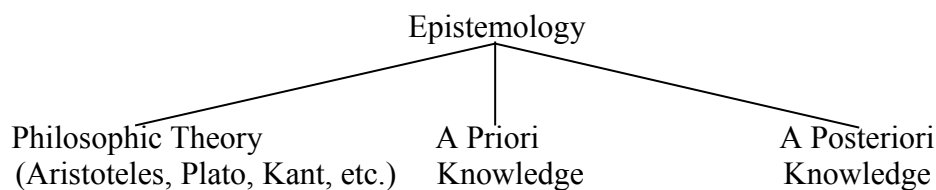


BAB 2 REPRESENTASI PENGETAHUAN

Pengetahuan (*Knowledge*)

- Definisi umum : fakta atau kondisi sesuatu atau keadaan yang timbul karena suatu pengalaman
- Pelajaran dari *knowledge* merupakan suatu *epistemology* yang merupakan bagian dari ilmu filsafat yang berkenaan dengan sifat, struktur dan keaslian dari *knowledge*.



- Priori Knowledge
 - Berarti yang mendahului (*pengetahuan datang sebelumnya dan bebas dari arti*)
 - Kebenaran yang universal dan tidak dapat disangkal tanpa kontradiksi
 - Contoh : pernyataan logika, hukum matematika
- Posteriori Knowledge
 - Knowledge yang diturunkan dari akal pikiran yang sehat.
 - Kebenaran atau kesalahan dapat dibuktikan dengan menggunakan pengalaman akal sehat.
 - Contoh : bola mata seseorang berwarna biru, tetapi ketika orang tersebut mengganti contact lens-nya, bisa jadi bola matanya menjadi berwarna hijau.
- Kategori Knowledge :
 - Procedural Knowledge
Bagaimana melakukan sesuatu
 - Declarative Knowledge
Mengetahui sesuatu itu benar atau salah
 - Tacit Knowledge
Tidak dapat diungkapkan dengan bahasa
- Knowledge pada ES
 - Analogi dengan ekspresi klasik Wirth :
ALGORITMA + STRUKTUR DATA = PROGRAM
 - Knowledge pada ES :
KNOWLEDGE + INFERENSI = ES

Teknik Representasi Pengetahuan

- 1) Aturan Produksi
- 2) Jaringan Semantik
- 3) Frame, Scemata, dan Script
- 4) Logika dan Himpunan

1. Aturan Produksi

- sering digunakan untuk merepresentasikan pengetahuan pada system pakar
- bentuk formalnya **Backus-Naus Form (BNF)**
 - ✓ metalanguange untuk mendefinisikan sintaks bahasa
 - ✓ suatu grammar haruslah lengkap dan unambiguous set dari aturan produksi untuk bahasa yang spesifik
 - ✓ *parse tree* adalah representasi grafis dari kalimat pada suatu bahasa
 - ✓ deskripsi sintaks tersedia dalam bahasa
 - ✓ tidak semua kalimat adalah benar

- Contoh :

$\langle \text{sentence} \rangle ::= \langle \text{subject} \rangle \langle \text{verb} \rangle \langle \text{end-mark} \rangle$

dimana,

- ◆ $\langle .. \rangle$ dan $::=$ adalah symbol metalanguange.
- ◆ $::=$ artinya “ditentukan sebagai” yang dalam BNF ekuivalen dengan \rightarrow .
- ◆ Term di dalam kurung disebut symbol **Nonterminal**, yang masih bisa direpresentasikan ke dalam bentuk lebih sederhana lagi.
- ◆ Nonterminal yang tidak dapat disederhanakan lagi disebut **Terminal**.

$\langle \text{sentence} \rangle \rightarrow \langle \text{subject} \rangle \langle \text{verb} \rangle \langle \text{end-mark} \rangle$

$\langle \text{subject} \rangle \rightarrow I \mid \text{You} \mid \text{We}$

$\langle \text{verb} \rangle \rightarrow \text{left} \mid \text{came}$

$\langle \text{end-mark} \rangle \rightarrow . \mid ? \mid !$

Produksinya :

I left.

You came?

We left ! dst.....

Keuntungan Aturan Produksi :

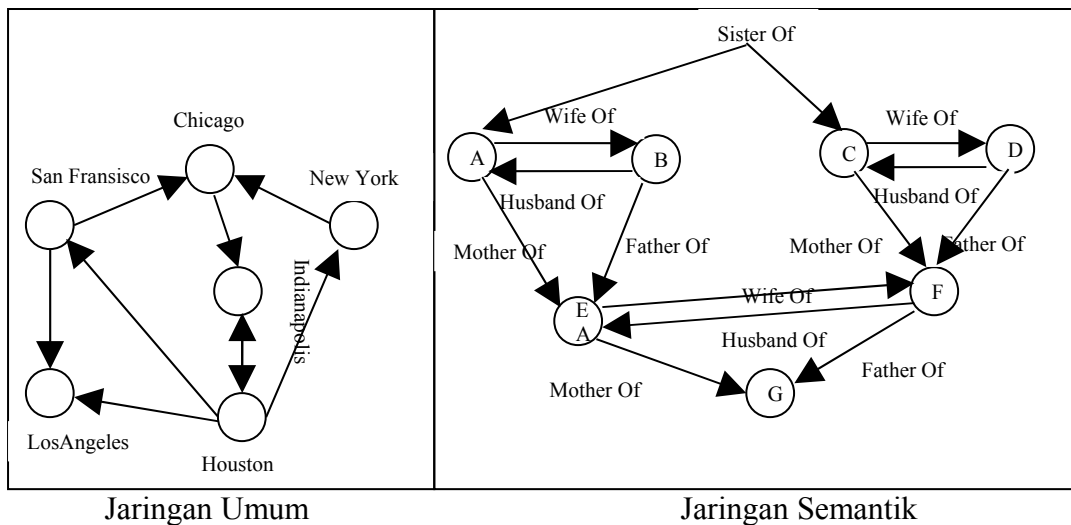
- Sederhana dan mudah dipahami
- Dasar bagi berbagai variant

Kelemahan Aturan Produksi :

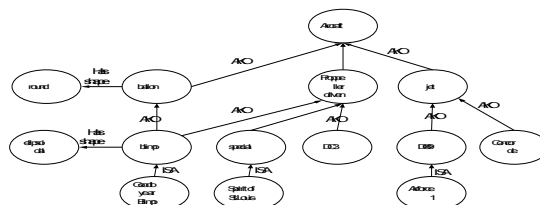
- Implementasi yang sederhana sering menyebabkan inefisien
- Beberapa tipe pengetahuan sulit direpresentasikan dalam aturan produksi

2. Jaringan Semantik

- Dibangun oleh M.R.Quillian, sebagai model memori manusia.
- Representasi grafis dari informasi Proporsional.
- Proposisi adalah pernyataan yang dapat bernilai benar atau salah.
- Disajikan dalam bentuk graf berarah
- Node merepresentasikan konsep, objek atau situasi :
 - Label ditunjukkan melalui penamaan
 - Node dapat berupa objek tunggal atau kelas
- Links merepresentasikan suatu hubungan :
 - Links adalah struktur dasar untuk pengorganisasian pengetahuan
 - Contoh jaringan semantic.



- Tipe link :
 - IS-A (ISA) berarti “contoh dari” dan merupakan anggota tertentu dari kelas.
 - A KIND OF (AKO) berarti “jenis dari” dan merelasikan antara suatu kelas dengan kelas lainnya. AKO merelasikan kelas individu ke kelas induk dari kelas-kelas dimana individu tersebut merupakan kelas anak.
 - HAS-A berarti “mempunyai” yang merelasikan suatu kelas menjadi subkelas. HAS-A berlawanan dengan AKO dan sering digunakan untuk merelasikan suatu objek ke bagian dari objek.



Keterangan : AKO = jenis dari
 ISA = contoh

Has shape = berbentuk

Ada 3 hal yaitu **OBJECT, ATTRIBUTE, VALUE (OAV)** Triplet, yang sering digunakan untuk membangun jaringan semantic.

1. **OBJECT** : dapat berupa fisik atau konsepsi
2. **ATTRIBUTE** : karakteristik objek
3. **VALUE** : ukuran spesifik dari atribut dalam situasi tertentu

Contoh :

Objek	Atribut	Value
Apel	Warna	Merah
Apel	Tipe	MCintosh
Apel	Jumlah	100
Anggur	Warna	Merah
Anggur	Tipe	Seedless
Anggur	Jumlah	500

Triplet OAV secara khusus digunakan untuk merepresentasikan fakta dan pola guna menyesuaikan fakta dalam aturan yang *antecedent*. Jaringan semantic untuk beberapa sistem terdiri dari node untuk objek, atribut dan nilai yang dihubungkan dengan IS A dan HAS A.

3. Frame, Schemata, dan Script

Frame

- Frame (Minsky, 1975) dipandang sebagai struktur data static yang digunakan untuk merepresentasikan situasi-situasi yang telah dipahami dan stereotype.
- Frame digunakan untuk merepresentasikan pengetahuan stereotype atau pengetahuan yang didasarkan kepada karakteristik yang sudah dikenal yang merupakan pengalaman masa lalu.
- Frame berupa kumpulan slot-slot (representasi entitas sebagai struktur objek) yang merupakan atribut untuk mendeskripsikan pengetahuan berupa kejadian, lokasi, situasi ataupun elemen-elemen lain. Frame digunakan untuk representasi pengetahuan deklaratif.
- Contoh 1 :

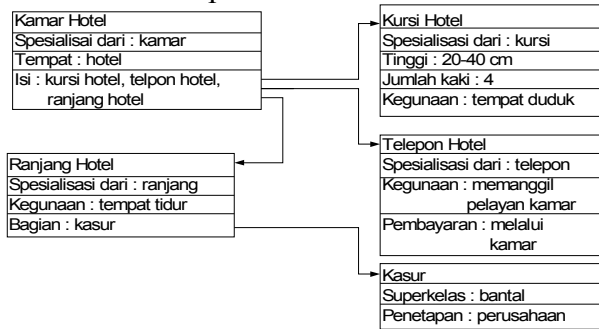
Frame Pohon

Spesialisasi dari : Tumbuhan
Jumlah batang : integer (default 1)
Jenis kulit : halus
Model daun : jenis pohon jarum, berganti daun
Bentuk daun : sederhana, berlekuk, campuran

Frame Pohon Perdu

Spesialisasi dari : Pohon
Jumlah batang : 3
Jenis kulit : halus
Model daun : berganti daun
Bentuk daun : sederhana, berlekuk

- Contoh 2 : Deskripsi frame untuk kamar hotel.



- Setiap frame individual dapat dipandang sebagai struktur data yang mirip record, berisi informasi yang relevan dengan entitas-entitas stereotype. Slot-slot dalam frame dapat berisi :
 - Informasi identifikasi frame
 - Hubungan frame dengan frame lain
 - Penggambaran persyaratan yang dibutuhkan frame
 - Informasi procedural untuk menggunakan struktur yang digambarkan
 - Informasi default frame
 - Informasi terbaru.

Slot merupakan kumpulan atribut / property yang menjelaskan objek yang direpresentasikan oleh frame. **Subslot** menjelaskan pengetahuan atau prosedur dari atribut pada slot.

Subslot dapat berupa :

- *Value* : menjelaskan tentang nilai dari suatu atribut
- *Default* : nilai yang digunakan jika suatu slot kosong atau tidak dideskripsikan pada frame instansiasi
- *Range* : menandakan jenis dari informasi yang dapat muncul pada slot tersebut (misal 0 sampai 100)
- *If Added* : berisi informasi procedural yang berupa suatu tindakan yang akan dikerjakan jika nilai dari slot diisi (atau berubah)
- *If Needed* : subslot ini digunakan pada kasus dimana tidak ada value pada slot. Suatu prosedur akan dikerjakan untuk memperoleh atau menghitung sebuah value.
- *Other* : slot bisa berisi frame, rule, jaringan semantic ataupun tipe lain dari informasi.

SCHEMATA

- ✓ Jaringan semantik contoh dari **Shallow knowledge Structure** karena seluruh pengetahuan jaringan semantik diisikan dalam link dan node
- ✓ **Concept schema** : dengan skema tsb kita dapat menunjukkan konsep.
- ✓ Contoh konsep mengenai binatang, setiap orang mempunyai persepsi sendiri mengenai bintang (berkaki 4 atau 2 , berbulu atau bersisik dsb)

- ❖ Banyak pengetahuan yang digunakan dalam proses penalaran kita sehari-hari yang sudah pasti dan sudah dikenal dengan baik. Hal ini didasarkan kepada berbagai penampilan situasi dan objek-objek khusus, dan proses yang tak bervariasi.
- ❖ Pengetahuan semacam itu kita sebut pengetahuan *stereotype*.
- ❖ Skema adalah satu metoda pengorganisasian, presentasi dan penggunaan pengetahuan stereotype agar komputer bisa menalar

Script

- Script (Schank & Abelson, Yale univ) merupakan representasi terstruktur yang menggambarkan urutan stereotip dari kejadian-kejadian dalam sebuah konteks khusus.
- Script mirip dengan frame, perbedaannya : Frame menggambarkan objek, sedangkan Script menggambarkan urutan peristiwa.
- Dalam menggambarkan urutan peristiwa, script menggunakan serangkaian slot yang berisi informasi tentang orang, objek dan tindakan-tindakan yang terjadi dalam suatu peristiwa.
- Elemen script yang tipikal :
 - ◆ Kondisi masukan : menggambarkan situasi yang harus dipenuhi sebelum terjadi suatu peristiwa yang ada dalam script.
 - ◆ Prop : mengacu kepada objek yang digunakan dalam urutan peristiwa yang terjadi.
 - ◆ Role : mengacu kepada orang-orang yang terlibat dalam script.
 - ◆ Hasil : kondisi yang ada sesudah peristiwa dalam script berlangsung.
 - ◆ Track : mengacu kepada variasi yang mungkin terjadi dalam script tertentu.
 - ◆ Scene : menggambarkan urutan peristiwa aktual yang terjadi.

- Contoh : Script pergi ke restoran
SCRIPT Restoran
Jalur (track) : fast food restoran
Peran (roles) : tamu, pelayan
Pendukung (prop) : conter, baki, makanan, uang, serbet, garam, merica, kecap, sedotan, dll
Kondisi masukan : tamu lapar –tamu punya uang
Adegan (scene) 1 : Masuk
 - Tamu parkir mobil
 - Tamu masuk restoran
 - Tamu antri
 - Tamu baca menu di list menu dan mengambil keputusan tentang apa yang akan diminta.

Adegan (scene) 2 : Pesanan

- Tamu memberikan pesanan pada pelayan
- Pelayan mengambil pesanan dan meletakkan makanan di atas baki
- Tamu membayar

Adegan (scene) 3 : Makan

- Tamu mengambil serbet, sedotan, garam, dll
- Tamu makan dengan cepat

Adegan (scene) 4 : Pulang

- Tamu membersihkan meja
- Tamu membuang sampah
- Tamu meninggalkan restoran
- Tamu naik mobil dan pulang

Hasil

- Tamu merasa kenyang
- Tamu senang
- Tamu kecewa
- Tamu sakit perut

- Keistimewaan Script :

1. Script menyediakan beberapa cara yang sangat alami untuk merepresentasikan “suatu informasi” yang lazim” dengan masalah yang bersumber dari sistem AI dari mula.
2. Script menyediakan struktur hirarki untuk merepresentasikan informasi melalui inklusi subscript dengan script.

4. Logika dan Himpunan

- Representasi pengetahuan dengan symbol logika merupakan bagian dari penalaran eksak.
- Bagian yang paling penting dalam penalaran adalah mengambil kesimpulan dari premis.
- Logika dikembangkan oleh filusuf Yunani, Aristoteles (abad ke 4 SM) didasarkan pada *silogisme*, dengan dua *premis* dan satu *konklusi*.

Contoh :

Premis : Semua laki-laki adalah makhluk hidup

Premis : Socrates adalah laki-laki

Konklusi : Socrates adalah makhluk hidup

Logika Proposisi

- Disebut juga kalkulus proposisi yang merupakan logika simbolik untuk memanipulasi proposisi.
- Proposisi merupakan pernyataan yang dapat bernilai *benar* atau *salah*.
- Operator logika yang digunakan :

Operator	Fungsi
\wedge	Konjungsi (AND/DAN)
\vee	Disjungsi (OR/ATAU)
\sim	Negasi (NOT/TIDAK)
\rightarrow	Implikasi/Kondisional (IF... THEN.../ JIKA... MAKA)
\leftrightarrow	Equivalensi/Bikondisional (IF AND ONLY IF / JIKA DAN HANYA JIKA) $p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$

- Kondisional merupakan operator yang analog dengan *production rule*.

Contoh 1 :

“ Jika hujan turun sekarang maka saya tidak pergi ke pasar”

Kalimat di atas dapat ditulis : $p \rightarrow q$

Dimana : p = hujan turun

q = saya tidak pergi ke pasar

Contoh 2 :

p = “Anda berusia 21 atau sudah tua”

q = “Anda mempunyai hak pilih”

Kondisional $p \rightarrow q$ dapat ditulis/berarti :

Kondisional	Berarti
$p \text{ implies } q$	Anda berusia 21 tahun atau sudah tua <i>implies</i> Anda mempunyai hak pilih.
Jika p maka q	Jika Anda berusia 21 tahun atau sudah tua, maka Anda mempunyai hak pilih.
p hanya jika q	Anda berusia 21 tahun atau sudah tua, hanya jika Anda mempunyai hak pilih.
p adalah (syarat cukup untuk q)	Anda berusia 21 tahun atau sudah tua adalah syarat cukup Anda mempunyai hak pilih.
q jika p	Anda mempunyai hak pilih, jika Anda berusia 21 tahun atau sudah tua.
q adalah (syarat perlu untuk p)	Anda mempunyai hak pilih adalah syarat perlu Anda berusia 21 tahun atau sudah tua.

- **Tautologi** : pernyataan gabungan yang selalu bernilai *benar*.
- **Kontradiksi** : pernyataan gabungan yang selalu bernilai *salah*.
- **Contingent** : pernyataan yang bukan tautology ataupun kontradiksi.

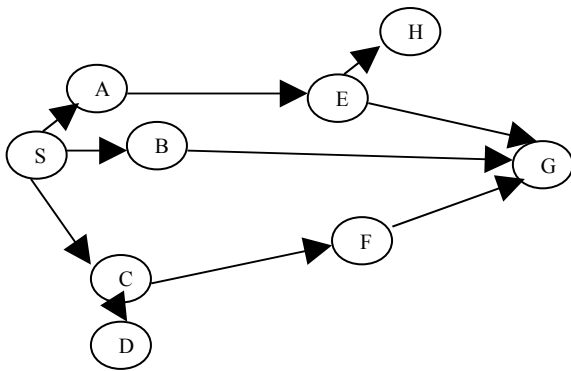
Tabel Kebenaran untuk logika konektif :

p	q	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$
T	T	T	T	T	T
T	F	F	T	F	F
F	T	F	T	T	F
F	F	F	F	T	T

Tabel kebenaran untuk negasi konektif :

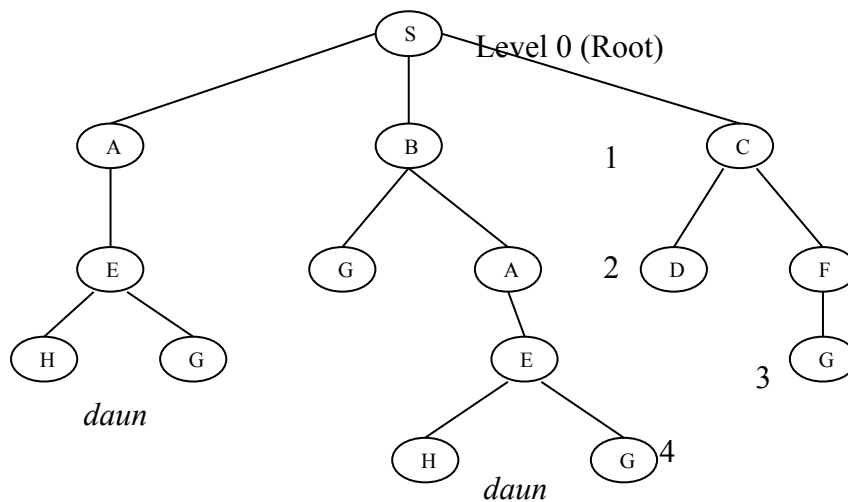
p	$\sim p$
T	F
F	T

STATE GRAPH



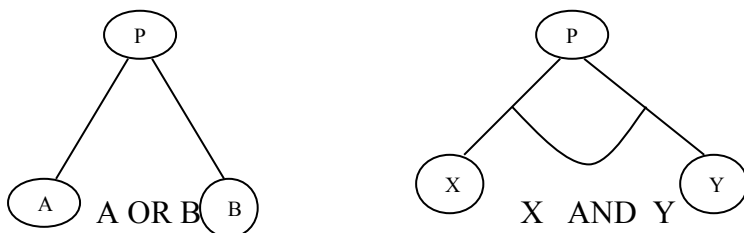
- Peta yg menunjukkan berbagai kota antara kota – kota yg akan dilaluinya agar mencapai kota tujuan yg diinginkan lebih cepat.
- A ... H : Node
- \longrightarrow : Ark/link

POHON PELACAKAN



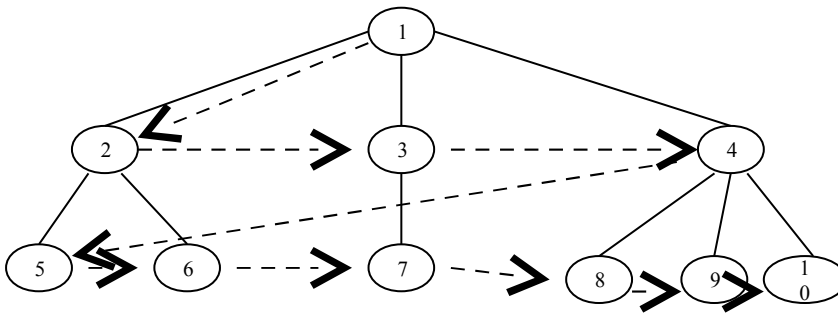
- Level \rightarrow sbg hirarki (menggambarkan kedalaman pohon)
- Node merupakan berbagai keadaan dlm ruang pelacakan. Ark \rightarrow operatornya.

POHON AND / OR

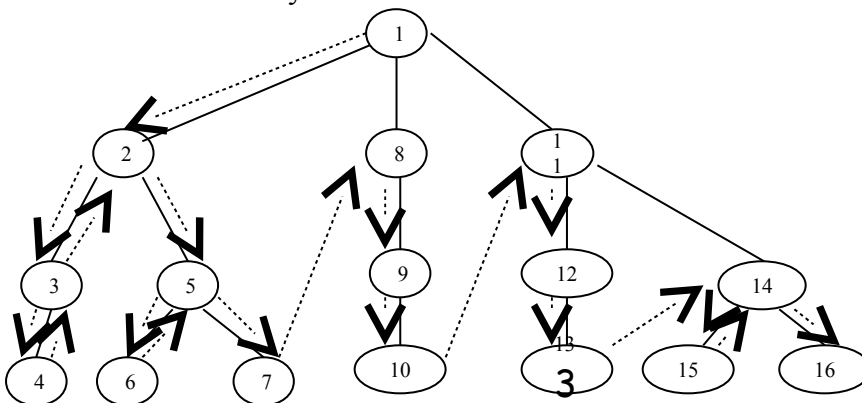


METODE PELACAKAN

1. **BLIND SEARCH (Pelacakan Buta)**
 - ✓ Merupakan sekumpulan prosedur yg digunakan dlm melacak ruang keadaan.
 - ✓ Menguji seluruh pohon dgn cara yg teratur dg menggunakan semua operator shg menghasilkan suatu solusi.
 - ✓ Lebih tepat u/ soal-soal kecil dg beberapa ruang keadaan dan tepat u/ komputer berkecepatan tinggi.
2. **BREADTH FIRST (Pelacakan Melebar Pertama)**
 - ✓ Menguji semua node dlm pohon pelacakan mulai dari node akar
 - ✓ Node yg ada pd setiap tingkat seluruhnya diuji sebelum pindah ke tingkat berikutnya.



3. **DEPTH FIRST (Pelacakan Pertama Mendalam)**
 - ✓ Jika keadaan tujuan tidak tercapai maka proses dilakukan dg jalan pelacakan backtrak ke node sebelumnya.
 - ✓ Menjamin bisa menemukan solusi *tapi waktu pelacakannya lama*.
 - ✓ Masalah utama : sering terjadi penyimpangan arah node tujuan yg sebenarnya.



4. **HEURISTIC SEARCH**
 - ✓ Istilah yg berasal dari bahasa Yunani yg berarti “ menemukan / menyingkap “
 - ✓ Membantu mengurangi wilayah pelacakan yg bisa menimbulkan berbagai alternatif solusi shg dapat membimbing ke tujuan yg diinginkan.
5. **HILL CLIMBING (Mendaki bukit)**
 - ✓ Merupakan pelacakan depth first yg memanfaatkan heuristik u/ menentukan jarak yg terpendek atau biaya terendah menuju tujuan yg diinginkan.
6. **BEST FIRST SEARCH (Pencarian Terbaik Pertama)**
 - ✓ Kombinasi dari breadth first dan depth first

