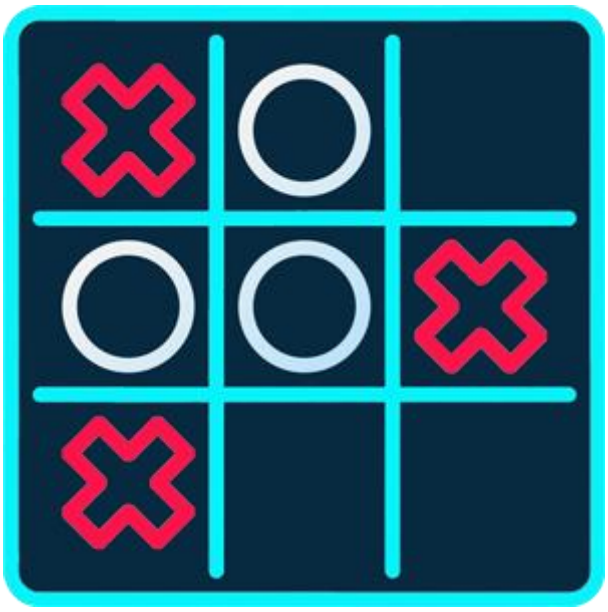
Modul 4

Game Sederhana menggunakan Prolog

Pendahuluan:



Tic Tac Toe merupakan permainan untuk dua orang pemain yang bertukaran memberikan tanda **X** atau **O** pada papan 3 x 3 di salah satu 9 kotak yang ada. Pemain dikatakan menang jika berhasil meletakkan tandanya (**X** atau **O**) dalam garis **horizontal, vertical** atau **diagonal berurutan**. Permainan menghasilkan kondisi seri jika seluruh kotak telah ditandai dan tidak ada tanda yang membentuk garis horizontal, vertical atau diagonal berurutan.

Selama memainkan Tic Tac Toe, fakta-fakta kondisi permainan akan terus perlu kita perbarui, hapus atau tambahkan ke dalam program. Untuk melakukan tersebut, kita dapat menggunakan fitur dalam Bahasa PROLOG, yaitu ***Facts Section*.**

Facts Section:

Facts Section merupakan section khusus pada PROLOG untuk mendeklarasikan *facts* dalam program yang merupakan bagian database yang dinamik (atau berubah terus menerus). Pada section ini, kita bisa melakukan perubahan selama masa *run time*.

Syntax Facts Section:

```
FACTS – namaFactSection              %namaFactSection opsional
      firstFact(typeData)
```

```
secondFact(typeData)
…
```

Tidak seperti section lainnya, kita dapat membuat beberapa fact section dengan memberikan namaFactSection.

Predicates yang bisa digunakan untuk mengolah Facts Section:

- `asserta(<the fact>, facts_sectionName)` `%memasukan facts di awal`
- `assertz(<the fact>, facts_sectionName)` `%memasukkan facts di akhir`
- `retract(<the fact>, facts_sectionName)` `%menghapus facts yg dipilih`
- `retractall(<the fact>,facts_sectionName)` `%menghapus semua facts`

fatcs_sectionName digunakan jika kita ada mendefinisikan nama fatcs session nya

Kita dapat menambahkan fact menggunakan asserta, dan assertz. Untuk melakukan penghapusan fact dapat dilakukan menggunakan retract dan retractall. Sedangkan untuk melakukan update dapat mengombinasikan predicates penghapusan dan predicates penambahan fact.

Untuk mengakses isi dari facts session sama dengan cara kita memanggil fact pada clauses. Contoh:

```
domains
        person = string
facts - daftarTeman
        teman(person, person)
predicates
        nondeterm go
        nondeterm init
        nondeterm cetakDaftarTeman
        nondeterm cariTeman(person)
        nondeterm filterTeman
clauses
        go:-
                init,
                cetakDaftarTeman,nl,
                cariTeman("Andi"), nl,
                filterTeman, nl.
        init:-
                assertz(teman("Andi", "Budi"), daftarTeman),
                assertz(teman("Andi", "Dudu"), daftarTeman),
                assertz(teman("Budi", "Dodo"), daftarTeman),
                assertz(teman("Dido", "Budi"), daftarTeman),
                assertz(teman("Andi", "Jojo"), daftarTeman).

        cetakDaftarTeman:-
                write("DAFTAR TEMAN"), nl,
                teman(Person, Person2),
                write(Person, " teman dari ", Person2), nl,
                1=2; %suatu statement yg false
                1=1. %suatu statement yang true

        cariTeman(Person):-
                write("DAFTAR TEMAN DARI ", Person), nl,
                teman(Person, Person2),
                write(Person2),nl,
                1=2; %suatu statement yg false
                1=1. %suatu statement yang true
```

```
cariTeman(Person):-
        write("DAFTAR TEMAN DARI ", Person), nl,
        teman(Person, Person2),
        write(Person2),nl,
        1=2; %suatu statement yg false
        1=1. %suatu statement yang true
filterTeman:-
        Filter = "Andi",
        write("DAFTAR TEMAN DARI ", Filter), nl,
        teman(Person, Person2),
        Person = Filter, %Kalau False maka berhenti disini
        write(Person2),nl,
        1=2; %suatu statement yg false
        1=1. %suatu statement yang true

goal
        go.
```

Perhatikan clauses cetakDaftarTeman, cariTeman, dan filterTeman. Cari perbedaan dan tarik kesimpulan cara kerjanya secara mandiri (Kita akan bahas dikelas).

Guided

Mari membuat game Tic Tac Toe yang terdiri dari dua player

```
domains
        kolom, baris = integer
        ox = char
facts - listKotak
        kotak(baris, kolom, ox)
predicates
        nondeterm opponent(ox, ox)
        nondeterm go
        nondeterm initKotak
        nondeterm tampilPapan(baris, kolom)
        nondeterm start(ox)
        nondeterm readMove(ox)
        nondeterm move(baris, kolom, ox)
        nondeterm validMove(baris, kolom)
        nondeterm anyValidMove
        nondeterm win(ox)
        nondeterm vertical(ox)
        nondeterm horizontal(ox)
        nondeterm diagonal(ox)
clauses
        %fakta
        opponent('O', 'X'). %Lawan dari O adalah X
        opponent('X', 'O'). %Lawan dari X adalah O
        %rule
        go:-
                initKotak,
                start('O'). %giliran pertama dimulai dari O

        initKotak:- %menginisialisasi posisi awal papan Tic Tac Toe
                assertz(kotak(1,1,' '), listKotak), assertz(kotak(1,2,' '), listKotak), assertz(kotak(1,3,' '), listKotak),
                assertz(kotak(2,1,' '), listKotak), assertz(kotak(2,2,' '), listKotak), assertz(kotak(2,3,' '), listKotak),
                assertz(kotak(3,1,' '), listKotak), assertz(kotak(3,2,' '), listKotak), assertz(kotak(3,3,' '), listKotak).


        tampilPapan(Baris,Kolom):-
                Baris <= 3,
                Kolom < 3,
                kotak(Baris, Kolom, OX),
                write(" ", OX ," ", "|"),
                Kolom1 = Kolom + 1,
                tampilPapan(Baris, Kolom1),!;

                Baris < 3,
                Kolom = 3,
                kotak(Baris, Kolom, OX),
                write(" ", OX ," "), nl,
                Baris1 = Baris + 1,
                tampilPapan(Baris1, 1),!;

                Baris = 3,
                Kolom = 3,
                kotak(Baris, Kolom, OX),
                write(" ", OX ," "), nl.

        start(OX):-
                opponent(OX, Player), %cek apakah lawan sudah menang
                win(Player), write(Player, " menang"), nl, tampilPapan(1,1),!;

                write("YOU: X"), nl,
                write("COM: O"), nl,
                tampilPapan(1,1),
                anyValidMove, %cek apakah ada gerakan yang memungkinkan, jika tidak berhenti disini dan seri
                readMove(OX),nl, %baca gerakan OX
                opponent(OX, Player), %cari lawan dari OX
                start(Player),!; %giliran lawan berikutnya

                write("Seri"), nl.
```

```
        anyValidMove:-
                kotak(_,_,' '). %selama ada yg kosong berarti masih bisa gerak

        readMove(OX):-
                write(OX, " move: "), nl,
                write("  Baris: "), readint(Baris),
                write("  Kolom: "), readint(Kolom),
                move(Baris, Kolom, OX).

        move(Baris, Kolom, OX):-
                validMove(Baris, Kolom), %cek apakah kotaknya uda berisi, jika uda invalid
                retract(kotak(Baris, Kolom, _), listKotak),
                assertz(kotak(Baris, Kolom, OX), listKotak);

                write("Inputan tidak valid"),nl,
                readMove(OX).
        validMove(Baris, Kolom):-
                kotak(Baris,Kolom, ' '), %Posisinya harus kosong
                Baris >= 1, Baris <= 3, %baris dan kolom diantara 1 dan 3
                Kolom >= 1, Kolom <= 3.
        win(OX):-
                horizontal(OX),!; vertical(OX),!; diagonal(OX).
        vertical(OX):-
                kotak(1, Kolom, OX), kotak(2, Kolom, OX), kotak(3, Kolom, OX).
        horizontal(OX):-
                kotak(Baris, 1, OX), kotak(Baris, 2, OX), kotak(Baris, 3, OX).
        diagonal(OX):-
                kotak(1, 1, OX), kotak(2, 2, OX), kotak(3, 3, OX),!;
                kotak(1, 3, OX), kotak(2, 2, OX), kotak(3, 1, OX).
goal
        go.
```

Format Pengumpulan: GD4_X_YYYY.zip

X = kelas

YYYYY = 5 digit NPM terakhir

Referensi

http://www.aistudy.co.kr/program/prolog/visual_prolog/Declaring%20the%20facts-sections.htm#1.%20Using%20the%20facts%20sections