

# Case Study : Search Algorithm

INF-303 Kecerdasan Buatan  
Jurusan Informatika – FMIPA UNSYIAH

**Irvanizam Zamanhuri, M.Sc**

**Zulfan, S.Si, M.Sc**

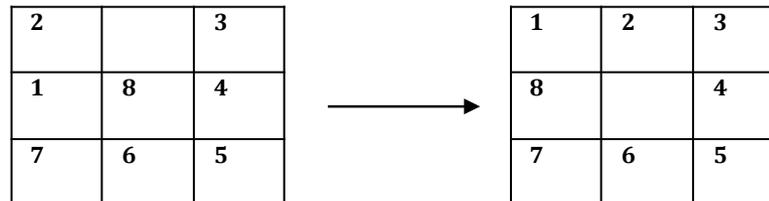
**Dalila Husna Yunardi, M.Sc**

# Contoh 1 : 8-puzzle dengan steepest ascent hill climbing

□□ Pada permainan 8-puzzle, keadaan awal dan tujuan adalah sebagai berikut:

**KEADAAN AWAL**

**TUJUAN**



□□ Operator yang digunakan untuk menggerakkan dari satu keadaan ke keadaan berikutnya adalah:

□□ Ubin kosong ke kanan

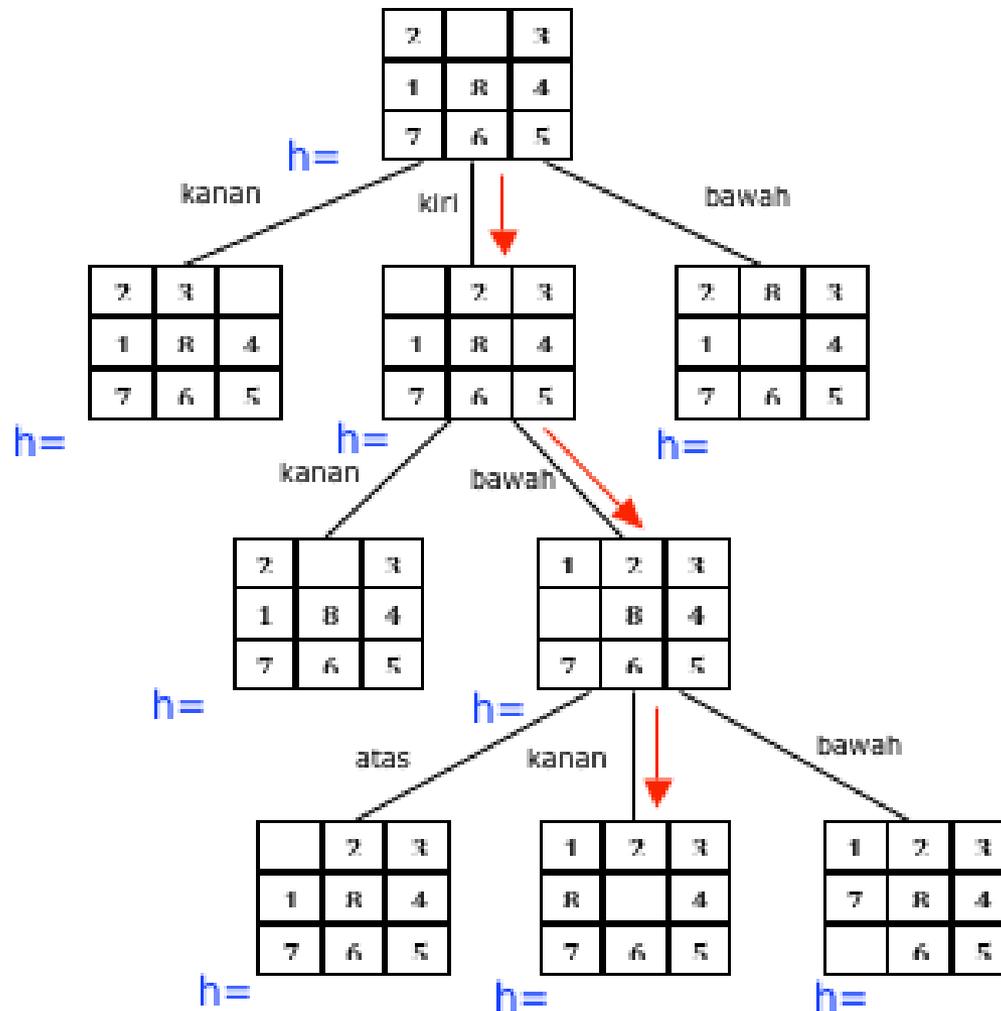
□□ Ubin kosong ke kiri

□□ Ubin kosong ke atas

□□ Ubin kosong ke bawah

# 8-puzzle (2/4)

- Dengan menggunakan bentuk pohon untuk merepresentasikan ruang keadaan, gunakanlah metode *Steepest Ascent Hill Climbing* untuk mencari langkah-langkah yang harus ditempuh dari keadaan awal sampai mendapatkan tujuan.
- Fungsi heuristik yang digunakan adalah jumlah ubin yang menempati posisi yang benar.



Gambar 1 Penyelesaian 8-puzzle dengan *steepest ascent hill climbing*.

# 8-puzzle (4/4) : Keterangan

- Pada Gambar 1 terlihat bahwa semula ada 3 operator yang bisa digunakan, yaitu ubin kosong digeser ke kanan, kiri dan bawah. Masing-masing kondisi hasil dari implementasi operator memberikan nilai heuristik 4, 6, dan 5. Nilai heuristik terbesar adalah 6, sehingga kondisi kedua yang dipilih.
- Pada tahap kedua, operator yang bisa digunakan hanya 2, yaitu ubin kosong digeser ke kanan dan ke bawah. masing- masing dengan nilai heuristik 5 dan 7. Nilai heuristik yang dihasilkan adalah 7, dan kondisi ini dipilih.
- Pada tahap ketiga, ada 3 operator yang digunakan, yaitu ubin kosong digeser ke atas, kanan, dan bawah. Masing- masing kondisi memiliki nilai heuristik 6, 8, dan 6. Karena nilai 8 merupakan solusi, maka pencarian telah berakhir.

# Latihan 1:

- Tuliskan langkah-langkah untuk menyelesaikan masalah 8-puzzle dengan menggunakan algoritma steepest ascent hill climbing untuk kasus dibawah ini:

**KEADAAN AWAL**

2		3
1	4	5
7	8	6



**TUJUAN**

1	2	3
4	5	6
7	8	

# Fungsi Heuristic: Manhattan Distance

Heuristic Manhattan distance digunakan untuk menghitung jumlah langkah yang dibutuhkan untuk mengubah keadaan awal menjadi solusi.

Kita hanya menghitung jumlah dari setiap ubin puzzle.

Contoh, Manhattan distance antara "123456780" adalah 5 dan "23145786" adalah 5

State : 2 3 1 4 5 7 8 6

KEADAAN AWAL

2		3
1	4	5
7	8	6



TUJUAN

1	2	3
4	5	6
7	8	

**Manhattan distance:**  $1+0+1+1+1+0+0+1=5$

## Latihan 2:

- Tuliskan langkah-langkah untuk menyelesaikan masalah 8-puzzle dengan menggunakan algoritma **steepest ascent hill climbing** dengan fungsi heuristic **manhattan distance** untuk kasus dibawah ini:

**KEADAAN AWAL**

2		3
1	8	4
7	6	5

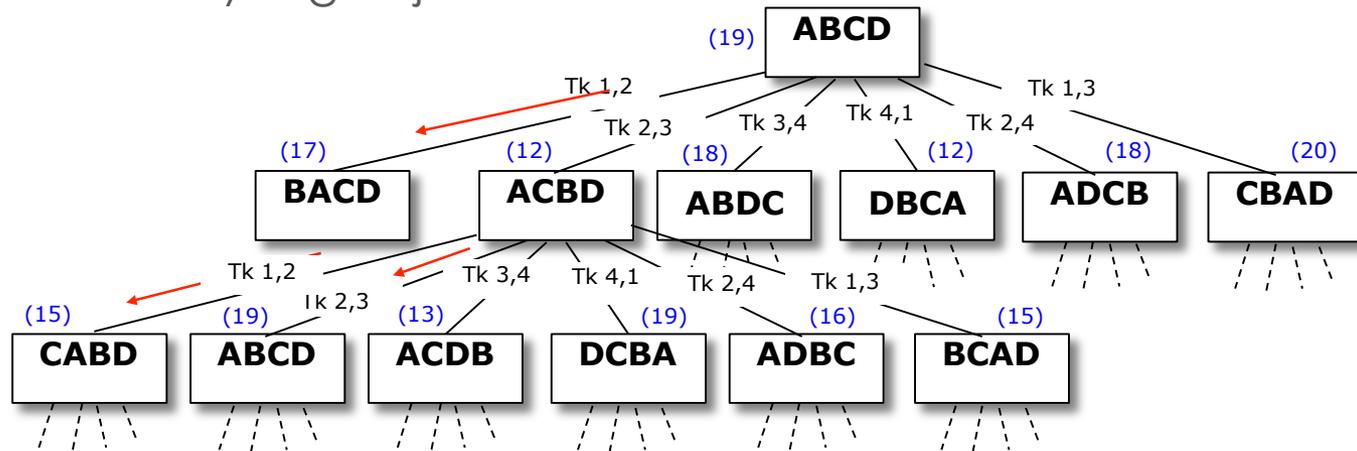


**TUJUAN**

1	2	3
8		4
7	6	5

# Contoh 2: Travelling Salesman Problem dengan steepest ascent hill climbing

- Menggunakan 6 operator untuk menukar posisi kota-kota yang bersebelahan.
- Fungsi heuristik yang digunakan adalah panjang lintasan yang terjadi.

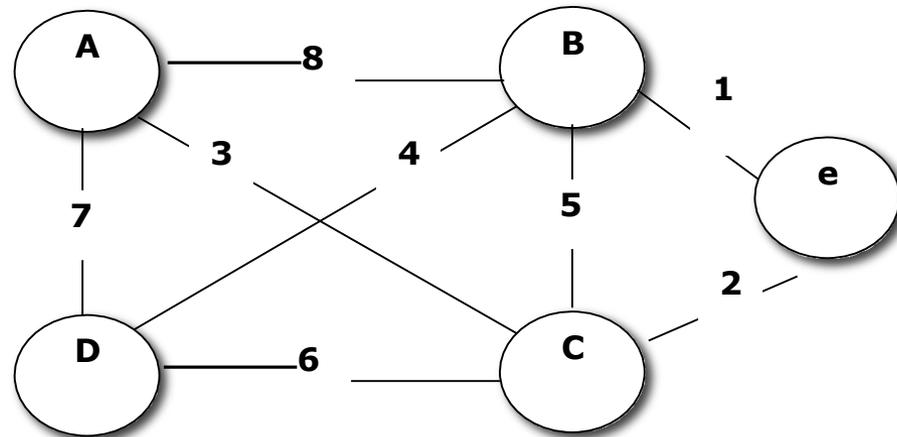


# Keterangan

- ■ Pada Gambar 2.20, terlihat bahwa, keadaan awal, lintasan terpilih adalah ABCD (19).
- ■ Pada level pertama, hill climbing akan memilih nilai heuristik terbaik dari keenam succesor yang ada, yaitu: BACD(17), ACBD (12), ABDC(18), DBCA(12), ADCB (18) atau CBAD(20). Tentu saja yang terpilih adalah ACBD, karena memiliki nilai heuristik paling kecil (=12).
- ■ Dari ACBD ini akan dipilih nilai heuristik terbaik dari succesornya yaitu: CABD(15), ABCD(19), ACDB(13), DCBA(19), ADBC(16) atau BCAD(15). Ternyata dari keenam successor tersebut memiliki nilai heuristik yang lebih besar disbanding dengan ACDB.
- ■ Sehingga tidak akan ada perubahan nilai keadaan (tetap ACDB). Hasil yang diperoleh, lintasannya adalah ACDB (12).

# Latihan 3:

- Selesaikan masalah Travelling Salesman Problem dengan menggunakan algoritma **simple hill climbing** dan **steepest ascent hill climbing** untuk kota-kota seperti pada peta berikut ini:



## Contoh 3 : Petani, Srigala, Sayuran, Kambing

- ■ Seorang petani akan menyeberangkan seekor kambing, seekor serigala, dan sayur-sayuran dengan sebuah boat yang melalui sungai. Boat hanya bisa memuat petani dan satu penumpang yang lain (kambing, serigala atau sayur-sayuran). Jika ditinggalkan oleh petani tersebut, maka sayur-sayuran akan dimakan oleh kambing, dan kambing akan dimakan oleh serigala.

# Petani, Srigala, Sayuran, Kambing



# Penyelesaian

## 1. Identifikasi ruang keadaan

□□ Permasalahan ini dapat dilambangkan dengan (JumlahKambing, JumlahSerigala, JumlahSayuran, JumlahBoat).

□□ Sebagai contoh: Daerah asal  $(0,1,1,1)$  berarti pada daerah asal tidak ada kambing, ada serigala, ada sayuran, dan ada boat.

# Penyelesaian (2/4)

## 2. Keadaan awal & tujuan

■ ■ Keadaan awal, pada kedua daerah:

■ ■ Daerah asal:  $(1,1,1,1)$

■ ■ Daerah seberang:  $(0,0,0,0)$

■ ■ Tujuan, pada kedua daerah:

■ ■ Daerah asal:  $(0,0,0,0)$

■ ■ Daerah seberang:  $(1,1,1,1)$

# Penyelesaian (3/4)

## 2. Aturan-aturan

Aturan-aturan dapat digambarkan seperti pada Tabel berikut:

<b>Aturan ke-</b>	<b>Aturan</b>
1.	Kambing menyeberang
2.	Sayuran menyeberang
3.	Serigala menyeberang
4.	Kambing kembali
5.	Sayuran kembali
6.	Serigala kembali
7.	Boat kembali

# Penyelesaian (4/4)

## 4. Solusi

Tabel solusi dapat dilihat seperti berikut:

Daerah Asal	Daerah Seberang	Aturan yang dipakai
(1,1,1,1)	(0,0,0,0)	1
(0,1,1,0)	(1,0,0,1)	7
(0,1,1,1)	(1,0,0,0)	3
(0,0,1,0)	(1,1,0,1)	4
(1,0,1,1)	(0,1,0,0)	2
(1,0,0,0)	(0,1,1,1)	7
(1,0,0,1)	(0,1,1,0)	1
(0,0,0,0)	(1,1,1,1)	solusi

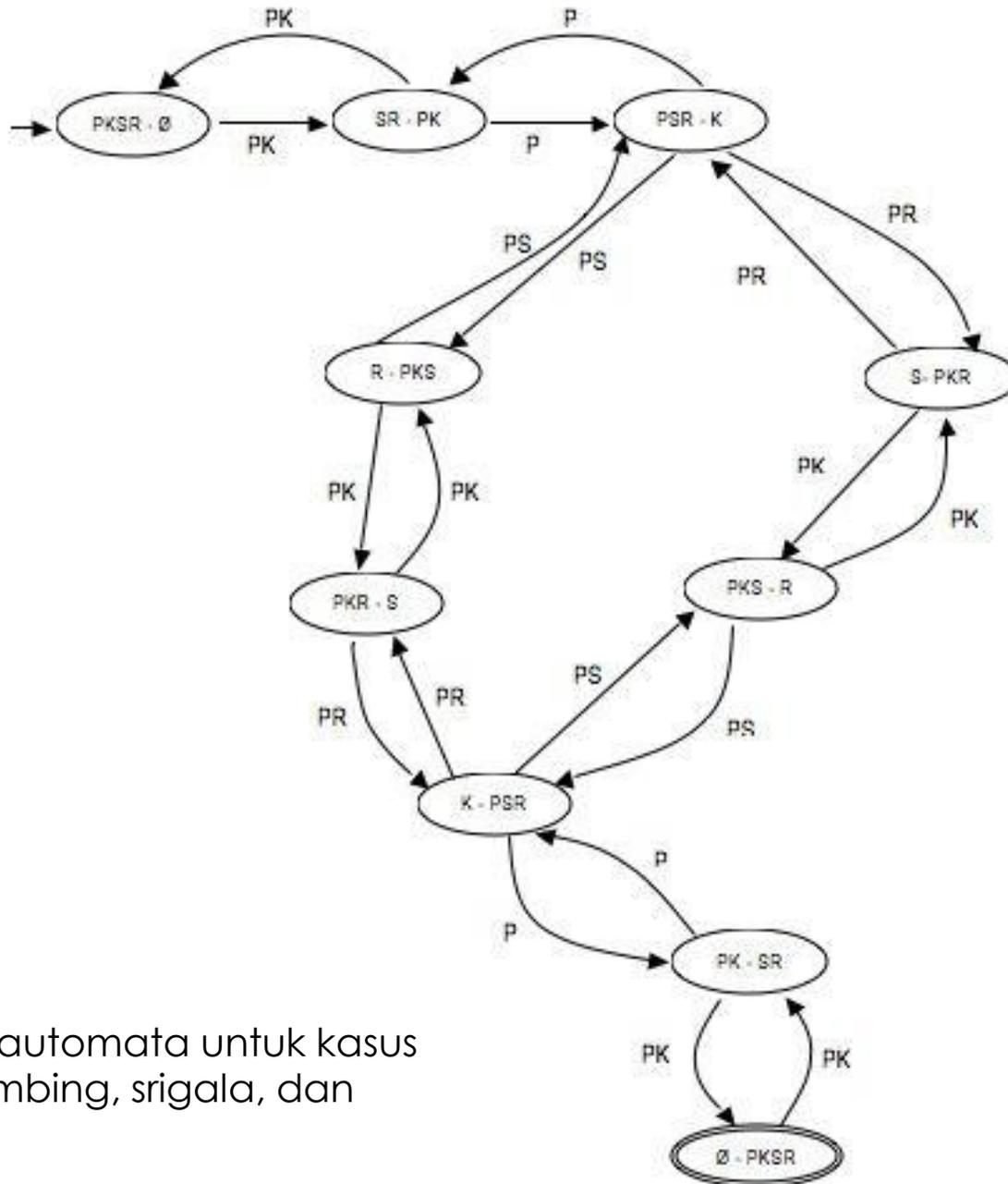
# Penyelesaian lain dengan menggunakan Finite State Automata

- ■ Masalah ini dapat dimodelkan dengan memperhatikan mereka yang menepati setiap sisi sungai.
- ■ Misalnya, kita akan menggunakan kombinasi pada setiap sisi sungai sebagai keadaan (state) yang mungkin.
- ■ Terdapat 16 kombinasi state untuk Petani (P), Kambing (K), Srigala(S), dan Rumput (R).

Sisi kiri	Sisi Kanan	Simbol State
PSKR	$\emptyset$	PSKR – $\emptyset$
SR	PK	SR – PK
SK	PR	SK – PR
KR	PS	KR – PS
PSR	K	PSR – K
PSK	R	PSK – R
PKR	S	PKR – S
PK	SR	PK – SR
PR	SK	PR – SK
PS	KR	PS – KR
K	PSR	K – PSR
R	PSK	R – PSK
S	PKR	S – PKR
SKR	P	SKR – P
P	SKR	P – SKR
$\emptyset$	PSKR	$\emptyset$ – PSKR

Dari 16 kombinasi state tersebut, hanya 10 state yang mungkin, yaitu :

Sisi kiri	Sisi Kanan	Simbol State
PSKR	$\emptyset$	PSKR – $\emptyset$
SR	PK	SR – PK
PSR	K	PSR – K
PSK	R	PSK – R
PKR	S	PKR – S
PK	SR	PK – SR
K	PSR	K – PSR
R	PSK	R – PSK
S	PKR	S – PKR
$\emptyset$	PSKR	$\emptyset$ – PSKR



Final state automata untuk kasus petani, kambing, srigala, dan rumput.