakta baru yang ditambahkan).

2. Proses Perhitungan Certainty Factor (CF) Perhitungan CF dalam implementasi ini didasarkan pada rumus-rumus dalam penelitian acuan

- a) CF Awal (Gejala): Ini adalah CF gabungan antara keyakinan pakar dan keyakinan pengguna. Dalam penelitian asli, ini disebut $CF(H,e)$.
 - $CF_{\text{gejala}} = CF_{\text{pakar}} \cdot CF_{\text{user}}$
 - CF_{pakar} diambil dari `rules.json` (misal $G2 = 0.8$).
 - CF_{user} diambil dari input UI (misal "Pasti Ada" = 0.8).
- b) CF Aturan (Kombinasi Premis AND): Jika sebuah aturan memiliki banyak premis (misal: IF $G2 \text{ AND } G3 \text{ AND } G4$), implementasi `engine.py` menentukan CF gabungan dari premis-premis tersebut dengan mengambil nilai minimum dari semua CF gejala.

$$CF_{\text{premis}} = \min(CF_{\text{gejala}_1}, CF_{\text{gejala}_2}, \dots)$$

Catatan: Ini berbeda dari metode studi kasus di paper yang menggabungkan CF setiap gejala secara sekuensial menggunakan rumus CF_{combine} . Metode min adalah interpretasi yang valid dan lebih sederhana untuk logika AND dalam CF).

- c) CF Kesimpulan (Rule Firing): CF akhir dari kesimpulan dihitung dengan mengalikan CF premis dengan CF aturan itu sendiri (cf_rule).

$$CF_calculated = CF_premis \cdot CF_rule$$

- d) Kombinasi Paralel: Jika dua aturan yang berbeda (misal R1 dan R5) sama-sama menghasilkan kesimpulan "D1", kedua CF tersebut harus digabungkan. Implementasi engine.py menggunakan fungsi combine_cf_parallel yang rumusnya identik dengan Persamaan (4) di penelitian asli :

- Jika CF1 dan CF2 positif:

$$CF_combine = CF1 + CF2 \cdot (1 - CF1)$$

- Jika CF1 dan CF2 beda tanda:

$$CF_combine = (CF1 + CF2) / (1 - \min(\text{abs}(CF1), \text{abs}(CF2)))$$

- Jika CF1 dan CF2 negatif:

$$CF_combine = CF1 + CF2 + (CF1 \cdot CF2)$$

F. Hasil Uji Coba

Simulasi pengujian dilakukan berdasarkan UI (depression_ui.py) dan inference engine (engine.py).

Skenario Uji 1: Depresi Vegetatif (Pria)

Input Pengguna:

Gender: Pria.

Gejala G2 (Tidak ada nafsu makan) -> Keyakinan: "Pasti Ada" (CF User: 0.8).

Gejala G3 (Cenderung menyendiri) -> Keyakinan: "Pasti Ada" (CF User: 0.8).

Gejala G4 (Insomnia) -> Keyakinan: "Pasti Ada" (CF User: 0.8).

Proses Inferensi (engine.py):

1. Fakta Awal: {G2: 0.8, G3: 0.8, G4: 0.8}. Gender: Pria.

Sistem Pakar Diagnosa Depresi

SISTEM PAKAR DIAGNOSA DEPRESI

Forward Chaining + Certainty Factor

TINGKAT KEYAKINAN GEJALA

☒ Pasti Ada (CF: 0.8)
 ☒ Mungkin Ada (CF: 0.4)
 ☐ Tidak Tahu (CF: 0.0)
 ☒ Mungkin Tidak (CF: -0.4)
 ☒ Pasti Tidak (CF: -0.8)

GEJALA YANG DIPILIH

G2: Tidak ada nafsu makan (CF: 0.8)
 G3: Cenderung menyendiri (CF: 0.8)
 G4: Insomnia (susah tidur) (CF: 0.8)

PILIH GEJALA YANG DIALAMI

☒ Tidak ada nafsu makan
 G2 (CF: 0.8)

☒ Cenderung menyendiri
 G3 (CF: 0.8)

☒ Insomnia (susah tidur)
 G4 (CF: 0.8)

TINGKAT KEYAKINAN:

☒ Pasti Ada
☒ Mungkin Ada
☐ Tidak Tahu
☒ Mungkin Tidak
☒ Pasti Tidak

TINGKAT KEYAKINAN:

☒ Pasti Ada
☒ Mungkin Ada
☐ Tidak Tahu

GENDER PASIEN: ☒ Pria ☐ Wanita

HASIL DIAGNOSA

Tampilan Output (UI):

HASIL DIAGNOSA

=====

DIAGNOSA UTAMA:

1. Depresi Vegetatif
 Kode: D1
 Tingkat Keyakinan: 0.999 (Sangat Tinggi)

KOMPLIKASI YANG MUNGKIN:

Tidak ada komplikasi yang teridentifikasi.

REKOMENDASI:

- Segera konsultasi dengan psikolog atau psikiater
- Pertimbangkan terapi medis dan psikologis
- Pantau kondisi secara berkala

Activate Windows
Go to Settings to activate Windows.

Skenario Uji 2: Depresi Psikotik (Wanita) dengan Gejala Negatif

Input Pengguna:

Gender: Wanita.

Gejala G19 (Delusi) -> Keyakinan: "Pasti Ada" (CF User: 0.8).

Gejala G20 (Halusinasi) -> Keyakinan: "Pasti Ada" (CF User: 0.8).

Gejala G21 (Tidak aman) -> Keyakinan: "Pasti Ada" (CF User: 0.8).

Gejala G22 (Diawasi) -> Keyakinan: "Pasti Ada" (CF User: 0.8).

Gejala G23 (Tidak berharga) -> Keyakinan: "Pasti Ada" (CF User: 0.8).

Gejala G7 (Bicara tidak nyambung) -> Keyakinan: "Tidak Ada" (CF User: -0.8).

Proses Inferensi (engine.py):

1. Fakta Awal: {G19: 0.8, G20: 0.8, G21: 0.8, G22: 0.8, G23: 0.8, G7: -0.8}.

Gender: Wanita.

SISTEM PAKAR DIAGNOSA DEPRESI
Forward Chaining + Certainty Factor

TINGKAT KEYAKINAN GEJALA

- ☒ Pasti Ada (CF: 0.8)
- ☒ Mungkin Ada (CF: 0.4)
- ☐ Tidak Tahu (CF: 0.0)
- ☒ Mungkin Tidak (CF: -0.4)
- ☒ Pasti Tidak (CF: -0.8)

PILIH GEJALA YANG DIALAMI

☐ Memutar atau meremas tangan (G6 (CF: 0.4))

Tingkat Keyakinan:

- ☒ Pasti Ada
- ☒ Mungkin Ada
- ☐ Tidak Tahu
- ☒ Mungkin Tidak
- ☒ Pasti Tidak

☒ Berbicara tidak nyambung (G7 (CF: -0.8))

Tingkat Keyakinan:

- ☒ Pasti Ada
- ☒ Mungkin Ada
- ☐ Tidak Tahu
- ☒ Mungkin Tidak
- ☒ Pasti Tidak

Tingkat Keyakinan:

- ☒ Pasti Ada

GEJALA YANG DIPILIH

G19: Mengalami delusi (halayalan) (CF: 0.8)
 G20: Mengalami halusinasi (CF: 0.8)
 G21: Merasa tidak aman (CF: 0.8)
 G22: Merasa sedang diawasi (CF: 0.8)
 G23: Merasa tidak berharga (CF: 0.8)
 G7: Berbicara tidak nyambung (CF: -0.8)

GENDER PASIEN: ☐ Pria ☒ Wanita

JALANKAN DIAGNOSA **RESET**

HASIL DIAGNOSA

Activate Windows
Go to Settings to activate Windows.

Tampilan Output (UI):

HASIL DIAGNOSA

=====

DIAGNOSA UTAMA:

1. Depresi Psikotik
 Kode: D4
 Tingkat Keyakinan: 0.948 (Sangat Tinggi)

KOMPLIKASI YANG MUNGKIN:

Tidak ada komplikasi yang teridentifikasi.

REKOMENDASI:

- Segera konsultasi dengan psikolog atau psikiater
- Pertimbangkan terapi medis dan psikologis
- Pantau kondisi secara berkala

Activate Windows
Go to Settings to activate Windows.

G. Kesimpulan

Implementasi ulang sistem pakar diagnosa depresi telah berhasil diselesaikan sesuai dengan arahan tugas. Sistem ini mengadopsi metodologi Forward Chaining dan Certainty Factor dari penelitian acuan.

Proses implementasi ulang ini melibatkan beberapa penyesuaian:

1. Platform: Migrasi dari sistem berbasis Web ke aplikasi desktop Python menggunakan Tkinter.
2. Struktur Data: Basis pengetahuan dialihkan ke format rules.json sesuai spesifikasi tugas, dengan tetap mempertahankan esensi aturan dari jurnal (termasuk pemisahan gender).
3. Modifikasi Aturan: Sistem berhasil diperkaya dengan aturan sekuensial (untuk diagnosis komplikasi) yang tidak ada dalam penelitian asli.
4. Logika CF: Inference engine (engine.py) mengimplementasikan rumus kombinasi paralel sesuai jurnal acuan dan menerapkan logika min untuk premis AND, yang merupakan metode valid untuk kalkulasi CF.

Secara keseluruhan, proyek ini berhasil mereplikasi, memahami, dan memodifikasi sistem pakar dari penelitian ilmiah, serta memenuhi semua deliverables yang disyaratkan dalam tugas.