

Dimas Kendika Fazrulfalah  
H1D023083  
Shift A

## Tugas 4

### Percobaan 1

1. Buat file silsilah.pl di Prolog dengan source code berikut

```
%SILSILAH KELUARGA
%FAKTA
%-- HUBUNGAN ORTU
% -- Predikat parent/2 (menerima dua parameter x dan y)
% -- berarti x adalah ortu dari y
parent(alya, bima).
parent(alya, satria).
parent(bima, david).
parent(bima, emma).
parent(satria, yunita).
parent(satria, grace).

% ATURAN
% -- Hubungan saudara kandung.
% -- Predikat sibling/2 (menerima dua parameter X dan Y)
% -- berarti X adalah saudara kandung Y apabila fakta-fakta sesuai dengan aturan.
sibling(X, Y) :-
    parent(Z, X),
    parent(Z, Y),
    X \= Y. % X tidak sama dengan Y (Garis miring adalah negasi)

% -- Hubungan kakek-nenek.
% -- Predikat grandparent/2 (menerima dua parameter X dan Y)
% -- berarti X adalah kakek/nenek dari Y apabila fakta-fakta sesuai dengan aturan.
grandparent(X, Y) :-
    parent(X, Z),
    parent(Z, Y).

% -- Hubungan turun-temurun.
% -- Predikat ancestor/2 (menerima dua parameter X dan Y)
% -- berarti X adalah pendahulu Y apabila fakta-fakta sesuai dengan aturan.
ancestor(X, Y) :-
    parent(X, Y).
```

```

ancestor(X, Y) :-
    parent(X, Z),
    ancestor(Z, Y).

```

2. Kemudian save buffer dan consult file tersebut di Prolog.
3. Masukkan query seperti berikut
 

```

parent(bima, Anak). // Siapa anaknya Bima?
grandparent(alya, Cucu). // Siapa cucunya Alya?
sibling(bima, satria). // Apakah Bima dan Satria saudara kandung?
ancestor(Leluhur, emma). // Siapa leluhurnya Emma?

```

#### Output

```

?- parent(bima, Anak).
Anak = david ,

?- grandparent(alya, Cucu).
Cucu = david ,

?- sibling(bima, satria).
true.

?- ancestor(Leluhur, emma).
Leluhur = bima

```

## Percobaan 2

1. Buat file malaria.pl di Prolog dengan source code berikut
 

```

% GEJALA PENYAKIT MALARIA
% DATABASE
% -- Menyimpan data apakah X memiliki gejala Y.
:- dynamic gejala/2.
% ATURAN
% -- Mengandung daftar gejala untuk beberapa jenis penyakit malaria,
% -- dengan predikat penyakit/2 (menerima dua parameter)
% -- yang berarti X mengidap penyakit Y.
% -- Predikat gejala/2 berarti X memiliki gejala Y.
penyakit(X, tertiana) :-
    gejala(X, nyeri_otot),
    gejala(X, muntah),
    gejala(X, kejang).
penyakit(X, quartana) :-
    gejala(X, nyeri_otot),
    gejala(X, menggigil),
    gejala(X, tidak_enak_badan).
penyakit(X, tropika) :-
    gejala(X, keringat_dingin),
    gejala(X, sakit_kepala),
    gejala(X, mimisan),
    gejala(X, mual).
penyakit(X, perniciososa) :-

```

```
gejala(X, menggigil),
gejala(X, tidak_enak_badan),
gejala(X, demam),
gejala(X, mimisan),
gejala(X, mual).
```

2. Save buffer dan consult file tersebut di Prolog
3. Masukkan query berikut  
penyakit(step, Penyakit). // Apa penyakit yang dimiliki Steph? assertz(gejala(step, nyeri\_otot)). // Menyimpan fakta X memiliki gejala Y assertz(gejala(step, menggigil)).  
assertz(gejala(step, tidak\_enak\_badan)).  
penyakit(step, Penyakit).  
retract(gejala(step, nyeri\_otot)). // Menghapus fakta X memiliki gejala Y  
assertz(gejala(step, demam)).  
assertz(gejala(step, mimisan)).  
assertz(gejala(step, mual)).  
penyakit(step, Penyakit).

## Output

```
?- penyakit(step, Penyakit).
false.
?- assertz(gejala(step, nyeri_otot)).
true.
?- assertz(gejala(step, menggigil)).
true.
?- assertz(gejala(step, tidak_enak_badan)).
true.
?- penyakit(step, Penyakit).
Penyakit = quartana
```

Pada awalnya steph tidak memiliki penyakit, namun karena dimasukan (assertz) gejala penyakit kepada steph, pada hasil terakhir steph memiliki penyakit berupa quartana.

```
?- retract(gejala(step, nyeri_otot)).
true.
?- assertz(gejala(step, demam)).
true.
?- assertz(gejala(step, mimisan)).
true.
?- assertz(gejala(step, mual)).
true.
?- penyakit(step, Penyakit).
Penyakit = perniciososa.
```

Kemudian ketika ada gejala yang dihilangkan (retract) dan ada gejala baru yang ditambahkan (assertz) maka Steph memiliki penyakit baru yakni Pernisiosa.

## Percobaan 3

1. Buat file pakar\_malaria.pl di Prolog dengan source code berikut  
% GEJALA PENYAKIT MALARIA

```

% DATABASE
% -- Menyimpan data apakah gejala X positif atau negatif.
:- dynamic gejala_pos/1.
:- dynamic gejala_neg/1.
% ATURAN
% -- Predikat pertanyaan/1 untuk menanyakan pertanyaan terkait gejala X.

pertanyaan(nyeri_otot) :-
    write("Apakah Anda merasa nyeri otot?").

pertanyaan(muntah) :-
    write("Apakah Anda muntah-muntah?").

pertanyaan(kejang) :-
    write("Apakah Anda mengalami kejang-kejang?").

pertanyaan(menggigil) :-
    write("Apakah Anda sering menggigil?").

pertanyaan(tidak_enak_badan) :-
    write("Apakah Anda merasa tidak enak badan?").

pertanyaan(keringat_dingin) :-
    write("Apakah Anda mengalami keringat dingin?").

pertanyaan(sakit_kepala) :-
    write("Apakah Anda sering sakit kepala?").

pertanyaan(mimisan) :-
    write("Apakah Anda sering mimisan?").

pertanyaan(mual) :-
    write("Apakah Anda merasa mual?").

pertanyaan(demam) :-
    write("Apakah Anda demam?").

% -- Predikat diagnosa/1 digunakan untuk menanyakan dan menyimpan status
% gejala X.

diagnosa(G) :-
pertanyaan(G),
    writeln(" (y/t)"),

```

```

    read(Jawaban),
Jawaban == y,
    assertz(gejala_pos(G)).
diagnosa(G) :-
    assertz(gejala_neg(G)),
fail.
% -- Predikat gejala/1 dipanggil untuk memeriksa status gejala dari
% database, -- atau menanyakan user terkait gejala tersebut.
gejala(G) :-
gejala_pos(G), !.
gejala(G) :-
gejala_neg(G), !,
fail.
gejala(G) :-
diagnosa(G).
% -- Mengandung daftar gejala untuk beberapa jenis penyakit malaria,
% -- dengan predikat penyakit/1 (menerima satu parameter)
% -- yang berarti mengidap penyakit X.
penyakit(tertiana) :-
gejala(nyeri_otot),
gejala(muntah),
gejala(kejang),
terdeteksi("Malaria Tertiana").
penyakit(quartana) :-
gejala(nyeri_otot),
gejala(menggigil),
gejala(tidak_enak_badan),
terdeteksi("Malaria Quartana").
penyakit(tropika) :-
gejala(keringat_dingin),
gejala(sakit_kepala),
gejala(mimisan),
gejala(mual),
terdeteksi("Malaria Tropika").
penyakit(pernisiosa) :-
gejala(menggigil),
gejala(tidak_enak_badan),
gejala(demam),
gejala(mimisan),
gejala(mual),
terdeteksi("Malaria Pernisiosa").
penyakit(_) :-
writeln("Tidak terdeteksi penyakit.").

```

```

% -- Predikat terdeteksi/1 dipanggil untuk mencetak penyakit.
terdeteksi(P) :-
write("Anda terdeteksi penyakit "),
writeln(P).
% -- Predikat clear_db/0 untuk membersihkan database gejala.
clear_db :-
retractall(gejala_pos(_)),
retractall(gejala_neg(_)).
% -- Predikat main/0 sebagai main loop sistem pakar.
main :-
write("\33\2J"), % Clear window
writeln("DIAGNOSA PENYAKIT THT"),
penyakit(_),
clear_db,
writeln("INGIN MENGULANG?"),
read(Jawaban), !,
Jawaban == y,
main.

```

2. Save buffer dan consult file tersebut di Prolog
3. Jalan query **main**.

#### Output

```

DIAGNOSA PENYAKIT THT
Apakah Anda merasa nyeri otot? (y/t)
|: y
|: .
Apakah Anda muntah-muntah? (y/t)
|: t.
Apakah Anda sering menggigil? (y/t)
|: t.
Apakah Anda mengalami keringat dingin? (y/t)
|: y.
Apakah Anda sering sakit kepala? (y/t)
|: y.
Apakah Anda sering mimisan? (y/t)
|: t.
Tidak terdeteksi penyakit.
DIAGNOSA PENYAKIT THT
Apakah Anda merasa nyeri otot? (y/t)
|: y.
Apakah Anda muntah-muntah? (y/t)
|: y.
Apakah Anda mengalami kejang-kejang? (y/t)
|: y.
Anda terdeteksi penyakit Malaria Tertiana

```

Nantinya akan muncul beberapa pertanyaan yang dapat dijawab dengan **y**. sebagai **YA** atau **t**. sebagai **TIDAK**. Dengan rules yang ada, akan memprediksi jawaban anda dan menampilkan penyakit yang diderita.

#### Percobaan 4

1. Buat file pakar\_malaria\_gui.pl di Prolog dengan source code berikut  
% GEJALA PENYAKIT MALARIA

% DATABASE

:- dynamic gejala\_pos/1.

:- dynamic gejala\_neg/1.

penyakit("Malaria Tertiana").

penyakit("Malaria Quartana").

penyakit("Malaria Tropika").

penyakit("Malaria Pernisiosa").

gejala(nyeri\_otot, "Malaria Tertiana").

gejala(muntah, "Malaria Tertiana").

gejala(kejang, "Malaria Tertiana").

gejala(nyeri\_otot, "Malaria Quartana").

gejala(menggigil, "Malaria Quartana").

gejala(tidak\_enak\_badan, "Malaria Quartana").

gejala(keringat\_dingin, "Malaria Tropika").

gejala(sakit\_kepala, "Malaria Tropika").

gejala(mimisan, "Malaria Tropika").

gejala(mual, "Malaria Tropika").

gejala(menggigil, "Malaria Pernisiosa").

gejala(tidak\_enak\_badan, "Malaria Pernisiosa").

gejala(demam, "Malaria Pernisiosa").

gejala(mimisan, "Malaria Pernisiosa").

gejala(mual, "Malaria Pernisiosa").

pertanyaan(nyeri\_otot, Y) :-

Y = "Apakah Anda merasa nyeri otot?".

pertanyaan(muntah, Y) :-

Y = "Apakah Anda muntah-muntah?".

pertanyaan(kejang, Y) :-

Y = "Apakah Anda mengalami kejang-kejang?".

pertanyaan(menggigil, Y) :-

Y = "Apakah Anda sering menggigil?".

pertanyaan(tidak\_enak\_badan, Y) :-

Y = "Apakah Anda merasa tidak enak badan?".

pertanyaan(keringat\_dingin, Y) :-

Y = "Apakah Anda mengalami keringat dingin?".

pertanyaan(sakit\_kepala, Y) :-

Y = "Apakah Anda sering sakit kepala?".

pertanyaan(mimisan, Y) :-

Y = "Apakah Anda sering mimisan?".

pertanyaan(mual, Y) :-

Y = "Apakah Anda merasa mual?".

pertanyaan(demam, Y) :-

Y = "Apakah Anda demam?".

1. Buka code editor dan buat file bernama pakar\_malaria\_gui.py
2. Install library python PySwip dengan perintah berikut

**pip install -U pyswip**

3. Kemudian isi file pakar\_malaria\_gui.py dengan source code berikut:

```
import tkinter as tk
from tkinter import ttk
from pyswip import Prolog
from tkinter import messagebox

prolog = Prolog()
prolog.consult('pakar_malaria_gui.pl')

penyakit = list()
gejala = dict()
index_penyakit = 0
index_gejala = 0
current_penyakit = ""
current_gejala = ""

def mulai_diagnosa():
    global penyakit, gejala, index_penyakit, index_gejala
    # Bersihkan database prolog
    prolog.retractall("gejala_pos(_)")
    prolog.retractall("gejala_neg(_)")
    start_btn.configure(state=tk.DISABLED)
    yes_btn.configure(state=tk.NORMAL)
    no_btn.configure(state=tk.NORMAL)
    # Mendapatkan daftar penyakit dan gejala
    penyakit = [p["X"].decode() for p in
list(prolog.query("penyakit(X)"))]
    for p in penyakit:
        gejala[p] = [g["X"] for g in list(prolog.query(f"gejala(X,\'{p}\'))")]
    index_penyakit = 0
    index_gejala = -1
    pertanyaan_selanjutnya()

def pertanyaan_selanjutnya(ganti_penyakit = False):
    global current_penyakit, current_gejala, index_penyakit, index_gejala
    # Atur index penyakit
    if ganti_penyakit:
        # Ganti ke penyakit selanjutnya
```



```

        index_penyakit += 1
        index_gejala = -1
# Apabila daftar penyakit sudah habis berarti tidak terdeteksi penyakit
if index_penyakit >= len(penyakit):
    hasil_diagnosa()
    return
current_penyakit = penyakit[index_penyakit]

# Atur index gejala
index_gejala += 1

# Apabila semua gejala dari penyakit habis, berarti terdeteksi penyakit tsb
if index_gejala >= len(gejala[current_penyakit]):
    hasil_diagnosa(current_penyakit)
    return
current_gejala = gejala[current_penyakit][index_gejala]

# Cek status gejala di database prolog
if list(prolog.query(f'gejala_pos({current_gejala})")):
    pertanyaan_selanjutnya()
    return
elif list(prolog.query(f'gejala_neg({current_gejala})")):
    pertanyaan_selanjutnya(ganti_penyakit=True)
    return

# Mendapatkan pertanyaan baru
pertanyaan =
list(prolog.query(f'pertanyaan({current_gejala},Y)"))[0]["Y"].decode()

# Set pertanyaan ke kotak pertanyaan
tampilkan_pertanyaan(pertanyaan)

def tampilkan_pertanyaan(pertanyaan):
    kotak_pertanyaan.configure(state=tk.NORMAL)
    kotak_pertanyaan.delete(1.0, tk.END)
    kotak_pertanyaan.insert(tk.END, pertanyaan)
    kotak_pertanyaan.configure(state=tk.DISABLED)

def jawaban(jwb):
    if jwb:
        prolog.assertz(f'gejala_pos({current_gejala})")
        pertanyaan_selanjutnya()
    else:

```

```

        prolog.assertz(f'gejala_neg({current_gejala})')
        pertanyaan_selanjutnya(ganti_penyakit=True)

def hasil_diagnosa(penyakit = ""):
    if penyakit:
        messagebox.showinfo("Hasil Diagnosa", f"Anda terdeteksi {penyakit}.")
    else:
        messagebox.showinfo("Hasil Diagnosa", "Tidak terdeteksi penyakit.")
    yes_btn.configure(state=tk.DISABLED)
    no_btn.configure(state=tk.DISABLED)
    start_btn.configure(state=tk.NORMAL)

# Inisialisasi window utama
root = tk.Tk()
root.title("Sistem Pakar Diagnosis Penyakit Malaria")

# Inisialisasi frame utama
mainframe = ttk.Frame(root, padding="3 3 12 12")
mainframe.grid(column=0, row=0, sticky=(tk.N, tk.W, tk.E, tk.S))
root.columnconfigure(0, weight=1)
root.rowconfigure(0, weight=1)

# Membuat widget yang diperlukan
ttk.Label(mainframe, text="Aplikasi Diagnosa Penyakit Malaria",
font=("Arial", 16)).grid(column=0, row=0, columnspan=3)

ttk.Label(mainframe, text="Kolom Pertanyaan:").grid(column=0, row=1)

kotak_pertanyaan = tk.Text(mainframe, height=4, width=40,
state=tk.DISABLED)

kotak_pertanyaan.grid(column=0, row=2, columnspan=3)

no_btn = ttk.Button(mainframe, text="Tidak", state=tk.DISABLED,
command=lambda: jawaban(False))

no_btn.grid(column=1, row=3, sticky=(tk.W, tk.E))

yes_btn = ttk.Button(mainframe, text="Ya", state=tk.DISABLED,
command=lambda: jawaban(True))

yes_btn.grid(column=2, row=3, sticky=(tk.W, tk.E))

```

```

start_btn = ttk.Button(mainframe, text="Mulai Diagnosa",
                        command=mulai_diagnosa)

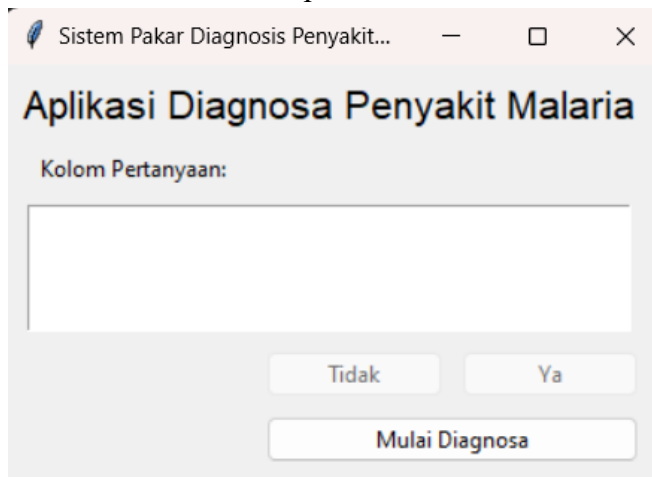
start_btn.grid(column=1, row=4, columnspan=2, sticky=(tk.W, tk.E))

# Tambah padding ke setiap widget
for widget in mainframe.winfo_children():
    widget.grid_configure(padx=5, pady=5)

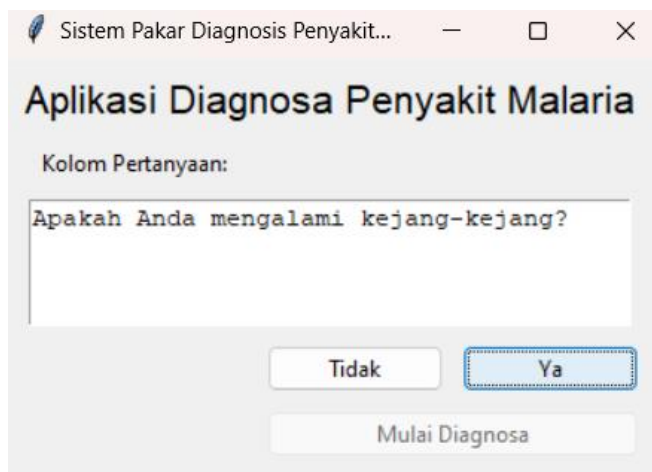
# Menjalankan GUI
root.mainloop()

```

4. Run code dengan perintah **python pakar\_malaria\_gui.py**
5. Kemudian akan muncul GUI seperti berikut



6. Kita bisa klik mulai diagnose untuk menampilkan pertanyaan dan bisa menjawab dengan klik Ya atau Tidak



7. Dengan pertanyaan yang kita jawab, akan mendiagnosa penyakit yang diderita

