

MODUL PRAKTIKUM KECERDASAN BUATAN

PERTEMUAN 8 – SISTEM PAKAR

Tools : Jupyter notebook

Bahasa Pemrograman : Python

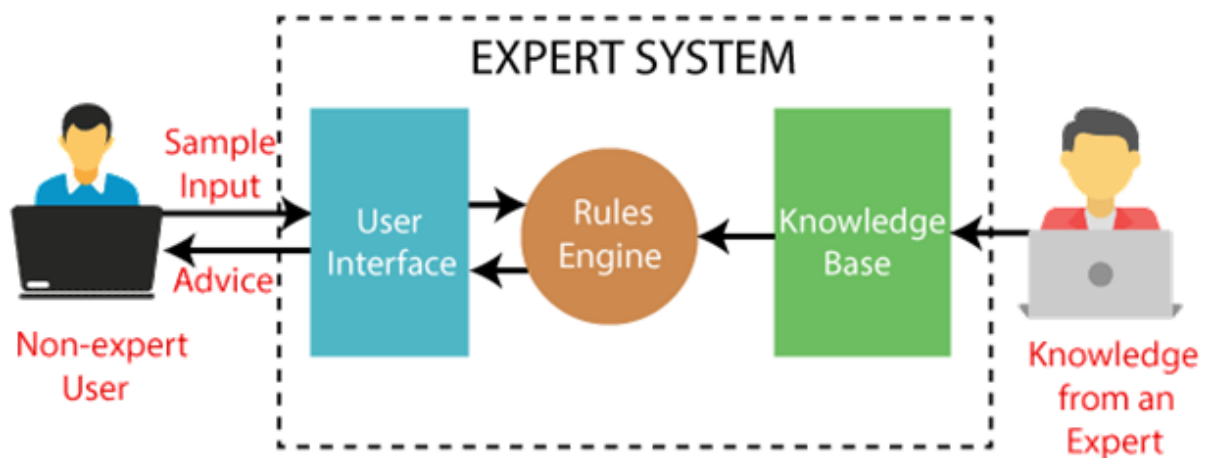
Dalam praktikum kali ini, kita akan belajar tentang bagaimana cara membuat Sistem Pakar menggunakan Python.

Sistem Pakar

Sistem pakar merupakan bagian dari kecerdasan buatan. Sistem pakar adalah program komputer yang dirancang untuk memecahkan masalah yang kompleks dan memberikan kemampuan pengambilan keputusan layaknya seorang pakar. Sistem pakar membantu dalam pengambilan keputusan menggunakan fakta secara heuristik seperti pakar. Disebut sistem pakar karena sistem tersebut berisi pengetahuan pakar dalam bidang tertentu (kedokteran, sains, dll) dan dapat memecahkan masalah kompleks apa pun dari bidang tersebut.

Kinerja sistem pakar didasarkan pada pengetahuan pakar yang disimpan dalam basis pengetahuannya (knowledge base/ KB). Data dalam basis pengetahuan ditambahkan oleh manusia yang ahli dalam domain tertentu ke dalam sistem untuk kemudian digunakan oleh pengguna non-ahli dalam memperoleh informasi. Semakin banyak pengetahuan yang disimpan dalam KB, semakin baik pula kinerja dari sistem tersebut. Sistem pakar yang baik menyelesaikan suatu masalah secara akurat, cepat dan mudah digunakan.

Adapun cara kerja dari sistem pakar dapat dilihat pada Gambar 8.1.



Gambar 8.1. Cara kerja sistem pakar (<https://www.javatpoint.com/expert-systems-in-artificial-intelligence>)

Komponen Sistem Pakar

Sistem pakar secara umum memiliki 3 (tiga) komponen utama, yaitu:

1. *User Interface*

Antarmuka pengguna (user interface) dapat memberikan kemudahan bagi pengguna dalam berinteraksi dengan sistem dan menemukan solusi.

2. *Inference Engine*

Mesin inferensi (*inference engine*) merupakan “otak” dari sistem pakar. Mesin inferensi menerapkan aturan inferensi ke basis pengetahuan untuk memperoleh kesimpulan atau menyimpulkan informasi baru dengan tujuan untuk membantu dalam memperoleh solusi.

Mesin inferensi membantu sistem pakar dalam mengekstrak pengetahuan dari basis pengetahuan. Terdapat dua jenis mesin inferensi, yaitu:

a. Deterministik

Kesimpulan yang diambil dari jenis mesin inferensi ini dianggap benar yang didasarkan pada fakta dan aturan.

b. Probabilistik

Kesimpulan yang diambil dari jenis mesin inferensi ini mengandung ketidakpastian yang didasarkan pada probabilitas.

Terdapat dua cara penarikan kesimpulan dalam rangka memperoleh solusi, yaitu:

a. Forward Chaining

Mesin inferensi melewati semua fakta, kondisi dan derivasi sebelum menyimpulkan hasil . Forward chaining bekerja dari keadaan awal dan mencapai tujuan (akhir keputusan). Sebagai contoh:

A

A -> B

B

He is running.

If he is running, he sweats.

He is sweating.

b. Backward Chaining

Mesin inferensi mengetahui keputusan atau tujuan akhir, sistem ini dimulai dari tujuan dan bekerja mundur untuk menentukan fakta apa yang harus ditegaskan agar tujuan dapat dicapai, yaitu bekerja dari tujuan (keputusan akhir) dan mencapai awal.

Sebagai contoh:

B

A -> B

A

He is sweating.

If he is running, he sweats.

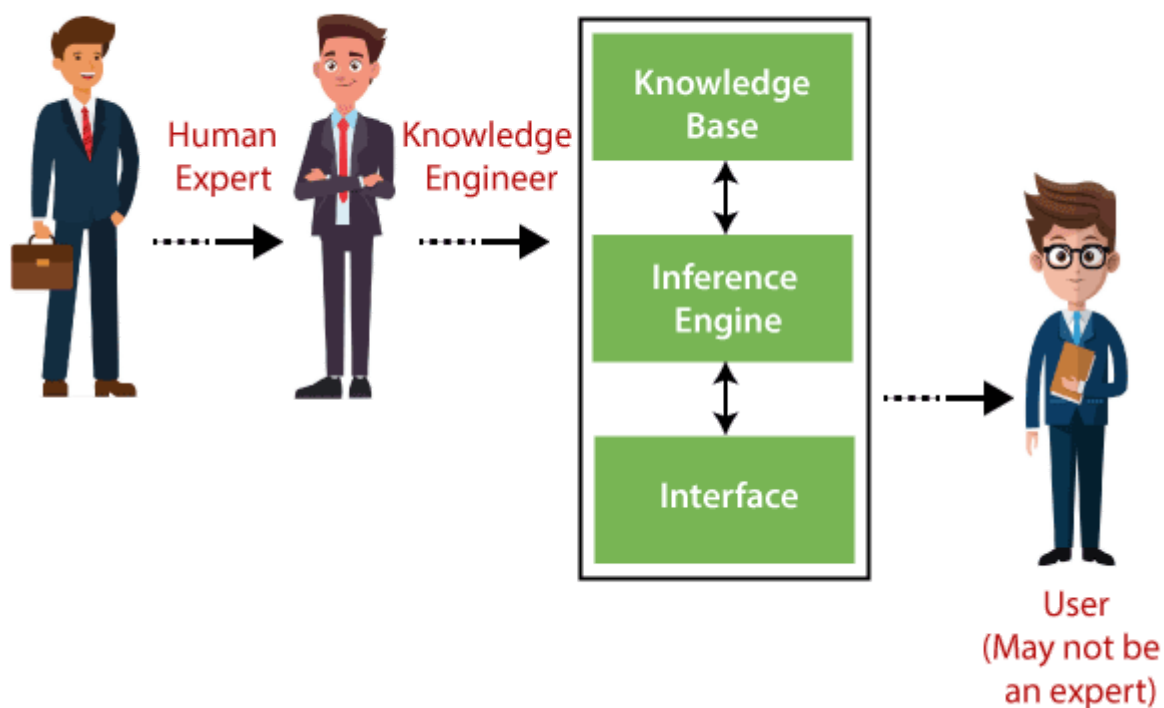
He is running.

3. Knowledge Base

Basis pengetahuan (*knowledge base*) menyimpan pengetahuan yang diperoleh dari pakar. Semakin banyak basis pengetahuan maka semakin baik pula Sistem Pakar. Basis pengetahuan serupa dengan database yang berisi informasi dan aturan dari domain atau subjek tertentu.

Seseorang juga dapat melihat basis pengetahuan sebagai kumpulan objek dan atributnya. Seperti singa adalah objek dan atributnya adalah mamalia, bukan hewan peliharaan, dll.

Hubungan antar komponen dalam sistem pakar dapat dilihat pada Gambar 8.2.



Gambar 8.2. Hubungan antar komponen dalam sistem pakar
(<https://www.javatpoint.com/expert-systems-in-artificial-intelligence>)

Pembuatan Sistem Pakar Sederhana

Pada sesi kali ini, kita akan membuat sistem pakar menggunakan Python. Pada sesi ini, kita akan membuat sistem pakar dengan mode forward chaining. Contoh kasus diambil dari paper berjudul “A Simple Expert System” yang ditulis oleh B.I. Blum. Sistem pakar yang akan dibuat merupakan

sistem pakar identifikasi hewan. Dimana dalam paper tersebut sudah didapatkan seperangkat aturan yang digunakan sebagai basis pengetahuan. Adapun basis pengetahuan yang akan digunakan dapat dilihat pada Tabel 8.1.

Tabel 8.1. Aturan sederhana dari basis pengetahuan

RULE 1		RULE 10	
On condition	[has hair]	On condition	[is carnivore]
Then	[is mammal]		[has tawny color]
			[has black stripes]
RULE 2		Then	[is tiger]
On condition	[gives milk]		GOAL
Then	[is mammal]		
RULE 3		RULE 11	
On condition	[has feathers]	On condition	[is ungulate]
Then	[is bird]		[has long legs]
			[has long neck]
RULE 4			[has tawny color]
On condition	[flies]	Then	[has dark spots]
	[lays eggs]		[is giraffe]
Then	[is bird]		GOAL
RULE 5		RULE 12	
On condition	[is mammal]	On condition	[is ungulate]
	[eats meat]		[has white color]
Then	[is carnivore]	Then	[has black stripes]
			[is zebra]
RULE 6			GOAL
On condition	[is mammal]		
	[has pointed teeth]	RULE 13	
	[has claws]	On condition	[is bird]
	[has foward pointing eyes]		[does not fly]
Then	[is carnivore]		[has long legs]
			[has long neck]
RULE 7		Then	[is black and white]
On condition	[is mammal]		[is ostrich]
	[has hoofs]		GOAL
Then	[is ungulate]		
RULE 8		RULE 14	
On condition	[is mammal]	On condition	[is bird]
	[chews cud]		[does not fly]
Then	[is ungulate]		[swims]
	[is even-toed]	Then	[is black and white]
			[is penguin]
			GOAL
RULE 9		RULE 15	
On condition	[is carnivore]	On condition	[is bird]
	[has tawny color]		[is good flier]
	[has dark spots]	Then	[is albatross]
Then	[is cheetah]		GOAL
	GOAL		

Pada praktikum kali ini, kita tidak sampai pada pembuatan *user interface*. Percobaan dilakukan sampai dengan pengecekan kebenaran dalam penarikan kesimpulan identifikasi hewan dengan metode *forward chaining*. Adapun rule yang digunakan adalah rule 1 sampai rule 10.

Percobaan 1

Forward Chaining - Expert System

Animal identification

```
In [18]: global facts
global is_changed

is_changed = True

## Existing facts and will be identified.
facts = [["gives milk", "x"], ["eats meat", "x"], ["tawny color", "x"]]

## the function of asserting facts
def assert_fact(fact):
    global facts
    global is_changed

    if not fact in facts:
        facts += [fact]
        is_changed = True

# knowledge base
while is_changed:
    is_changed = False
    for xf in facts:
        if xf[0] == "has hair":
            assert_fact(["is mammal", xf[1]])
        if xf[0] == "gives milk":
            assert_fact(["is mammal", xf[1]])
        if xf[0] == "has feathers":
            assert_fact(["is bird", xf[1]])
        if xf[0] == "flies" and ["lays egg", xf[1]] in facts:
            assert_fact(["is bird", xf[1]])
        if xf[0] == "is mammal" and ["eats meat", xf[1]] in facts:
            assert_fact(["is carnivore", xf[1]])
        # Use "|" as a line separator in multiline statements.
        if xf[0] == "is mammal" and ["has pointed teeth", xf[1]] and ["has claws", xf[1]] and \
            ["has forward pointing eyes", xf[1]] in facts:
            assert_fact(["is carnivore", xf[1]])
        if xf[0] == "is mammal" and ["has hoofs", xf[1]] in facts:
            assert_fact(["is ungulate", xf[1]])
            assert_fact(["is even-toed", xf[1]])
        if xf[0] == "is mammal" and ["has hoofs", xf[1]] in facts:
            assert_fact(["is ungulate", xf[1]])
        if xf[0] == "is mammal" and ["cews cud", xf[1]] and ["is ungulate", xf[1]] in facts:
            assert_fact(["is even-toed", xf[1]])
        if xf[0] == "is carnivore" and ["tawny color", xf[1]] and ["has dark spots", xf[1]] in facts:
            assert_fact(["is cheetah", xf[1]])
        if xf[0] == "is carnivore" and ["tawny color", xf[1]] and ["has black stripes", xf[1]] in facts:
            assert_fact(["is tiger", xf[1]])
    print(facts)
```

Pada percobaan kali ini, sistem akan mengidentifikasi hewan x termasuk dalam jenis apa. Dimana terdapat beberapa fakta terkait hewan x, yaitu: “gives milk”, “eats eat”, dan “tawny color”.

Output

```
[['gives milk', 'x'], ['eats meat', 'x'], ['tawny color', 'x'], ['is mammal', 'x'], ['is carnivore', 'x']]
```

Berdasarkan fakta-fakta yang ada dapat disimpulkan bahwa hewan x termasuk dalam karnivora.

Percobaan 2

Forward Chaining - Expert System

Animal identification

```
In [2]: global facts
global is_changed

is_changed = True

## Existing facts and will be identified.
facts = [["gives milk","y"], ["eats meat","y"], ["tawny color","y"], ["has dark spots","y"]]

## the function of asserting facts
def assert_fact(fact):
    global facts
    global is_changed

    if not fact in facts:
        facts += [fact]
        is_changed = True
```

```
# knowledge base
while is_changed:
    is_changed = False
    for xf in facts:
        if xf[0] == "has hair":
            assert_fact(["is mammal",xf[1]])
        if xf[0] == "gives milk":
            assert_fact(["is mammal",xf[1]])
        if xf[0] == "has feathers":
            assert_fact(["is bird",xf[1]])
        if xf[0] == "flies" and ["lays egg",xf[1]] in facts:
            assert_fact(["is bird",xf[1]])
        if xf[0] == "is mammal" and ["eats meat",xf[1]] in facts:
            assert_fact(["is carnivore",xf[1]])
        # Use "\n" as a line separator in multiline statements.
        if xf[0] == "is mammal" and ["has pointed teeth",xf[1]] and ["has claws",xf[1]] and \
        ["has forward pointing eyes",xf[1]] in facts:
            assert_fact(["is carnivore",xf[1]])
        if xf[0] == "is mammal" and ["has hoofs",xf[1]] in facts:
            assert_fact(["is ungulate",xf[1]])
            assert_fact(["is even-toed",xf[1]])
        if xf[0] == "is mammal" and ["has hoofs",xf[1]] in facts:
            assert_fact(["is ungulate",xf[1]])
        if xf[0] == "is mammal" and ["cews cud",xf[1]] and ["is ungulate",xf[1]] in facts:
            assert_fact(["is even-toed",xf[1]])
        if xf[0] == "is carnivore" and ["tawny color",xf[1]] and ["has dark spots",xf[1]] in facts:
            assert_fact(["is cheetah",xf[1]])
        if xf[0] == "is carnivore" and ["tawny color",xf[1]] and ["has black stripes",xf[1]] in facts:
            assert_fact(["is tiger",xf[1]])
    print(facts)
```

Pada percobaan kali ini, sistem akan mengidentifikasi hewan y termasuk dalam jenis apa. Dimana terdapat beberapa fakta terkait hewan y, yaitu: “gives milk”, “eats eat”, “tawny color”, dan “has dark spots”

Output

```
[['gives milk', 'y'], ['eats meat', 'y'], ['tawny color', 'y'], ['has dark spots', 'y'], ['is mamma
l', 'y'], ['is carnivore', 'y'], ['is cheetah', 'y']]
```

Berdasarkan fakta-fakta yang ada dapat disimpulkan bahwa hewan y merupakan cheetah.

Latihan: Menambahkan rule 11 sampai dengan rule 15 pada sistem pakar

1. Tambahkan rule 11 sampai dengan rule 15 untuk melengkapi sistem pakar di atas!
2. Gunakan input dari pengguna agar sistem pakar semakin interaktif!

3. Lakukan pengecekan dalam proses inferensi! (Apakah sudah benar atau masih terjadi kesalahan?)
4. Jelaskan kesimpulan yang Anda dapat dari percobaan kali ini!

Referensi:

Blum, B.I. "A Simple Expert System". The John Hopkins University - Applied Physics Laboratory.

<https://www.javatpoint.com/expert-systems-in-artificial-intelligence>

<https://expertisefinder.com/expert-systems/>

<https://www.geeksforgeeks.org/difference-between-backward-and-forward-chaining/>