

TUGAS INDIVIDU MATA KULIAH KECERDASAN BUATAN

Kifni Taufik Darmawan

15/388485/PPA/04924

Tugas :

Cari permasalahan yg bisa diselesaikan dengan searching solution, selain problem traveling. Buatlah problem space dan state spacenya. Jelaskan bagaimana anda menyimpan knowledge !

Problem Solving pada permainan Teka – Teki "Sliding Block Puzzle"

Permainan sliding block puzzle pertama kali diciptakan oleh Noyes Chapman. Versi pertama yang diciptakanya adalah versi 15 Puzzle. Permainan ini mulai populer pada tahun 1880. Puzzle ini terdiri dari 15 ubin yang diberi angka 1 sampai 14 pada 4x4 frame. Tujuan permainan ini adalah untuk mengurutkan ubin dari angka 1 sampai 15.

Permainan ini dikembangkan menjadi beberapa versi, 8-Puzzle adalah versi sederhana yang menggunakan 8 buah ubin dalam 3x3 frame. Tugas ini akan membahas proses problem solving permainan 8-puzzle menggunakan searching solution.

1) Problem Space & Space State

State Space

State space dalam kasus ini adalah gambaran konfigurasi dari 8 ubin dalam papan

- *Initial state*

Adalah kondisi awal konfigurasi 8 ubin dalam papan sebelum proses penyelesaian dimulai. Berikut adalah konfigurasi yang digunakan dalam proses problem solving dalam tugas ini :

2	8	3
1	6	4
7		5

Gambar 1. Initial State

- *Goal state*

Adalah konfigurasi ubin dalam papan yang diinginkan sebagai target akhir permainan. Berikut adalah goal state yang dalam proses problem solving yang diinginkan dalam tugas ini :

1	2	3
8		4
7	6	5

Gambar 2. Goal State

- *Action*

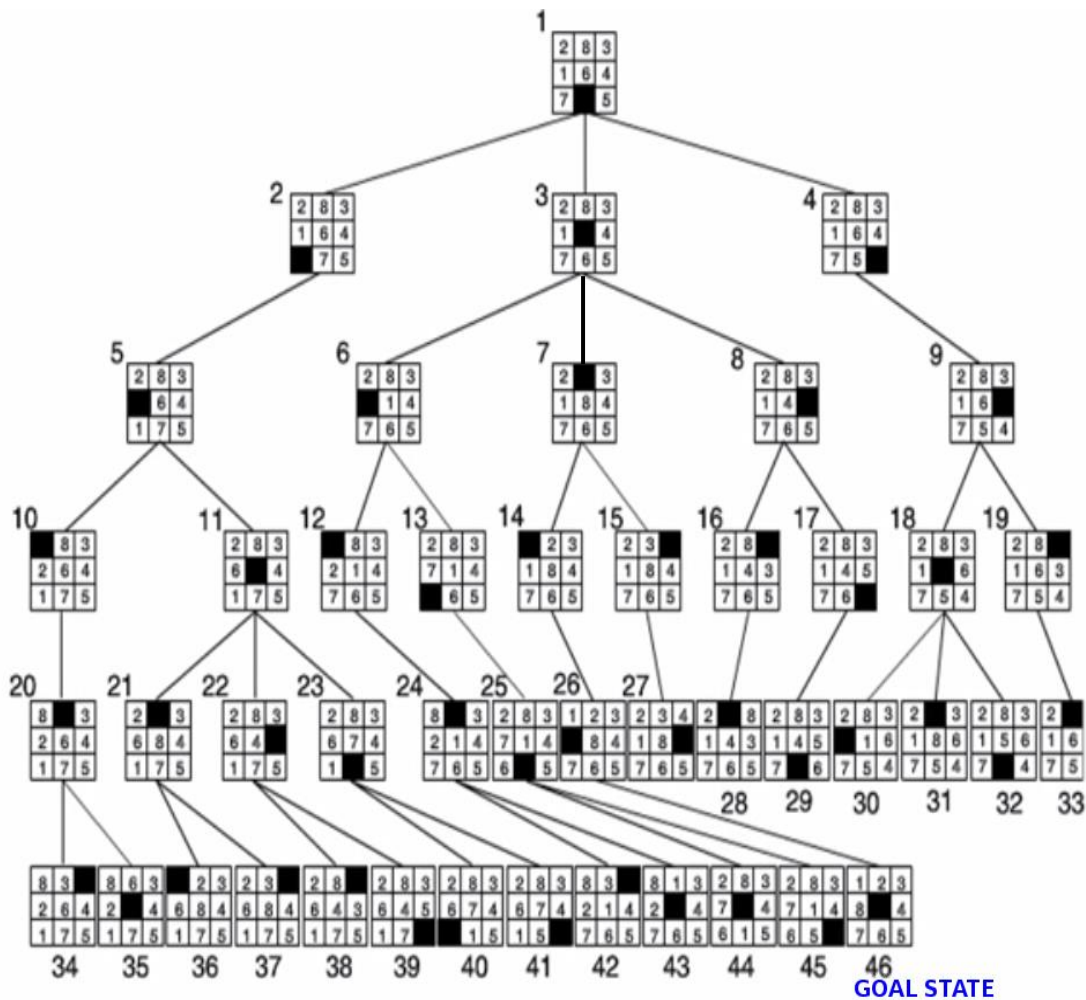
Action dalam permainan ini adalah perpindahan posisi ubin ke area yang kosong (dalam gambar ditunjukkan dengan blok hitam), baik ke kanan, kiri, bawah maupun atas.

- *Problem space*

Problem space direpresentasikan dengan directed graph, dimana setiap *node* merepresentasikan *state* dan *path (jalur)* merepresentasikan operasi yang dilakukan untuk mengubah state.

Untuk memudahkan implementasi dalam algoritma pencarian, problem space sering direpresentasikan dalam bentuk tree.

Berikut ini adalah representasi dari proses problem solving untuk menemukan goal state.



Gambar 3. Problem Space

2) Bagaimana menyimpan knowledge

- *State*

State kasus ini dalam dipresentasikan dalam multi-dimension array. Berikut adalah contoh representasinya dalam bahasa pemrograman PHP :

```
$goal = array(
    array(1,2,3),
    array(8,0,4),
    array(7,6,5)
);
```

- *Problem space*

Problem space dalam kasus ini direpresentasikan dalam bentuk tree. Berikut adalah representasi knowledge. Dalam bahasa pemrograman PHP, bentuk tree ini menggunakan *multidimension associative array*, array berikut berisi kumpulan array yang menunjukkan node pada tree, setiap node mempunya atribut 'state' yang menunjukkan konfigurasi posisi ubin dalam papan permainan, 'parent' menunjukan index array dari parent, dan 'childs' yang merupakan array yang berisi index dari anak yang dimilikinya, suatu node dapat memiliki lebih dari satu anak.

Berikut adalah contoh kode untuk merepresentasikan problem space dalam kasus ini :

```
$p_space = array(
    1 => (
        'state' => array(
            array(2,8,3),
            array(1,6,4),
            array(7,0,5)
        ),
        'parent' = null,
        'childs' = array(2,3,4)
    ),
    2=>(
        'state' => array(
            array(2,8,3),
            array(1,6,4),
            array(0,7,5)
        ),
        'parent' = 1,
        'childs' = array(5)
    ),
    2=>(
        'state' => array(
            array(2,8,3),
            array(1,6,4),
            array(0,7,5)
        ),
        'parent' = 1,
        'childs' = array(5)
    ),
    3=>(
        'state' => array(
            array(2,8,3),
            array(1,0,4),
            array(7,6,5)
        ),
        'parent' = 1,
        'childs' = array(6,7,8)
    ),
);
```

```

//dan seterusnya
.....
.....
//sampai ditemukan goal state yang diinginkan
46=>(
    'state' => array(
        array(1,2,3),
        array(8,0,4),
        array(7,6,5)
    ),
    'parent' = 26,
    'childs' = array()
)
)

```

Referensi :

1. *Improving sliding-block puzzle solving*. **Spaans, Ruben**. Trondheim : Norwegian University of Science and Technology, 2010.
2. PHP Arrays. *PHP Manual*. [Online] The PHP Group. [Cited: 09 17, 2015.]
<http://php.net/manual/en/language.types.array.php>.
3. **Spaans, Ruben**. Solving Sliding-block puzzles. *www.pvv.ntnu.no*. [Online] 2009. [Cited: 09 17, 2015.]
<http://www.pvv.ntnu.no/~spaans/spec-cs.pdf>.