TUGAS UTS ADVANCE AI

ZEBRA PUZZLE LOGIC PROGRAM

OLEH:

NAMA : TOBIAS MIKHA SULISTIYO

NIM : 2024002503

I. PENDAHULUAN

Perkembangan kemampuan manusia dalam berpikir menjadi landasan dalam pengambilan Keputusan. Fakta serta aturan yang dimiliki manusia membentuk kerangka logika yang digunakan untuk menentukan sikap dan tindakan manusia (Osiurak et al., 2018). Seiring dengan perkembangan teknologi, manusia berusaha merangkum kemampuan logika yang dimiliki ke dalam sistem komputasi dengan harapan supaya mesin dapat berpikir dalam kondisi yang sudah ditentukan.

Pada awalnya komputer diciptakan untuk menghitung maupun memberikan logika sederhana. Namun seiring perkembangan teknologi komputer logika yang dimiliki manusia ditanamkan ke dalam komputer. Perkembangan teknologi komputer semakin nyata melalui hukum Moore's (*Moore's Law*) yang menyatakan bahwa komputer meningkat dua kali lipat setiap tahunnya dengan biaya yang tetap (Shalf, 2020). Dengan adanya perkembangan teknologi tersebut menjadi peluang bahwa komputer bukan hanya sebagai alat hitung, namun sebagai tempat pengetahuan dan logika.

Gagasan logika komputer dalam berpikir mulai muncul pada era 1950 an melalui pendapat Alan Turing dan John McCarthy. Alan Turing berpendapat dalam *Turing Test,* apakah mesin dapat menirukan manusia sehingga jawabannya menyerupai manusia (Brynjolfsson, 2022). Pendapat Alan Turing yang menjadi cikal bakal perkembangan Ai di dunia. Istilah *Artificial Intelligence (Ai)* ditemukan dan digunakan oleh John McCarthy dalam penelitiannya tentang simulasi kecerdasan manusia (Haenlein & Kaplan, 2019).

Tentunya dalam perkembangan Ai tak luput dengan peran serta bahasa pemrograman. Bahasa pemrograman menjadi jembatan antara logika manusia dengan implementasi komputasi. Dalam konteks Ai, bahasa pemrograman seperti *Programming Logic (Prolog)* berperan penting dalam menyatakan hubungan logis, fakta dan aturan serta memanfaatkan mekanisme logika untuk menemukan solusi. Dalam laporan ini, prolog digunakan untuk menyelesaikan *"Zebra Puzzle"* karena kemampuan prolog dalam memodelkan dan mendeklarasikan logika.

II. MASALAH

Zebra puzzle merupakan teka-teki yang melibatkan banyak kondisi dan terhubung oleh berbagai atribut yang beragam . Karena melibatkan banyak kondisi dan setiap kondisi saling terhubung, maka diperlukan atribut yang tepat untuk menyelesaikan teka-teki ini (Berman et al., 2024). Setiap *zebra puzzle* memiliki atribut unik yang saling terhubung. Atribut-atribut ini bervariasi, mulai dari deskripsi hingga pengurutan. Berikut ini *clue* yang digunakan dalam *zebra puzzle* yang harus diselesaikan.

1. *The Englishman lives in the red house. The Spaniard owns the dog.*
2. *The Norwegian lives in the first house on the left.*
3. *The green house is immediately to the right of the ivory house.*
4. *The man who eats Hershey bars lives in the house next to the man with the fox.*
5. *Kit Kats are eaten in the yellow house.*
6. *The Norwegian lives next to the blue house. The Smarties eater owns snails.*
7. *The Snickers eater drinks orange juice. The Ukrainian drinks tea.*
8. *The Japanese eats Milky Ways.*
9. *Kit Kats are eaten in a house next to the house where the horse is kept.*
10. *Coffee is drunk in the green house.*
11. *Milk is drunk in the middle house.*

III. SOLUSI

Dalam perancangan solusinya, saya mengusulkan ada 2 metode. Yaitu permutasi dan *library* yang dimiliki oleh prolog.

Logic code untuk solusi 1.

1. Membuat *function* untuk membantu dalam logika tetangga/bersebelahan
   ➔ tetangga(A,B,List): A kemudian B atau B kemudian A dalam List
2. Membuat logika bahwa selalu di kanan
   ➔ kanan(A,B,List): A pasti di sebelah B dalam List
3. Logika untuk beberapa bantuan

➔ Solusi: terdiri dari 5 rumah dengan warna berbeda. Setiap rumah terdapat data makanan, minuman, hewan, dan kewarganegaraan yang berbeda.

4. Fakta berdasarkan *clue* yang diberikan
   ➔ Norwegia pemilik rumah pertama.
   ➔ Rumah ketiga meminum susu.
   ➔ Rumah kedua berwarna biru.
   ➔ Rumah hijau selalu bersebelahan dengan rumah *ivory*.
   ➔ Rumah hijau selalu meminum kopi
5. Metode *nested looping* untuk menemukan pasangan yang sesuai berdasarkan fakta yang sudah pasti dan fakta tambahan yang belum dimunculkan.
   ➔ Setiap fakta mutlak yang dideklarasikan sebelumnya kemudian disimpan dalam *array*, dari sisa fakta yang belum dimasukkan kemudian diuji berdasarkan fakta mutlak yang ada.

Karena saya kurang begitu mengerti untuk *logic flow* cara kedua, saya menggunakan bantuan Ai untuk *create code*-nya. Untuk *flow* langkah kedua yaitu menggunakan *library* `clpfd`. *Library* tersebut sudah menyediakan fungsi untuk data yang saling berpasangan dan bersebelahan sehingga tidak perlu metode pengulangan untuk saling membandingkan berdasarkan fakta yang diberikan.

IV. IMPLEMENTASI

Solusi 1. Menggunakan Permutasi.

```
tetangga(A, B, List) :- nextto(A, B, List); nextto(B, A, List).
kanan(A, B, List) :- append( _,[B,A|_], List).

solution(Houses) :-
    Houses = [
        [N1,C1,D1,F1,P1],
        [N2,C2,D2,F2,P2],
        [N3,C3,D3,F3,P3],
        [N4,C4,D4,F4,P4],
        [N5,C5,D5,F5,P5]
```

```
        ],

        N1 = norwegian,
        C2 = blue,
        D3 = milk,
        kanan([_, green, _, _, _], [_, ivory, _, _, _], Houses),
        member([_, green, coffee, _, _], Houses),

        permutation([englishman,    spaniard,    norwegian,    ukrainian,
    japanese],
                    [N1,N2,N3,N4,N5]),
        permutation([red, green, ivory, blue, yellow],
                    [C1,C2,C3,C4,C5]),
        permutation([coffee, milk, orange_juice, tea, water],
                    [D1,D2,D3,D4,D5]),
        permutation([hershey, kitkat, smarties, snickers, milky_ways],
                    [F1,F2,F3,F4,F5]),
        permutation([dog, snails, fox, horse, zebra],
                    [P1,P2,P3,P4,P5]),

        member([englishman, red, _, _, _], Houses),
        member([spaniard, _, _, _, dog], Houses),
        tetangga([_, _, _, hershey, _], [_, _, _, _, fox], Houses),
        member([_, yellow, _, kitkat, _], Houses),
        member([_, _, _, smarties, snails], Houses),
        member([_, _, orange_juice, snickers, _], Houses),
        member([ukrainian, _, tea, _, _], Houses),
        member([japanese, _, _, milky_ways, _], Houses),
        tetangga([_, _, _, kitkat, _], [_, _, _, _, horse], Houses).
```

Solusi 2. Menggunakan *Library* Prolog:

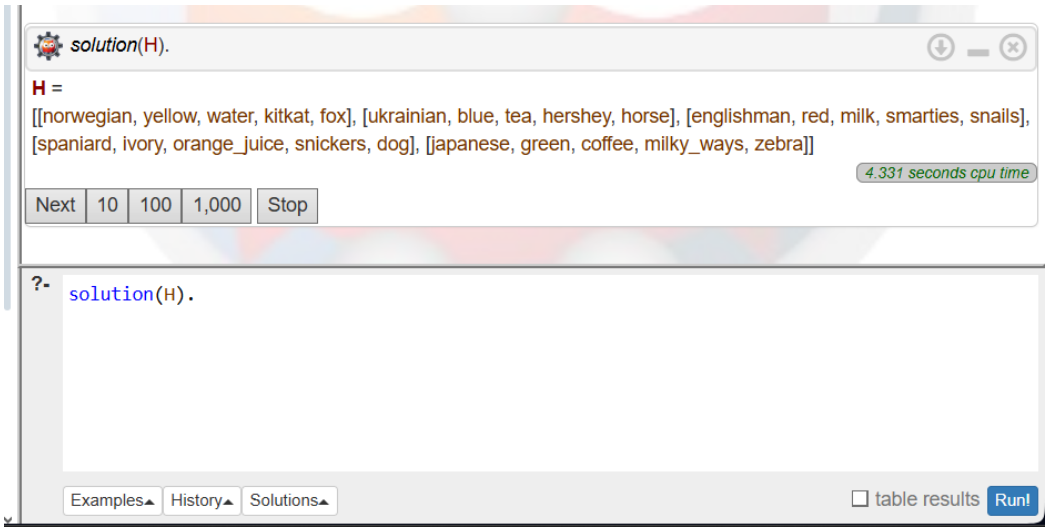
```
    :- use_module(library(clpfd)).

    zebra_clpfd(Houses) :-
        % Setiap variabel = posisi rumah (1..5)
        Vars = [
```

```
  Eng, Spa, Nor, Ukr, Jap,                    % nationality
  Red, Green, Ivory, Blue, Yellow,            % color
  Coffee, Milk, OJ, Tea, Water,               % drink
  Hershey, KitKat, Smarties, Snickers, Milky, % food
  Dog, Snails, Fox, Horse, Zebra              % pet
],
Vars ins 1..5,

% Unik per kategori
all_different([Eng,Spa,Nor,Ukr,Jap]),
all_different([Red,Green,Ivory,Blue,Yellow]),
all_different([Coffee,Milk,OJ,Tea,Water]),
all_different([Hershey,KitKat,Smarties,Snickers,Milky]),
all_different([Dog,Snails,Fox,Horse,Zebra]),

% Aturan puzzle:
Eng #= Red,                 % Englishman di rumah merah
Spa #= Dog,                 % Spaniard punya dog
Nor #= 1,                   % Norwegian di rumah 1 (paling kiri)
Green #= Ivory + 1,         % Green tepat di kanan Ivory
abs(Hershey - Fox) #= 1,    % Hershey bertetangga dengan Fox
KitKat #= Yellow,           % KitKat dimakan di rumah Yellow
abs(Nor - Blue) #= 1,       % Norwegian bertetangga dengan Blue
Smarties #= Snails,         % Smarties -> Snails
Snickers #= OJ,             % Snickers -> Orange Juice
Ukr #= Tea,                 % Ukrainian -> Tea
Jap #= Milky,               % Japanese -> Milky Ways
abs(KitKat - Horse) #= 1,   % KitKat bertetangga dengan Horse
Coffee #= Green,            % Coffee diminum di Green
Milk #= 3,                  % Milk di rumah tengah (3)

% Labeling
labeling([ffc, bisect, up],
         [Green,Ivory,Red,Blue,Yellow,
          Coffee,Milk,OJ,Tea,Water,
          Eng,Spa,Nor,Ukr,Jap,
          Hershey,KitKat,Smarties,Snickers,Milky,
          Dog,Snails,Fox,Horse,Zebra]),
```

```
    % Susun output: list 5 rumah, tiap rumah = [Nationality, Color,
Drink, Food, Pet]
    pairs_keys_values(NatPairs,
      [englishman,spaniard,norwegian,ukrainian,japanese],
      [Eng,Spa,Nor,Ukr,Jap]),
    pairs_keys_values(ColPairs,
      [red,green,ivory,blue,yellow],
      [Red,Green,Ivory,Blue,Yellow]),
    pairs_keys_values(DrinkPairs,
      [coffee,milk,orange_juice,tea,water],
      [Coffee,Milk,OJ,Tea,Water]),
    pairs_keys_values(FoodPairs,
      [hershey,kitkat,smarties,snickers,milky_ways],
      [Hershey,KitKat,Smarties,Snickers,Milky]),
    pairs_keys_values(PetPairs,
      [dog,snails,fox,horse,zebra],
      [Dog,Snails,Fox,Horse,Zebra]),

    findall([N,C,D,F,P],
      ( between(1,5,Pos),
        member(N-Pos, NatPairs),
        member(C-Pos, ColPairs),
        member(D-Pos, DrinkPairs),
        member(F-Pos, FoodPairs),
        member(P-Pos, PetPairs)
      ),
      Houses).
```

V. HASIL

Dalam teka teki zebra yang diberikan, zebra berada di rumah orang Jepang. Orang yang meminum air adalah orang Norwegia. Jawaban tersebut diambil dari beberapa sumber yang menyatakan jawaban tersebut. Dari hasil jawaban tersebut dicoba diselesaikan secara komputasi melalui dua solusi yaitu metode permutasi dan menggunakan *library.*



Gambar 1. Solusi menggunakan metode permutasi

Pada Gambar 1 metode permutasi dapat memperoleh jawaban yang diinginkan. Dengan menggunakan metode permutasi, program dapat menemukan hasil yang diharapkan dalam 4.3 detik. Hal ini cenderung lama, dan beberapa kali apabila jaringan kurang stabil memunculkan eror *time* limit karena waktu *looping* yang cenderung lama. Oleh karena itu, Saya mencoba mencari solusi lain melalui bantuan Ai.

Gambar 2. Solusi menggunakan library Prolog

Solusi kedua yaitu dengan menggunakan *library* yang sudah disediakan oleh Prolog. Prinsip kerja pada solusi kedua yaitu dengan mengubah seluruh *clue* menjadi logika aritmetika yang sudah disediakan *library* tersebut. Solusi kedua dapat menampilkan jawaban yang sesuai dengan waktu eksekusi kurang dari 1 detik seperti yang

diperlihatkan pada Gambar 2. Solusi kedua cenderung lebih cepat dikarenakan *library* yang digunakan memang sudah dirancang untuk menyelesaikan teka teki yang ada.

VI. KESIMPULAN

Kedua Solusi baik menggunakan permutasi / *looping* secara manual maupun menggunakan *library* yang sudah disediakan dapat menampilkan jawaban yang tepat. Penggunaan *library* memang dapat membantu dalam pemecahan masalah yang ada. Namun tentunya logika program yang dirancang juga harus menyesuaikan dengan spesifikasi maupun kemampuan *library* yang digunakan.

Meskipun cara *looping*/permutasi dapat digunakan, namun tentunya *resource* yang digunakan lebih besar dibandingkan menggunakan *library*. Untuk saran kedepannya apabila memang seorang developer dapat menggunakan *library* kusus untuk membantu dalam *nested loop* yang ada, lebih baik memanfaatkan *library* yang ada untuk meminimalisir penggunaan *resource* komputer. Namun apabila developer kurang bisa begitu menguasai *library* yang ada, dapat menggunakan *nested loop* dengan handle beberapa kondisi supaya tidak begitu memakan banyak *resource*.

REFERENSI

Berman, S., McKeown, K., & Ray, B. (2024). *Solving Zebra Puzzles Using Constraint-Guided Multi-Agent Systems*. http://arxiv.org/abs/2407.03956

Brynjolfsson, E. (2022). *The Turing Trap: The Promise & Peril of Human-Like Artificial Intelligence*. https://doi.org/https://doi.org/10.48550/arXiv.2201.04200

Haenlein, M., & Kaplan, A. (2019). A brief history of artificial intelligence: On the past, present, and future of artificial intelligence. *California Management Review*, *61*(4), 5–14. https://doi.org/10.1177/0008125619864925

Osiurak, F., Navarro, J., & Reynaud, E. (2018). How our cognition shapes and is shaped by technology: A common framework for understanding human tool-use

interactions in the Past, Present, and Future. In *Frontiers in Psychology* (Vol. 9, Issue MAR). Frontiers Media S.A. https://doi.org/10.3389/fpsyg.2018.00293

Shalf, J. (2020). The future of computing beyond Moore's Law. In *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* (Vol. 378, Issue 2166). Royal Society Publishing. https://doi.org/10.1098/rsta.2019.0061