

Praktikum Kecerdasan Buatan

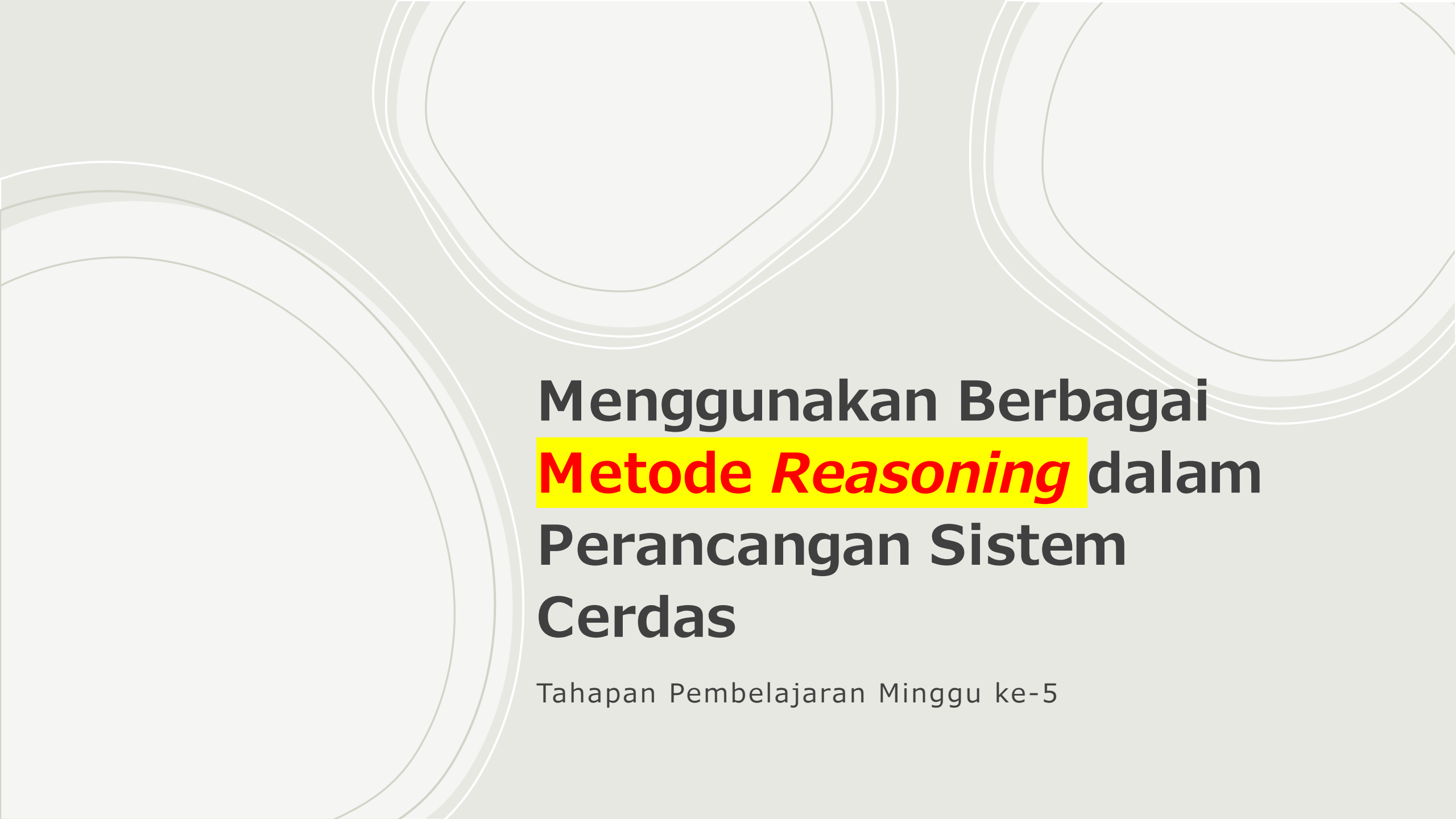
Fitri Nuraeni, M.Kom

PS Teknik Informatika (S1) - Jurusan Ilmu Komputer

Institut Teknologi Garut

2024





Menggunakan Berbagai **Metode *Reasoning*** dalam Perancangan Sistem Cerdas

Tahapan Pembelajaran Minggu ke-5

Capaian Pembelajaran

L5: Mahasiswa mampu menggunakan berbagai metode reasoning dalam perancangan sistem cerdas (C3, A2, P3)

[5.1] Ketepatan mendemonstrasikan metode penalaran runut maju

[5.2] Ketepatan mendemonstrasikan metode penalaran runut mundur



Metode Inferensi 1: ***Forward Chaining***

Bagian 1

latihanFC01.pl

Salin code prolog disamping
pada file **latihanFC01.pl**.

```
/* Fakta */
saudara(john, mary).
saudara(john, paul).
saudara(mary, peter).
saudara(peter, anne).
orang_tua(bob, john).
orang_tua(bob, mary).
orang_tua(bob, paul).
orang_tua(mary_senior, mary).
orang_tua(mary_senior, paul).

/* Aturan */
anak(X, Y) :- orang_tua(Y, X).
keponakan(X, Y) :- anak(X, Z), saudara(Z, Y).
cucu(X, Y) :- anak(X, Z), orang_tua(Z, Y).
paman(X, Y) :- saudara(X, Z), orang_tua(Z, Y).
```

latihanFC01.pl (lanjutan)

Salin code prolog disamping
pada file **latihanFC01.pl**.

```
/* Predikat untuk melakukan penalaran forward chaining */
penalaran :-
    write('Masukkan nama orang: '),
    read>Nama), tulis_hubungan>Nama), !.

/* Fungsi untuk menulis semua hubungan yang terkait dengan nama yang diberikan */
tulis_hubungan>Nama) :-
    anak>Nama, OrangTua),
    write>Nama), write(' adalah anak dari '), write(OrangTua), nl,
    fail. /* Gagal agar Prolog mencoba aturan lain jika ada */
tulis_hubungan>Nama) :-
    keponakan>Nama, Paman),
    write>Nama), write(' adalah keponakan dari '), write(Paman), nl,
    fail.
tulis_hubungan>Nama) :-
    cucu>Nama, Kakek),
    write>Nama), write(' adalah cucu dari '), write(Kakek), nl,
    fail.
tulis_hubungan>Nama) :-
    paman>Nama, Keponakan),
    write>Nama), write(' adalah paman dari '), write(Keponakan), nl,
    fail.
tulis_hubungan(_).

/* Jalankan penalaran */
:- penalaran.
```

latihanFC01.pl (lanjutan)

Panggil pengetahuan tersebut melalui perintah **File<Consult** pilih **latihanFC01.pl**

Jika sudah sukses terpanggil, ketikkan perintah: **penalaran.**

Coba masukan nama:

1. mary
2. john
3. paul

Screenshoot hasil inferensinya seperti gambar disamping

?- penalaran.

Masukkan nama orang: mary.

mary adalah anak dari bob

mary adalah anak dari mary_senior

mary adalah cucu dari john

mary adalah cucu dari mary

mary adalah cucu dari paul

mary adalah cucu dari mary

mary adalah cucu dari paul

true.

latihanFC02.pl

Salin code berikut:

```
:- dynamic user_fact/1.
:- dynamic inferred_fact/1.

% Predikat untuk menghapus semua fakta sebelum program dijalankan
clear_facts :-
    retractall(user_fact(_)),
    retractall(inferred_fact(_)).

% Predikat untuk meminta input fakta dari pengguna
ask_for_facts :-
    write('Masukkan fakta-fakta awal (ketik "sunny."/"dry."/"wet."/"rainy."/"selesai." untuk berhenti): '), nl,
    read(Fact),
    (
        Fact \= selesai ->
        (
            assert(user_fact(Fact)),
            ask for facts
        );
        true
    ).
```


latihanFC02.pl (lanjutan)

Salin code berikut:

```
% Rules
rule(play) :- user_fact(sunny), user_fact(dry).
rule(not_play) :- user_fact(sunny), user_fact(wet).
rule(not_play) :- user_fact(rainy).

% Inference engine
infer :-
    rule(X),
    not(inferred_fact(X)),
    assert(inferred_fact(X)),
    fail.
infer.
```

latihanFC02.pl (lanjutan)

Salin code berikut:

```
% Print inferred facts
print_facts :-
    write('Facts:'), nl,
    user_fact(X),
    write('- '), write(X), nl,
    fail.
print_facts :-
    write('Inferred Facts:'), nl,
    inferred_fact(X),
    write('- '), write(X), nl,
    fail.
print_facts.

% Mulai program
start :-
    clear_facts, % Membersihkan fakta sebelum program dijalankan
    ask_for_facts,
    infer,
    print_facts.
```

latihanFC02.pl (lanjutan)

Panggil pengetahuan tersebut melalui perintah **File<Consult** pilih **latihanFC02.pl**

Jika sudah sukses terpanggil, ketikan perintah: **start.**

Coba masukan fact:

1. sunny & dry
2. Sunny & wet
3. rainy

Screenshoot hasil inferensinya seperti gambar disamping

?- start.

Masukkan fakta-fakta awal (ketik "sunny."/"dry."/"wet."/"rainy."/"selesai." untuk berhenti):

|: sunny.

Masukkan fakta-fakta awal (ketik "sunny."/"dry."/"wet."/"rainy."/"selesai." untuk berhenti):

|: dry.

Masukkan fakta-fakta awal (ketik "sunny."/"dry."/"wet."/"rainy."/"selesai." untuk berhenti):

|: selesai.

Facts:

- sunny

- dry

Inferred Facts:

- play

true.



Metode Inferensi 2: ***Backward Chaining***

Bagian 2

latihanBC01.pl

Salin code berikut:

```
/* Fakta */
orangtua(ahmad, budi).
orangtua(ahmad, candra).
orangtua(budi, cindy).
orangtua(budi, david).
orangtua(candra, emily).
orangtua(candra, fahri).
orangtua(david, grace).

/* Aturan */
kakek(Kakek, Cucu) :-
    orangtua(Kakek, Anak),
    orangtua(Anak, Cucu).

saudara(Saudara1, Saudara2) :-
    orangtua(Orangtua, Saudara1),
    orangtua(Orangtua, Saudara2),
    Saudara1 \== Saudara2.

paman(Paman, Keponakan) :-
    saudara(Paman, Orangtua),
    orangtua(Orangtua, Keponakan).

/* Rules untuk inferensi */
cucu(Cucu, Kakek) :-
    kakek(Kakek, Cucu).
cucu(Cucu, Kakek) :-
    orangtua(Orangtua, Cucu),
    cucu(Orangtua, Kakek).
```

latihanBC01.pl (lanjutan)

Panggil pengetahuan tersebut melalui perintah **File<Consult** pilih **latihanBC01.pl**

Jika sudah sukses terpanggil, ketikan query:

1. `cucu(ahmad, cindy).`
2. `cucu(ahmad, Cucu).`
3. `paman(budy, Ponakan).`

Screenshoot hasil inferensinya.

latihanBC02.pl

Salin code berikut:

```
:- dynamic user_fact/1.
:- dynamic inferred_fact/1.

% Predikat untuk menghapus semua fakta sebelum program dijalankan
clear_facts :-
    retractall(user_fact(_)),
    retractall(inferred_fact(_)).

% Predikat untuk meminta input fakta dari pengguna
ask_for_facts :-
    write('Masukkan fakta-fakta awal (ketik "sunny."/"dry."/"wet."
/"rainy."/"selesai." untuk berhenti): '), nl,
    read(Fact),
    (
        Fact \= selesai ->
        (
            assert(user_fact(Fact)),
            ask for facts
        );
        true
    ).

% Rules
can_prove(play) :- user_fact(sunny), user_fact(dry). % Fakta play
dapat dibuktikan jika sunny dan dry terpenuhi
can_prove(not_play) :- user_fact(sunny), user_fact(wet). % Fakta n
ot_play dapat dibuktikan jika sunny dan wet terpenuhi
can_prove(not_play) :- user_fact(rainy). % Fakta not_play dapat di
buktikan jika rainy terpenuhi
```

latihanBC02.pl

Salin code berikut:

```
% Inference engine
infer(Fact) :-
    can_prove(Fact),
    assert(inferred_fact(Fact)), % Menyimpulkan fakta yang dapat d
ibuktikan
    !. % Memberhentikan pencarian setelah menemukan solusi

infer(_). % Inferensi gagal jika tidak ada aturan yang memungkinka
n untuk membuktikan fakta tersebut

% Print inferred facts
print_facts :-
    write('Facts:'), nl,
    user_fact(X),
    write('- '), write(X), nl,
    fail.
print_facts :-
    write('Inferred Facts:'), nl,
    inferred_fact(X),
    write('- '), write(X), nl,
    fail.
print_facts.

% Mulai program
start :-
    clear_facts, % Membersihkan fakta sebelum program dijalankan
    ask_for_facts,
    infer(play), % Inferensi backward chaining dimulai dengan fakt
a yang ingin kita buktikan
    infer(not_play), % Menggunakan inferensi backward chaining unt
uk menguji aturan lain
    print_facts.
```


latihanBC02.pl (lanjutan)

Panggil pengetahuan tersebut melalui perintah **File<Consult** pilih **latihanBC02.pl**

Jika sudah sukses terpanggil, ketikan perintah: **start.**

Coba masukan fact:

1. sunny & dry
2. Sunny & wet
3. rainy

Screenshoot hasil inferensinya seperti gambar disamping

?- start.

Masukkan fakta-fakta awal (ketik "sunny."/"dry."/"wet."/"rainy."/"selesai." untuk berhenti):

|: sunny.

Masukkan fakta-fakta awal (ketik "sunny."/"dry."/"wet."/"rainy."/"selesai." untuk berhenti):

|: wet.

Masukkan fakta-fakta awal (ketik "sunny."/"dry."/"wet."/"rainy."/"selesai." untuk berhenti):

|: selesai.

Facts:

- sunny
- wet

Inferred Facts:

- not_play

true.

Penilaian

Silakan minta 2 orang (intruktur dan/ atau asdos) untuk menilai hasil pekerjaan anda.

Dokumentasikan pekerjaan anda dalam bentuk PDF lalu submit ke LMS

- *Hasil pekerjaan disubmit oleh setiap mahasiswa dengan mencantumkan nama & nim seluruh anggota kelompok*
- *Tidak disubmit berarti nilai praktikum **DIBATALKAN**.*

L5: Mahasiswa Mampu Menggunakan Berbagai Metode *Reasoning* Dalam Perancangan Sistem Cerdas

Next