



# PENCARIAN DAN PELACAkan HEURISTIK

# OUTLINE

- Definisi Pencarian Heuristik
- Empat Metode Pencarian Heuristik
- Pembangkitan & Pengujian (Generate & Test)
- Pembangkitan Bukit (Hill Climbing)
- Pencarian Terbaik Pertama (Best First Search)



# DEFINISI PENCARIAN HEURISTIK

- Heuristik adalah sebuah teknik yang mengembangkan efisiensi dalam proses pencarian, namun dengan kemungkinan mengorbankan kelengkapan (completeness).
- Fungsi heuristik digunakan untuk mengevaluasi keadaan-keadaan problema individual dan menentukan seberapa jauh hal tersebut dapat digunakan untuk mendapatkan solusi yang diinginkan.

I.4

# EMPAT METODE PENCARIAN HEURISTIK

- A. Pembangkit & Pengujian (Generate and Test)
- B. Pendakian Bukit (Hill Climbing)
- C. Pencarian terbaik Pertama (Best First Search)
- D. Alpha Beta Prunning, **Means-End-Anlysis**, **Constraint Satisfaction**, Simulated Anealing, dll

# PEMBANGKITAN & PENGUJIAN

## (Generate and Test)

- Penggabungan antara *depth-first search* dengan *pelacakan mundur (backtracking)*, yaitu bergerak ke belakang menuju pada suatu keadaan awal.
- Bergerak ke belakang menuju pada suatu keadaan awal
- Nilai Pengujian berupa jawaban “ya” atau “tidak”

# PEMBANGKITAN & PENGUJIAN

## (Generate and Test)

### *Algoritma :*

1. Bangkitkan suatu kemungkinan solusi (membangkitkan suatu titik tertentu atau lintasan tertentu dari keadaan awal).
2. Uji untuk melihat apakah node tersebut benar-benar merupakan solusinya dengan cara membandingkan node tersebut atau node akhir dari suatu lintasan yang dipilih dengan kumpulan tujuan yang diharapkan.
3. Jika solusi ditemukan, keluar. Jika tidak, ulangi kembali langkah pertama.

# PEMBANGKITAN & PENGUJIAN

## (Generate and Test)

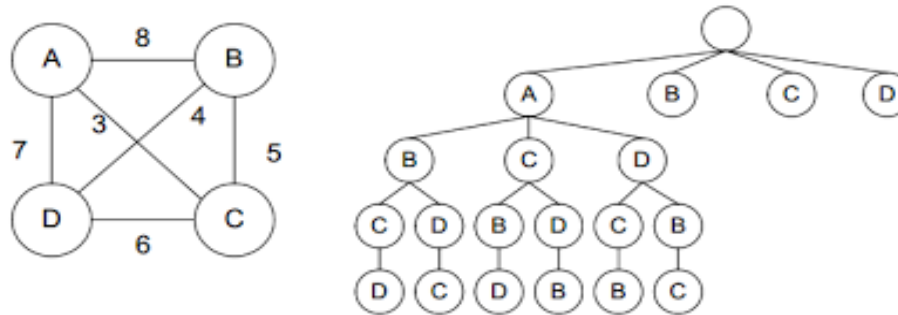
*Contoh* : “Travelling Salesman Problem (TSP)”

Seorang salesman ingin mengunjungi  $n$  kota. Jarak antara tiap-tiap kota sudah diketahui. Kita ingin mengetahui ruter terpendek dimana setiap kota hanya boleh dikunjungi tepat 1 kali. Misalkan ada 4 kota dengan jarak antara tiap-tiap kota seperti berikut ini :

# PEMBANGKITAN & PENGUJIAN

## (Generate and Test)

**Contoh** : “Travelling Salesman Problem (TSP)”



Alur pencarian dengan *Generate and Test*

Pencarian ke-	Lintasan	Panjang Lintasan	Lintasan terpilih	Panjang Lintasan terpilih
1	ABCD	19	ABCD	19
2	ABDC	18	ABDC	18
3	ACBD	12	ACBD	12
4	ACDB	13	ACBD	12
5	ADBC	16	ACBD	12
Dst.....				

# PENDAKIAN BUKIT

## (Hill Climbing)

- Proses pengujian dilakukan dengan menggunakan fungsi heuristic.
- Pembangkitan keadaan berikutnya tergantung pada feedback.
- Tes yang berupa fungsi heuristic ini akan menunjukkan seberapa baiknya nilai terkaan yang diambil terhadap keadaan-keadaan lainnya yang mungkin.

# PENDAKIAN BUKIT

## (Hill Climbing)

Metode ini hampir sama dengan metode pembangkitan dan pengujian, hanya saja proses pengujian dilakukan dengan menggunakan fungsi heuristic. Pembangkitan keadaan berikutnya tergantung pada feedback dari prosedur pengetesan. Tes yang berupa fungsi heuristic ini akan menunjukkan seberapa baiknya nilai terkaan yang diambil terhadap keadaan-keadaan lainnyayang mungkin.

# PENDAKIAN BUKIT

## (Hill Climbing)

### *Algoritma:*

1. Cari operator yang belum pernah digunakan; gunakan operator ini untuk mendapatkan keadaan yang baru.
  - a) Kerjakan langkah-langkah berikut sampai solusinya ditemukan atau sampai tidak ada operator baru yang akan diaplikasikan pada keadaan sekarang : Cari operator yang belum digunakan; gunakan operator ini untuk mendapatkan keadaan yang baru.
  - b) Evaluasi keadaan baru tersebut : – Jika keadaan baru merupakan tujuan, keluar – Jika bukan tujuan, namun nilainya lebih baik daripada keadaan sekarang, maka jadikan keadaan baru tersebut menjadi keadaan sekarang. – Jika keadaan baru tidak lebih baik daripada keadaan sekarang, maka lanjutkan iterasi.

# PENDAKIAN BUKIT

## (Hill Climbing)

### *Contoh:*

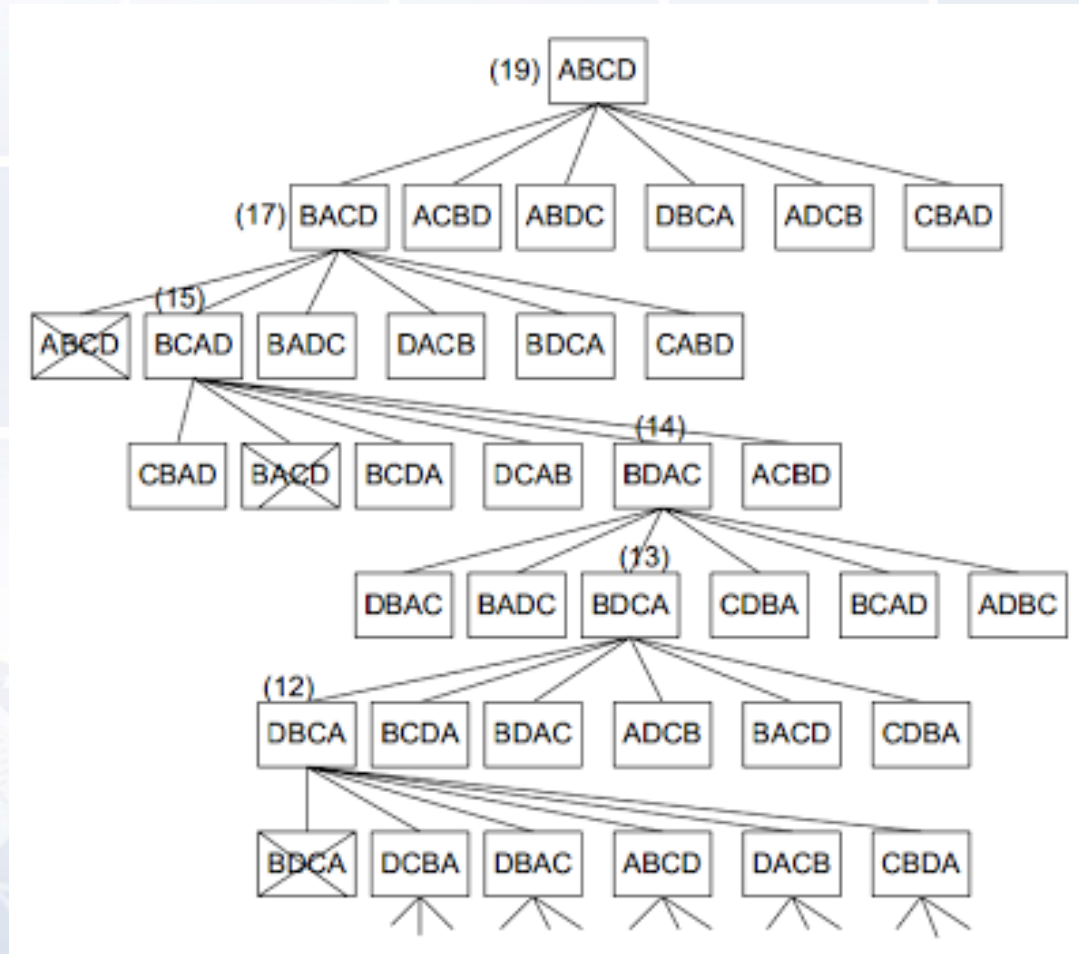
TSP dengan Simple Hill Climbing Disini ruang keadaan berisi semua kemungkinan lintasan yang mungkin.

Operator digunakan untuk menukar posisi kota-kota yang bersebelahan. Apabila ada  $n$  kota, dan kita ingin mencari kombinasi lintasan dengan menukar posisi urutan 2 kota, maka kita akan mendapatkan sebanyak  $n!/2!(n-2)!$

atau sebanyak 6 kombinasi. *Fungsi heuristic yang digunakan adalah panjang lintasan yang terjadi.*

# PENDAKIAN BUKIT (Hill Climbing)

- Contoh:*



# PENCARIAN TERBAIK PERTAMA (BEST FIRST SEARCH)

- Kombinasi dari metode depth-first search dan breadth-first search.
- Pencarian diperbolehkan mengunjungi node yang ada di level yang lebih rendah
- jika ternyata node pada level yang lebih tinggi ternyata memiliki nilai heuristic yang lebih buruk.

# PENCARIAN TERBAIK PERTAMA (BEST FIRST SEARCH)

- Fungsi Heuristik yang digunakan merupakan prakiraan (estimasi) cost dari initial state ke goal state, yang dinyatakan dengan :
- $f'(n) = g(n) + h'(n)$ 
  - $f'$  = Fungsi evaluasi
  - $g$  = cost dari initial state ke current state
  - $h'$  = prakiraan cost dari current state ke goal state

# PENCARIAN TERBAIK PERTAMA (BEST FIRST SEARCH)

## *Contoh:*

Misalkan kita memiliki ruang pencarian seperti pada gambar berikut. Node M merupakan keadaan awal dan node T merupakan tujuannya. Biaya edge yang menghubungkan node M dengan node A adalah biaya yang dikeluarkan untuk bergerak dari kota M ke kota A. Nilai  $g$  diperoleh berdasarkan biaya edge minimal. Sedangkan nilai  $h'$  di node A merupakan hasil perkiraan terhadap biaya yang diperlukan dari node A untuk sampai ke tujuan.  $h'(n)$  bernilai  $\infty$  jika sudah jelas tidak ada hubungan antara node  $n$  dengan node tujuan (jalan buntu). Kita bisa merunut nilai untuk setiap node.

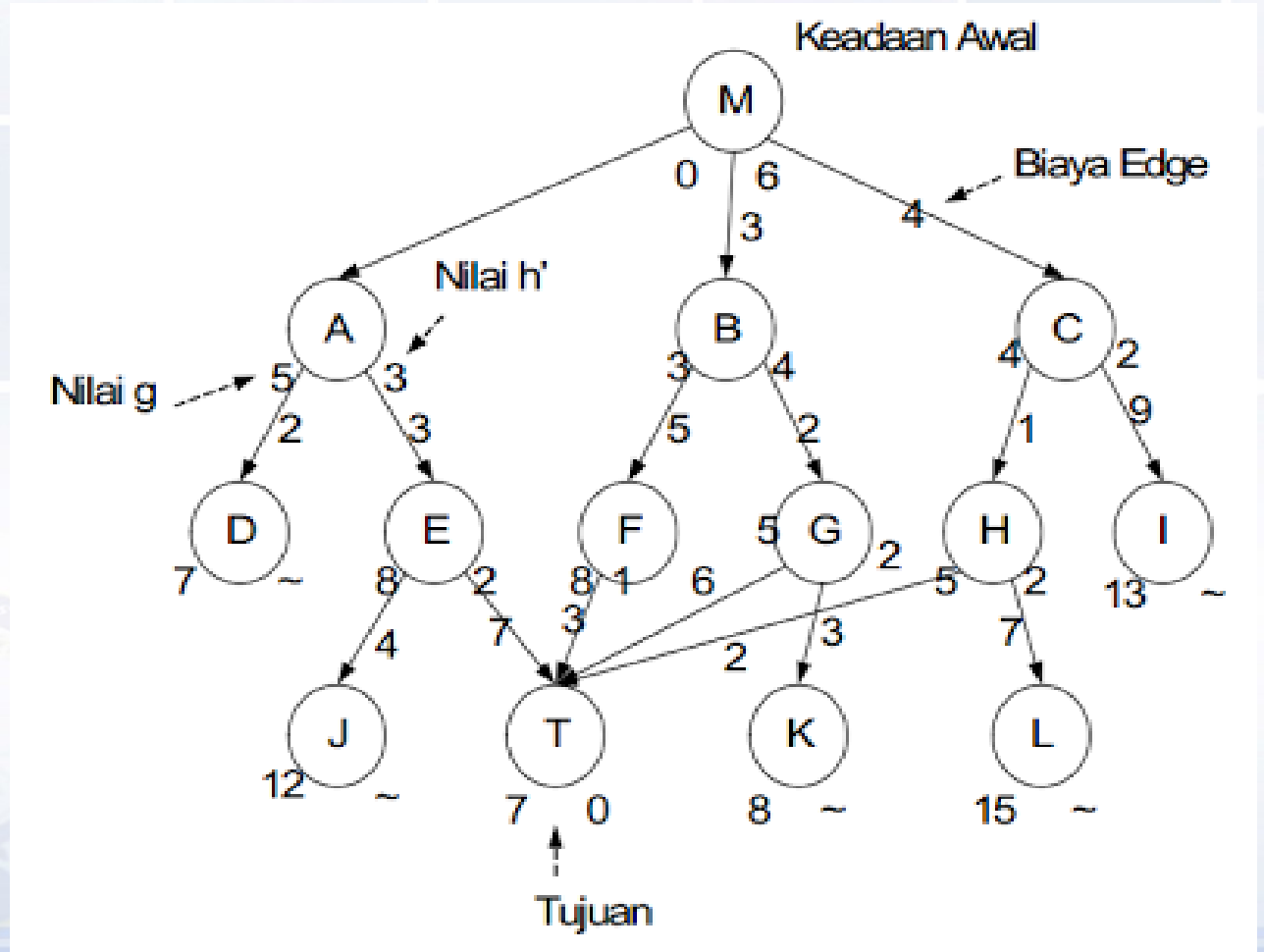
# PENCARIAN TERBAIK PERTAMA (BEST FIRST SEARCH)

*Contoh:*

Tabel status tiap node

Node (n)	g(n)	h'(n)	f(n)
Tabel Status Tiap Node			
A	5	3	8
B	3	4	7
C	4	2	6
D	7	~	~
E	8	2	10
F	8	1	9
G	5	2	7
H	5	2	7
I	13	~	~
J	12	~	~
K	8	~	~
L	15	~	~
T	7	0	7

Solusi : M-C H-T



# Means-End-Anlysis

- MEA adalah strategi penyelesaian masalah yang diperkenalkan pertama kali dalam GPS (General Problem Solver) [Newell & Simon, 1963].
- Proses pencarian berdasarkan ruang masalah yang menggabungkan aspek penalaran forward dan backward.
- Perbedaan antara state current dan goal digunakan untuk mengusulkan operator yang mengurangi perbedaan itu.
- Keterhubungan antara operator dan perbedaan tsb disajikan sebagai pengetahuan dalam sistem (pada GPS dikenal dengan Table of Connections) atau mungkin ditentukan sampai beberapa pemeriksaan operator jika tindakan operator dapat dipenetrasi.

# Means-End-Anlysis

- Contoh OPERATOR first-order predicate calculus dan operator2 tertentu mengijinkan perbedaan korelasi task-independent terhadap operator yang menguranginya.
- Kapan pengetahuan ada tersedia mengenai pentingnya perbedaan, perbedaan yang paling utama terpilih pertama lebih lanjut meningkatkan rata-rata capaian dari MEA di atas strategi pencarian Brute-Force.
- Bagaimanapun, bahkan tanpa pemesanan dari perbedaan menurut arti penting, MEA meningkatkan metode pencarian heuristik lain (di rata-rata kasus) dengan pemusatan pemecahan masalah pada perbedaan yang nyata antara current state dengan goal-nya.

# Constraint Satisfaction

Problem search standard :

- state adalah "black box"
- setiap struktur data yang mendukung fungsi successor, fungsi heuristik dan tes goal.

CSP:

- state didefinisikan sebagai variabel  $X_i$  dengan nilai dari domain  $D_i$  – Tes goal adalah sekumpulan constraint yang menspesifikasikan kombinasi dari nilai subset variabel.

Contoh sederhana adalah bahasa representasi formal.

CSP ini merupakan algoritma general-purpose dengan kekuatan lebih daripada algoritma pencarian standar.

# Constraint Satisfaction

*Contoh* : Pewarnaan Peta

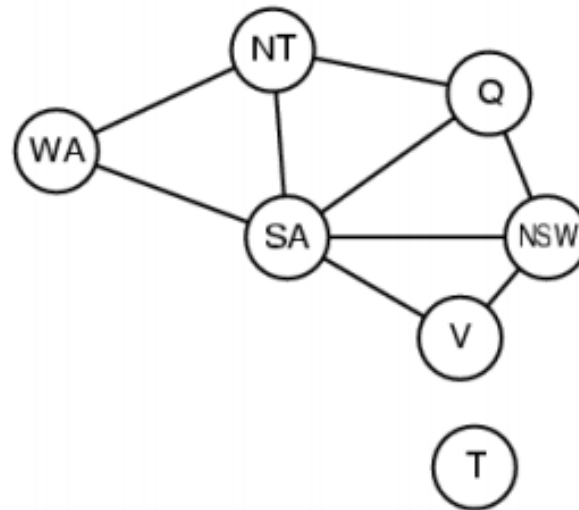


# Constraint Satisfaction

- Variabel WA, NT, Q, NSW, V, SA, T
- Domain  $D_i = \{\text{red, green, blue}\}$
- Constraints : daerah yang bertetangga dekat harus memiliki warna yang berbeda.
- Contoh  $WA \neq NT$ , atau  $(WA, NT)$   
 $\{(\text{red, green}), (\text{red, blue}), (\text{green, red}),$   
 $(\text{green, blue}), (\text{blue, red}), (\text{blue, green})\}$
- Solusi lengkap dan konsisten, contoh :  $WA = \text{red}, NT = \text{green}, Q = \text{red}, NSW = \text{green}, V = \text{red}, SA = \text{blue}, T = \text{green}$

# Constraint Graf

- Binary CSP biner : setiap constraint merelasikan dua variabel
- Graf Constraint : node adalah variabel, arc adalah constraint



😊 **END** 😊

