

# **BAHAN AJAR KECERDASAN BUATAN**



**R. JOKO MUSRIDHO, S.T., M.Phil**

**Universitas Pahlawan Tuanku Tambusai**

**2022**

## **BAB I**

### **PENGANTAR KECERDASAN BUATAN**

#### **1. 1 DEFINISI**

Kecerdasan buatan (Artificial Intelligence) :

Bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia.

Menurut John McCarthy, 1956, AI :

Untuk mengetahui dan memodelkan proses – proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia.

Cerdas = memiliki pengetahuan + pengalaman, penalaran (bagaimana membuat keputusan & mengambil tindakan), moral yang baik.

Agar mesin bisa cerdas (bertindak seperti & sebaik manusia) maka harus diberi bekal pengetahuan & mempunyai kemampuan untuk menalar.

2 bagian utama yg dibutuhkan untuk aplikasi kecerdasan buatan :

- a. basis pengetahuan (knowledge base): berisi fakta-fakta, teori, pemikiran & hubungan antara satu dengan lainnya.
- b. motor inferensi (inference engine) : kemampuan menarik kesimpulan berdasarkan pengetahuan

#### **1.2 BEDA KECERDASAN BUATAN & KECERDASAN ALAMI**

Kelebihan kecerdasan buatan :

1. Lebih bersifat permanen. Kecerdasan alami bisa berubah karena sifat manusia lupa. Kecerdasan buatan tidak berubah selama sistem komputer & program tidak mengubahnya.
2. Lebih mudah diduplikasi & disebarkan. Mentransfer pengetahuan manusia dari 1 orang ke orang lain membutuhkan proses yang sangat lama & keahlian tidak akan pernah dapat diduplikasi dengan lengkap. Jadi jika pengetahuan terletak pada suatu sistem komputer, pengetahuan tersebut dapat disalin dari komputer tersebut & dapat dipindahkan dengan mudah ke komputer yang lain.
3. Lebih murah. Menyediakan layanan komputer akan lebih mudah & murah dibandingkan mendatangkan seseorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.
4. Bersifat konsisten dan teliti karena kecerdasan buatan adalah bagian dari teknologi komputer sedangkan kecerdasan alami senantiasa berubah-ubah

5. Dapat didokumentasi. Keputusan yang dibuat komputer dapat didokumentasi dengan mudah dengan cara melacak setiap aktivitas dari sistem tersebut. Kecerdasan alami sangat sulit untuk direproduksi.
6. Dapat mengerjakan beberapa task lebih cepat dan lebih baik dibanding manusia

Kelebihan kecerdasan alami :

1. Kreatif : manusia memiliki kemampuan untuk menambah pengetahuan, sedangkan pada kecerdasan buatan untuk menambah pengetahuan harus dilakukan melalui sistem yang dibangun.
2. Memungkinkan orang untuk menggunakan pengalaman atau pembelajaran secara langsung. Sedangkan pada kecerdasan buatan harus mendapat masukan berupa input-input simbolik.
3. Pemikiran manusia dapat digunakan secara luas, sedangkan kecerdasan buatan sangat terbatas.

### 1.3 BEDA KECERDASAN BUATAN DAN PROGRAM KONVENSIONAL

	Kecerdasan Buatan	Program Konvensional
Fokus Pemrosesan	Konsep simbolik/numerik (pengetahuan)	Data dan informasi
Pencarian	Heuristik	Algoritma
Sifat input	Bisa tidak lengkap	Harus lengkap
Keterangan	Disediakan	Biasanya tidak disediakan
Struktur	Kontrol dipisahkan dari pengetahuan	Kontrol terintegrasi dengan informasi (data)
Sifat output	Kuantitatif	Kualitatif
Kemampuan menalar	Ya	Tidak

Program kecerdasan buatan dapat ditulis dalam semua bahasa komputer, baik dalam bahasa C, Pascal, Basic dan bahasa pemrograman lainnya. Tetapi dalam perkembangan selanjutnya dikembangkan bahasa pemrograman yang khusus untuk aplikasi kecerdasan buatan yaitu LISP dan PROLOG.

### 1.4 SEJARAH KECERDASAN BUATAN

Tahun 1950-an Alan Turing, seorang pionir AI dan ahli matematika Inggris melakukan percobaan Turing (Turing Test) yaitu sebuah computer melalui terminalnya ditempatkan pada jarak jauh. Di ujung yang satu ada terminal dengan software AI dan diujung lain ada sebuah terminal dengan seorang operator. Operator itu tidak mengetahui kalau diujung terminal lain dipasang software AI. Mereka berkomunikasi dimana terminal dikung memberikan respon terhadap serangkaian pertanyaan yang diajukan oleh operator. Dan sang operator itu mengira bahwa ia sedang berkomunikasi dengan operator lainnya

yang berada pada terminal lain.

Turing beranggapan bahwa jika mesin dapat membuat seseorang percaya bahwa dirinya berkomunikasi dengan orang lain, maka dapat dikatakan bahwa mesin tersebut cerdas (seperti layaknya manusia).

### 1.5 APLIKASI KECERDASAN BUATAN

Lingkup utama kecerdasan buatan :

1. Sistem pakar (expert system) : komputer sebagai sarana untuk menyimpan pengetahuan para pakar sehingga komputer memiliki keahlian menyelesaikan permasalahan dengan meniru keahlian yang dimiliki pakar.

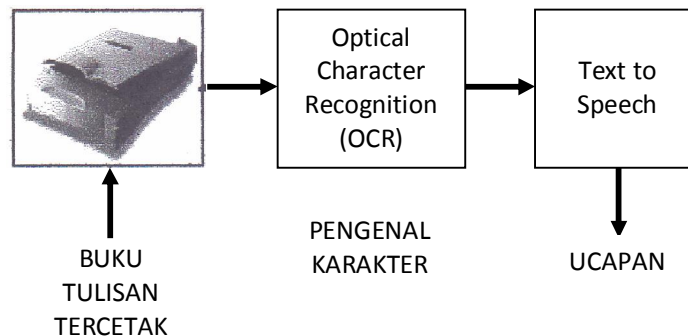
Pengolahan bahasa alami (natural language processing) : user dapat berkomunikasi dengan komputer menggunakan bahasa sehari-hari, misal bahasa Inggris, bahasa Indonesia, bahasa Jawa, dll, contoh

Text summarization : suatu sistem yang dapat membuat ringkasan hal-hal penting dari suatu wacana yang diberikan.

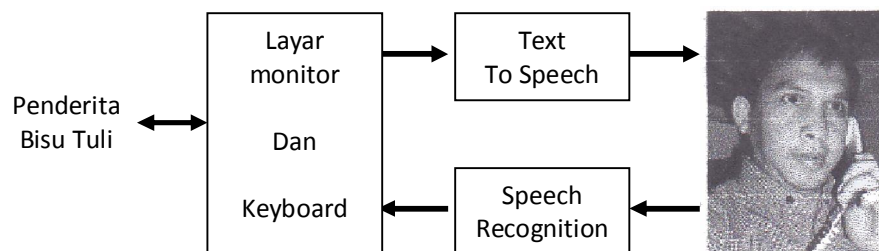
2. Pengenalan ucapan (speech recognition) : manusia dapat berkomunikasi dengan komputer menggunakan suara.

Contoh :

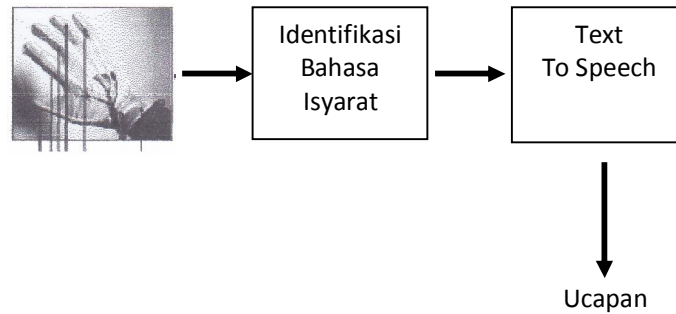
- Memberikan instruksi ke komputer dengan suara,
- Alat bantu membaca, untuk tunanetra, mempunyai masukan berupa teks tercetak (misalnya buku) dan mempunyai keluaran berupa ucapan dari teks tercetak yang diberikan.



- Telepon untuk penderita bisu-tuli

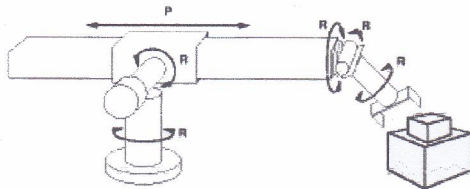


- Alat untuk tuna wicara




### 3. Robotika & sistem sensor

- Sistem sensor pada mesin cuci yaitu menggunakan sensor optik, mengeluarkan cahaya ke air dan mengukur bagaimana, cahaya, tersebut sampai ke ujung lainnya. Makin kotor, maka sinar yang sampai makin redup. Sistem juga mampu menentukan jenis kotoran tersebut daki/minyak. Sistem juga bisa menentukan putaran yang tepat secara, otomatis berdasarkan jenis dan banyaknya kotoran serta jumlah yang akan dicuci.
- Robotika



### 4. Computer vision : menginterpretasikan gambar atau objek-objek tampak melalui computer

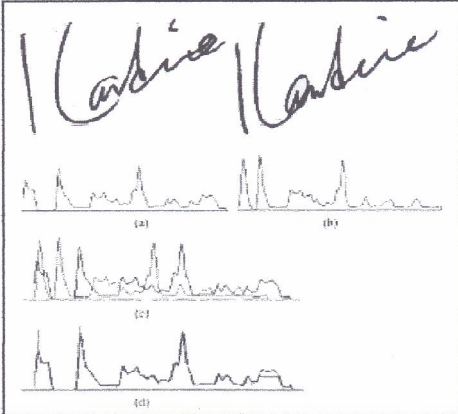


Designated Writing Area			Identity
Chinese:	English:	Numeral:	
		9	8
			No.1 d=24.02 s=0.650
Data from William Go			

Pengenalan angka

Pengenalan pola sidik jari seseorang

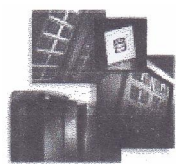
Total: 20 Done: 20 First: 20 Second: 0 Third: 0 Other: 0 Reject: 0  
0 1 2 3 4 5 6 7 8 9  
0 1 2 3 4 5 6 7 8 9



Pengenalan deteksi tanda tangan asli/palsu  
(tanda tangan yang dibuat oleh orang yang sama/berbeda)

5. Intelligent computer-aided instruction : komputer dapat digunakan sebagai tutor yang dapat melatih & mengajar  
Contoh : Learn to speak English

6. Game playing



1997, Deep Blue mengalahkan Garry Kasparov, the World Chess Champion

Deep Blue chess machine menggunakan komputer IBM, dibuat tabors 1990-an oleh Hsu, Campbell, Tan, Hoarse, Brody, Benjamin



Deep Blue mampu mengevaluasi 200 juta posisi bidak

**1.6 SOFT Computing**

Soft computing merupakan inovasi baru dalam membangun sistem cerdas yaitu sistem yang memiliki keahlian seperti manusia pada domain tertentu, mampu beradaptasi dan belajar agar dapat bekerja lebih baik jika terjadi perubahan lingkungan. Soft computing mengeksplorasi adanya toleransi terhadap ketidaktepatan, ketidakpastian, dan kebenaran parsial untuk dapat diselesaikan dan dikendalikan dengan mudah agar sesuai dengan reality (Prof Lotfi A Zadeh, 1992).

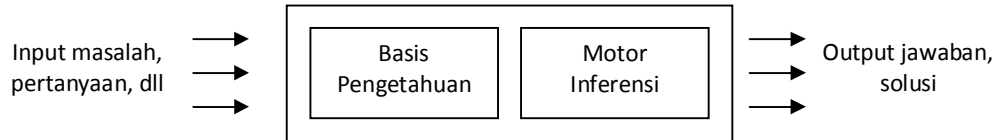
Metodologi-metodologi yang digunakan dalam Soft computing adalah :

1. Sistem Fuzzy (mengakomodasi ketidaktepatan) 4 Logika Fuzzy (fuzzy logic)
2. Jaringan Syaraf (menggunakan pembelajaran) 4 Jaringan Syaraf Tiruan (neural network)
3. Probabilistic Reasoning (mengakomodasi ketidakpastian)
4. Evolutionary Computing (optimasi) 4 Algoritma Genetika

## BAB II MASALAH DAN RUANG MASALAH

### 2.1 MASALAH DAN METODE PEMECAHAN MASALAH

Sistem yang menggunakan kecerdasan buatan akan memberikan output berupa solusi dari suatu masalah berdasarkan kumpulan pengetahuan yang ada.



Gambar 2.1 sistem yang menggunakan kecerdasan buatan

Pada gambar 2.1, input yang diberikan pada sistem yang menggunakan kecerdasan buatan adalah berupa masalah. Sistem harus dilengkapi dengan sekumpulan pengetahuan yang ada pada basis pengetahuan. Sistem harus memiliki motor inferensi agar mampu mengambil kesimpulan berdasarkan fakta atau pengetahuan. Output yang diberikan berupa solusi masalah sebagai hasil dari inferensi.

Secara umum, untuk membangun suatu sistem yang mampu menyelesaikan masalah, perlu dipertimbangkan 4 hal :

1. Mendefinisikan masalah dengan tepat.
2. Pendefinisian ini mencakup spesifikasi yang tepat mengenai keadaan awal dan solusi yang diharapkan.
3. Menganalisis masalah tersebut serta mencari beberapa teknik penyelesaian masalah yang sesuai.
4. Merepresentasikan pengetahuan yang perlu untuk menyelesaikan masalah tersebut.
5. Memilih teknik penyelesaian masalah yang terbaik

### 2.2 MENDEFINISIKAN MASALAH SEBAGAI SUATU RUANG KEADAAN

Misalkan permasalahan yang dihadapi adalah permainan catur, maka harus ditentukan :

1. Posisi awal pada papan catur  
Posisi awal setiap permainan catur selalu sama, yaitu semua bidak diletakkan di atas papan catur dalam 2 sisi, yaitu kubu putih dan kubu hitam.
2. Aturan – aturan untuk melakukan gerakan  
aturan – aturan ini sangat berguna untuk menentukan gerakan suatu bidak, yaitu melangkah dari satu keadaan ke keadaan lain. Misalkan untuk mempermudah menunjukkan posisi bidak, setiap kotak ditunjukkan dalam huruf (a,b,c,d,e,f,g,h) pada arah horisontal dan angka (1,2,3,4,5,6,7,8) pada arah vertikal. Suatu aturan untuk menggerakkan bidak dan posisi (e,2) ke (e,4) dapat ditunjukkan dengan aturan :



if bidak putih pada kotak (e,2),  
 and kotak (e,3) kosong,  
 and kotak (e,4) kosong  
 then gerakkan bidak dari (e,2) ke (e,4)

3. Tujuan (goal)

Tujuan yang ingin dicapai adalah posisi pada papan catur yang menunjukkan kemenangan seseorang terhadap lawannya. Kemenangan ini ditandai dengan posisi raja yang sudah tidak dapat bergerak lagi.

Contoh tersebut menunjukkan representasi masalah dalam Ruang Keadaan (State Space), yaitu suatu ruang yang berisi semua keadaan yang mungkin. Kita dapat memulai bermain catur dengan menempatkan diri pada keadaan awal, kemudian bergerak dari satu keadaan ke keadaan yang lain sesuai dengan aturan yang ada, dan mengakhiri permainan jika salah satu telah mencapai tujuan.

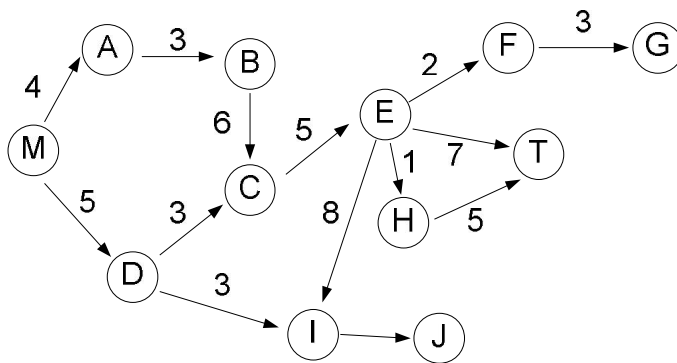
Jadi untuk mendeskripsikan masalah dengan baik harus :

1. Mendefinisikan suatu ruang keadaan (state space)
2. Menetapkan satu atau lebih keadaan awal (initial state)
3. Menetapkan satu atau lebih tujuan (goal state)
4. Menetapkan kumpulan aturan

Ada beberapa cara untuk merepresentasikan Ruang Keadaan, antara lain :

**GRAPH KEADAAN**

Graph terdiri dari node-node yang menunjukkan keadaan yaitu keadaan awal dan keadaan baru yang akan dicapai dengan menggunakan operator. Node-node dalam graph keadaan saling dihubungkan dengan menggunakan arc (busur) yang diberi panah untuk menunjukkan arah dan suatu keadaan ke keadaan berikutnya.



Gambar 2.2 Graph Keadaan

Graph keadaan dengan node M menunjukkan keadaan awal, node T adalah tujuan. Ada 4 lintasan dari M ke T :

- M-A-B-C-E-T
- M-A-B-C-E-H-T
- M-D-C-E-T

M-D-C-E-H-T

Lintasan buntu atau lintasan yang tidak sampai ke tujuan

M-A-B-C-E-F-G

M-A-B-C-E-I-J

M-D-C-E-F-G

M-D-C-E-I-J

M-D-I-J

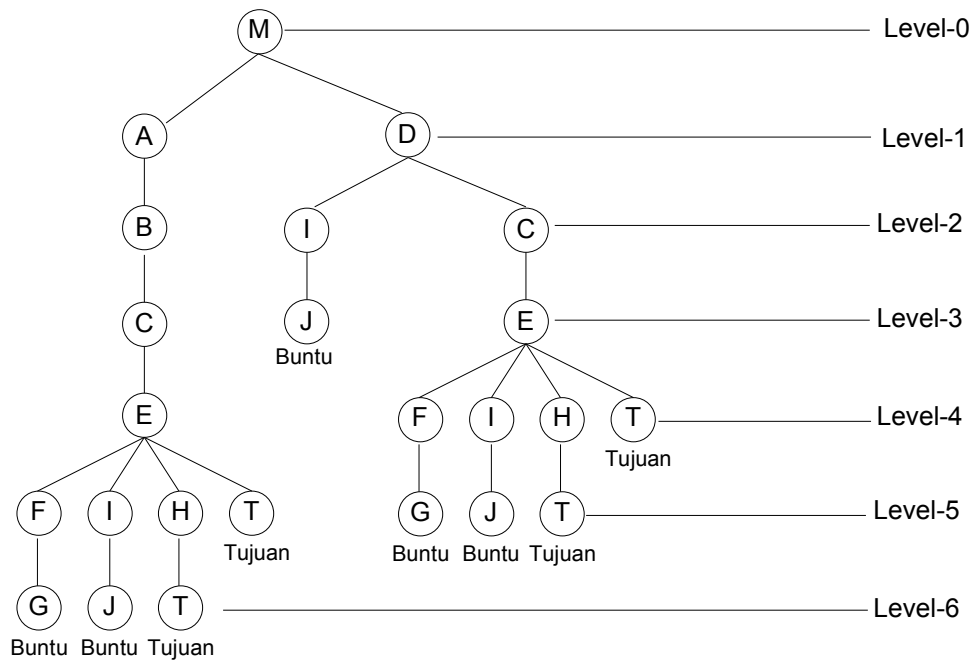
### POHON PELACAKAN / PENCARIAN

Struktur pohon digunakan untuk menggambarkan keadaan secara hirarkis. Node yang terletak pada level-0 disebut 'akar'.

Node akar : menunjukkan keadaan awal & memiliki beberapa percabangan yang terdiri atas beberapa node yang disebut 'anak'.

Node-node yang tidak memiliki anak disebut 'daun' menunjukkan akhir dari suatu pencarian, dapat berupa tujuan yang diharapkan (goal) atau jalan buntu (dead end).

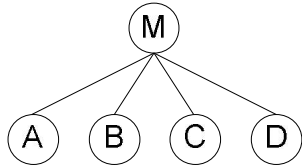
Gambar berikut menunjukkan pohon pencarian untuk graph keadaan dengan 6 level.



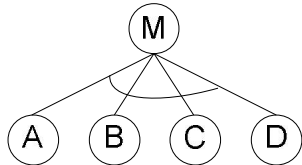
Gambar 2.3 Pohon Pelacakan

### POHON AND/OR

Masalah M dicari solusinya dengan 4 kemungkinan yaitu A OR B OR C OR D.



Masalah M hanya dapat diselesaikan dengan A AND B AND C AND D



Contoh dengan menggunakan pohon AND/OR tujuan yang dicapai pada pohon digambar sebelumnya bisa dipersingkat hanya sampai level-2 saja.

### Contoh 1 : Masalah EMBER

Ada 2 ember masing-masing berkapasitas 4 galon (ember A) dan 3 galon (ember B). Ada pompa, air yang akan digunakan untuk mengisi air pada ember tersebut. Bagaimana dapat mengisi tepat 2 galon air ke dalam ember berkapasitas 4 galon?

Penyelesaian :

1. Identifikasi ruang keadaan (state space)

Permasalahan ini dapat digambarkan sebagai himpunan pasangan bilangan bulat

$x$  = jumlah air yang diisikan ke ember 4 galon (ember A)

$y$  = jumlah air yang diisikan ke ember 3 galon (ember B)

Ruang keadaan =  $(x,y)$  sedemikian hingga,  $x \in (0,1,2,3,4)$  dan  $y \in (0,1,2,3)$

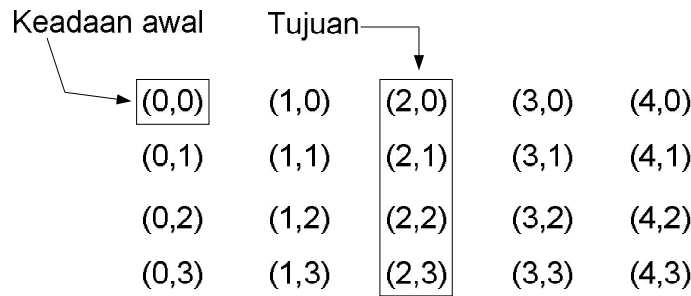
2. Keadaan awal & tujuan

Keadaan awal : kedua ember kosong =  $(0,0)$

Tujuan : ember 4 galon berisi 2 galon air =  $(2,n)$  dengan sembarang  $n$

3. Keadaan ember

Keadaan ember bisa digambarkan sebagai berikut :



4. Aturan-aturan

Diasumsikan kita dapat mengisi ember air itu dari pompa, air, membuang air dari ember ke luar, menuangkan air dari ember yang satu ke ember yang lain.

Kita buat beberapa aturan-aturan yang dapat digambarkan sebagai berikut

Aturan ke	Jika	Maka
1	$(x,y)$ $x < 4$	$(4,y)$ Isi ember A
2	$(x,y)$ $y < 3$	$(x,3)$ Isi ember B
3	$(x,y)$ $x > 0$	$(x - d,y)$ Tuang sebagian air keluar dari ember A
4	$(x,y)$ $y > 0$	$(x,y - d)$ Tuang sebagian air keluar dari ember B
5	$(x,y)$ $x > 0$	$(0,y)$ Kosongkan ember A dengan membuang airnya
6	$(x,y)$ $y > 0$	$(x,0)$ Kosongkan ember B dengan membuang airnya
7	$(x,y)$ $x+y \geq 4$ dan $y > 0$	$(4,y - (4 - x))$ Tuang air dari ember B ke ember A sampai ember A penuh
8	$(x,y)$ $x+y \geq 3$ dan $x > 0$	$(x - (3 - y),3)$ Tuang air dari ember A ke ember B sampai ember B penuh
9	$(x,y)$ $x+y \leq 4$ dan $y > 0$	$(x+y, 0)$ Tuang seluruh air dari ember B ke ember A
10	$(x,y)$ $x+y \leq 3$ dan $x > 0$	$(0,x+y)$ Tuang seluruh air dari ember A

		ke ember B
11	(0,2)	(2,0) Tuang 2 galon air dari ember B ke ember A

5. Representasi ruang keadaan dengan pohon pelacakan

Pencarian suatu solusi dapat dilukiskan dengan menggunakan pohon. Tiap-tiap node menunjukkan satu keadaan. Jalur dari parent ke child, menunjukkan 1 operasi. Tiap node memiliki node child yang menunjukkan keadaan yang dapat dicapai oleh parent.

Solusi yang ditemukan :

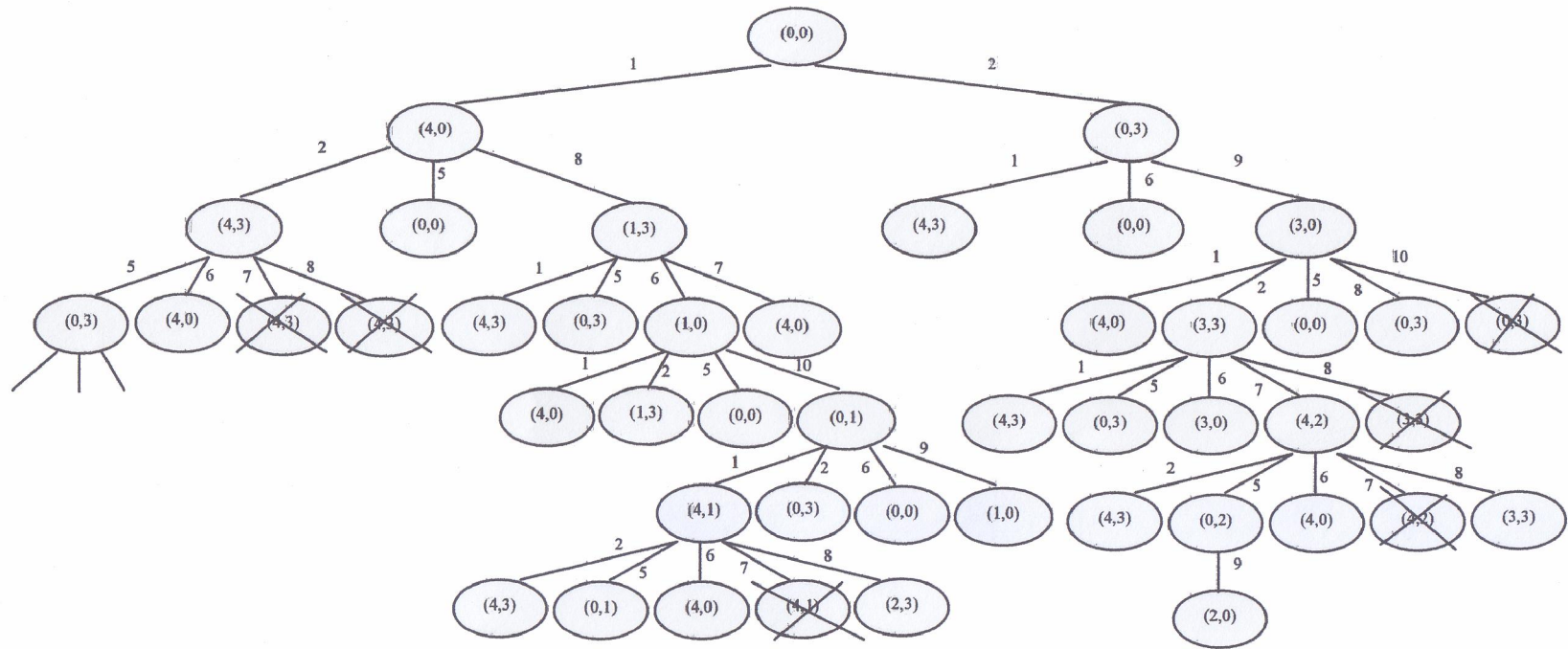
Solusi 1

Isi ember A	Isi ember B	Aturan yang dipakai
0	0	1
4	0	8
1	3	6
1	0	10
0	1	1
4	1	8
2	3	Solusi

Solusi 2

Isi ember A	Isi ember B	Aturan yang dipakai
0	0	2
0	3	9
3	0	2
3	3	7
4	2	5
0	2	9
2	0	Solusi

Representasi ruang keadaan untuk kasus EMBER





### BAB III METODE PELACAKAN/PENCARIAN

Hal penting dalam menentukan keberhasilan sistem cerdas adalah kesuksesan dalam pencarian. Pencarian = suatu proses mencari solusi dari suatu permasalahan melalui sekumpulan, kemungkinan ruang keadaan (state space). Ruang keadaan = merupakan suatu ruang yang berisi semua keadaan yang mungkin.

Untuk mengukur performansi metode pencarian, terdapat empat kriteria yang dapat digunakan :

- Completeness : apakah metode tersebut menjamin penemuan solusi jika solusinya memang ada?
- Time complexity berapa lama waktu yang diperlukan?
- Space complexity berapa banyak memori yang diperlukan
- Optimality : apakah metode tersebut menjamin menemukan solusi yang terbaik jika terdapat beberapa solusi berbeda?

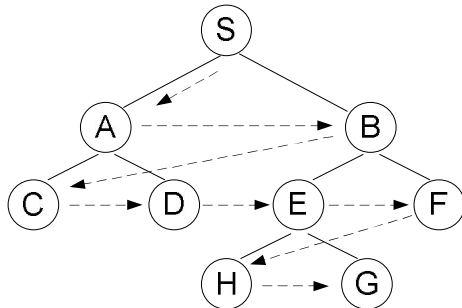
Teknik pencarian :

- A. Pencarian buta (blind search) : tidak ada informasi awal yang digunakan dalam proses pencarian
  1. Pencarian melebar pertama (Breadth – First Search)
  2. Pencarian mendalam pertama (Depth – First Search)
- B. Pencarian terbimbing (heuristic search) : adanya informasi awal yang digunakan dalam proses pencarian
  1. Pendakian Bukit (Hill Climbing)
  2. Pencarian Terbaik Pertama (Best First Search)

#### 3. 1 Pencarian Buta (blind search)

##### 1. Breadth – First Search

Semua node pada level n akan dikunjungi terlebih dahulu sebelum mengunjungi node-node pada level n+1. Pencarian dimulai dari node akar terus ke level l dari kiri ke kanan, kemudian berpindah ke level berikutnya dari kiri ke kanan hingga solusi ditemukan.





Keuntungan :

- tidak akan menemui jalan buntu, menjamin ditemukannya solusi (jika solusinya memang ada) dan solusi yang ditemukan pasti yang paling baik
- jika ada 1 solusi, maka breadth – first search akan ditemukannya jika ada lebih dari 1 solusi, maka solusi minimum akan ditemukan.
- Kesimpulan : complete dan optimal

Kelemahan :

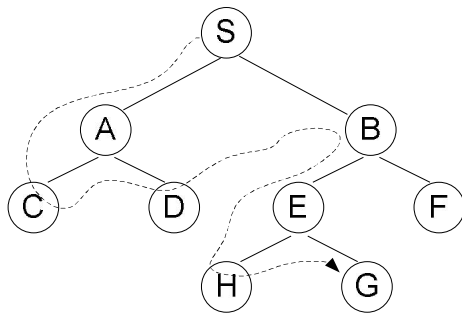
- membutuhkan memori yang banyak, karena harus menyimpan semua simpul yang pernah dibangkitkan. Hal ini harus dilakukan agar BFS dapat melakukan penelusuran simpul-simpul sampai di level bawah
- membutuhkan waktu yang cukup lama

## 2. Depth – First Search

Pencarian dilakukan pada suatu simpul dalam setiap level dan yang paling kiri.

Jika pada level yang paling dalam tidak ditemukan solusi, maka pencarian dilanjutkan pada simpul sebelah kanan dan simpul yang kiri dapat dihapus dari memori.

Jika pada level yang paling dalam tidak ditemukan solusi, maka pencarian dilanjutkan pada level sebelumnya. Demikian seterusnya sampai ditemukan solusi.



Keuntungan :

- Membutuhkan memori relatif kecil, karena hanya node-node pada lintasan yang aktif saja yang disimpan
- Secara kebetulan, akan menemukan solusi tanpa harus menguji lebih banyak lagi dalam ruang keadaan, jadi jika solusi yang dicari berada pada level yang dalam dan paling kiri, maka DFS akan ditemukannya dengan cepat → waktu cepat

Kelemahan:

- Memungkinkan tidak ditemukannya tujuan yang diharapkan, karena jika pohon yang dibangkitkan mempunyai level yang sangat dalam (tak

- terhingga) → tidak complete karena tidak ada jaminan menemukan solusi
- Hanya mendapat 1 solusi pada setiap pencarian, karena jika terdapat lebih dari satu solusi yang sama tetapi berada pada level yang berbeda, maka DFS tidak menjamin untuk menemukan solusi yang paling baik → tidak optimal.

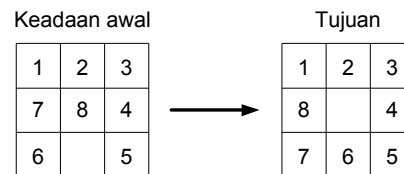
### 3.2 Heuristic Search

Pencarian buta tidak selalu dapat diterapkan dengan baik, hal ini disebabkan waktu aksesnya yang cukup lama dan besarnya memori yang diperlukan. Untuk masalah dengan ruang masalah yang besar, teknik pencarian buta bukan metode yang baik karena keterbatasan kecepatan computer dan memori.

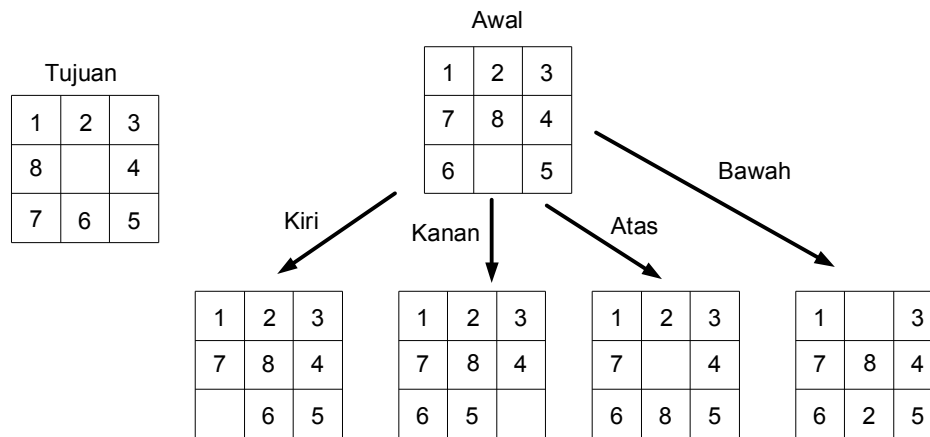
Metode heuristic search diharapkan bisa menyelesaikan permasalahan yang besar. Metode heuristic search menggunakan suatu fungsi yang menghitung biaya perkiraan (estimasi) dari suatu simpul tertentu menuju ke simpul tujuan → disebut fungsi heuristic.

Aplikasi yang menggunakan fungsi heuristic : Google, Deep blue chess Machine  
 Misal kasus 8-puzzle. Ada 4 operator yang dapat digunakan untuk menggerakkan dari satu keadaan ke keadaan yang baru.

1. Ubin kosong digeser ke kiri
2. Ubin kosong digeser ke kanan
3. Ubin kosong digeser ke bawah
4. Ubin kosong digeser ke atas

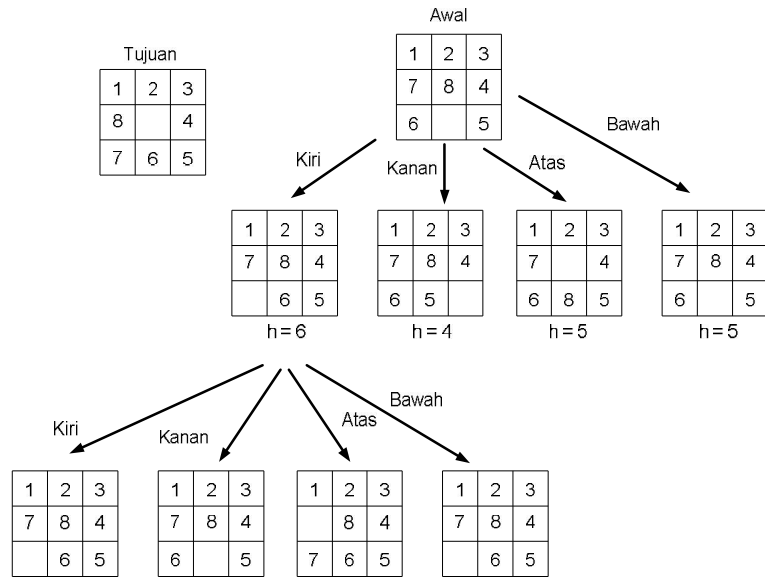


Langkah awal

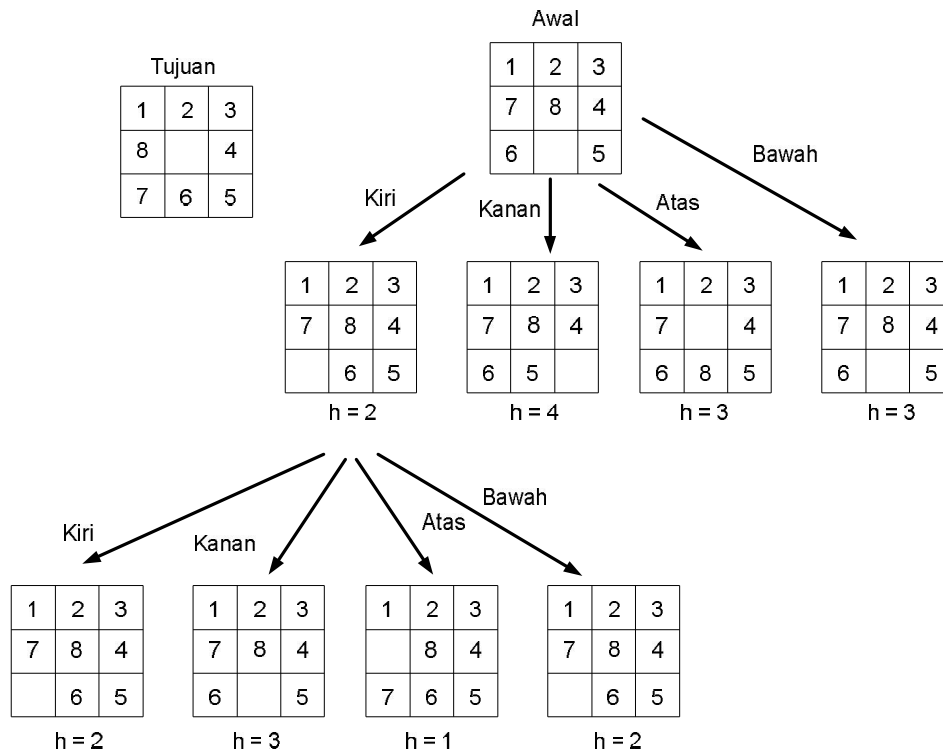


Pada pencarian heuristic perlu diberikan informasi, yaitu :

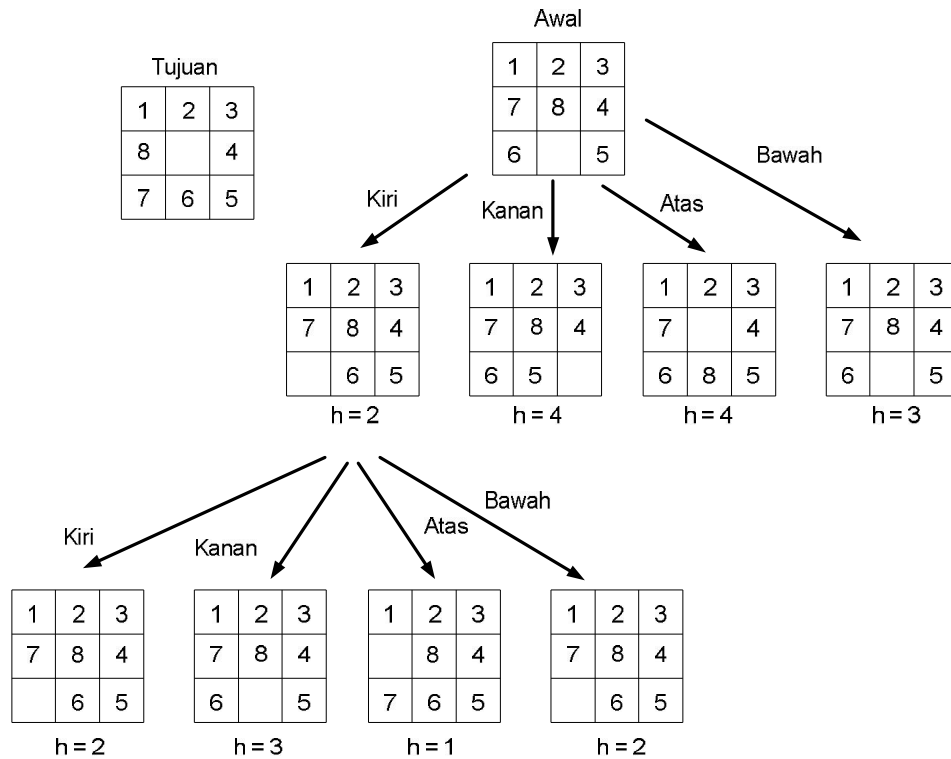
- Untuk jumlah ubin yang menempati posisi yang benar
- Jumlah yang lebih tinggi adalah yang lebih diharapkan (lebih baik)



- Untuk jumlah ubin yang menempati posisi yang salah  
Jumlah yang lebih kecil adalah yang diharapkan (lebih baik)



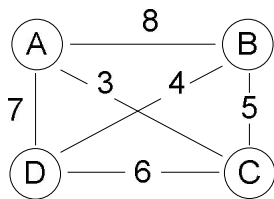
- Menghitung total gerakan yang diperlukan untuk mencapai tujuan  
Jumlah yang lebih kecil adalah yang diharapkan (lebih baik)



### 1. Hill Climbing

Contoh : Traveling Salesman Problem (TSP)

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



Solusi – solusi yang mungkin dengan menyusun kata-kata dalam urutan abjad, misal :

A – B – C – D : dengan panjang lintasan (=19)

A – B – D – C : (-18)

A – C – B – D : (-12)

A – C – D – B : (-13)

Dst

#### a. Metode simple hill climbing

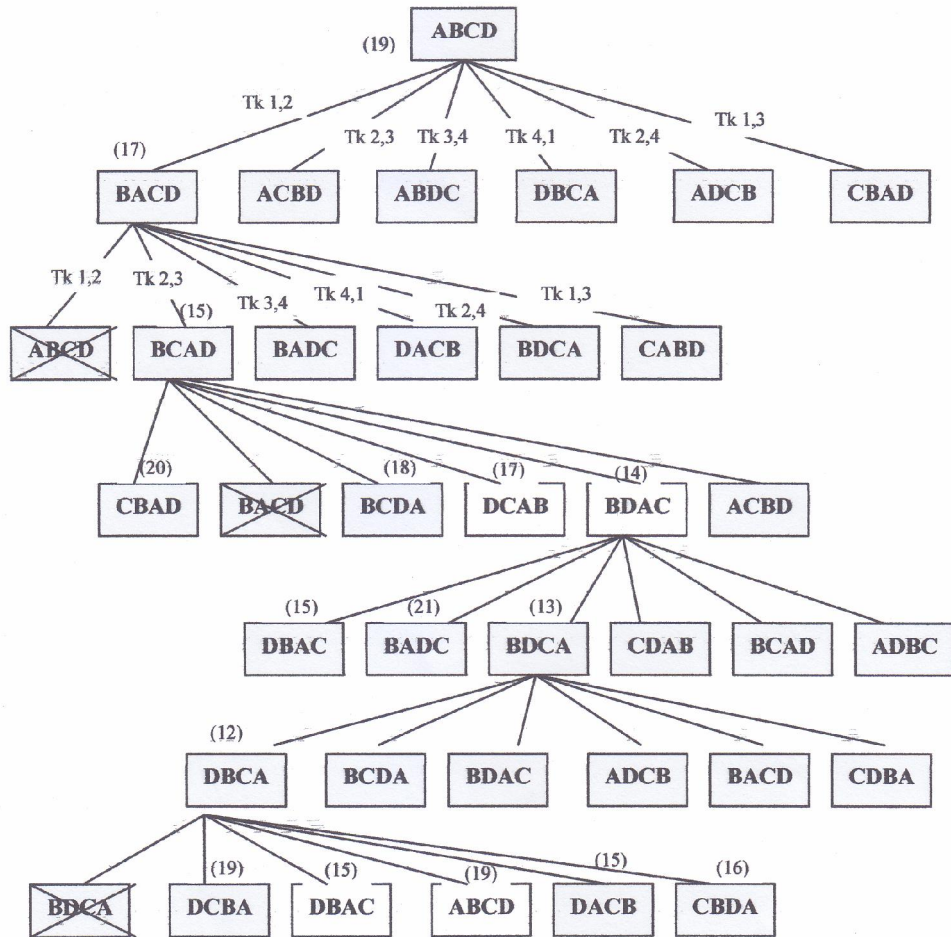
Ruang keadaan berisi semua kemungkinan lintasan yang mungkin. Operator digunakan untuk menukar posisi kota-kota yang bersebelahan. Fungsi heuristik yang digunakan adalah panjang lintasan yang terjadi.

Operator yang akan digunakan adalah menukar urutan posisi 2 kota dalam 1 lintasan. Bila ada  $n$  kota, dan ingin mencari kombinasi lintasan dengan menukar posisi urutan 2 kota, maka akan didapat sebanyak :

$$\frac{n!}{2!(n-2)!} = \frac{4!}{2!(4-2)!} = 6 \text{ kombinasi}$$

Keenam kombinasi ini akan dipakai semuanya sebagai operator, yaitu :

Tukar 1,2 = menukar urutan posisi kota ke -1 dengan kota ke -2  
Tukar 2,3 = menukar urutan posisi kota ke -2 dengan kota ke -3  
Tukar 3,4 = menukar urutan posisi kota ke -3 dengan kota ke -4  
Tukar 4,1 = menukar urutan posisi kota ke -4 dengan kota ke -1  
Tukar 2,4 = menukar urutan posisi kota ke -2 dengan kota ke -4  
Tukar 1,3 = menukar urutan posisi kota ke -1 dengan kota ke -3



Keadaan awal, lintasan ABCD (=19).

Level pertama, hill climbing mengunjungi BACD (=17), BACD (=17) < ABCD (=19), sehingga BACD menjadi pilihan selanjutnya dengan operator Tukar 1,2

Level kedua, mengunjungi ABCD, karena operator Tukar 1,2 sudah dipakai BACD, maka pilih node lain yaitu BCAD (=15), BCAD (=15) < BACD (=17)

Level ketiga, mengunjungi CBAD (=20), CBAD (=20) > BCAD (=15), maka pilih node lain yaitu BCDA (=18), pilih node lain yaitu DCAB (=17), pilih node lain yaitu BDAC (=14), BDAC (=14) < BCAD (=15)

Level keempat mengunjungi DBAC (=15), DBAC (=15) > BDAC (=14), maka pilih node lain yaitu BADC (=21), pilih node lain yaitu BDCA (=13), BDCA (=13) < BDAC (=14)

Level kelima, mengunjungi DBCA (=12), DBCA (=12) < BDCA (=13)

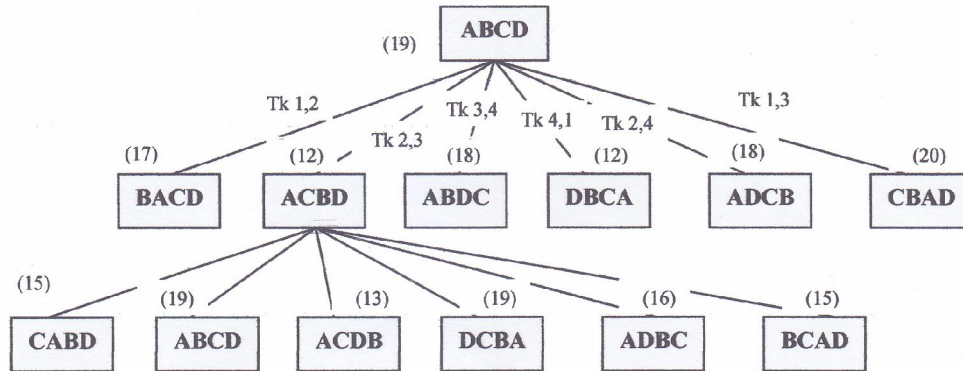
Level keenam, mengunjungi BDCA, karena operator Tukar 1,2 sudah dipakai DBCA, maka pilih node lain yaitu DCBA, pilih DBAC, pilih ABCD, pilih DACB, pilih CBDA

Karena sudah tidak ada node yang memiliki nilai heuristik yang lebih kecil

dibanding nilai heuristik DBCA, maka node DBCA (=12) adalah lintasan terpendek (SOLUSI)

b. Metode steepest – ascent hill climbing

Steepest – ascent hill climbing hampir sama dengan simple – ascent hill climbing, hanya saja gerakan pencarian tidak dimulai dari kiri, tetapi berdasarkan nilai heuristik terbaik.



Keadaan awal, lintasan ABCD (=19).

Level pertama, hill climbing memilih nilai heuristik terbaik yaitu ACBD (=12) sehingga ACBD menjadi pilihan selanjutnya.

Level kedua, hill climbing memilih nilai heuristik terbaik, karena nilai heuristik lebih besar dibanding ACBD, maka hasil yang diperoleh lintasannya tetap ACBD (=12)

## BAB IV REPRESENTASI PENGETAHUAN

Dua bagian dasar sistem kecerdasan buatan (menurut Turban)

- Basis pengetahuan :  
Berisi fakta tentang objek-objek dalam domain yang dipilih dan hubungan diantara domain-domain tersebut
- Inference Engine :  
Merupakan sekumpulan prosedur yang digunakan untuk menguji basis pengetahuan dalam menjawab suatu pertanyaan, menyelesaikan masalah, atau membuat keputusan

Basis pengetahuan berisi struktur data yang dapat dimanipulasi oleh suatu sistem inferensi yang menggunakan pencairan dan teknik pencocokan pola pada basis pengetahuan yang bermanfaat untuk menjawab pertanyaan, menggambarkan kesimpulan atau bentuk lainnya sebagai suatu fungsi kecerdasan.

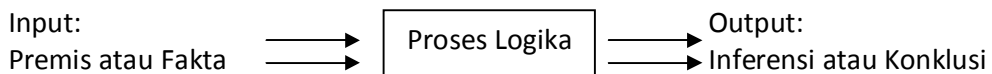
Karakteristik representasi pengetahuan

1. Dapat diprogram dengan bahasa komputer dan disimpan dalam memori
2. Fakta dan pengetahuan lain yang terkandung didalamnya dapat digunakan untuk melakukan penalaran

Dalam menyelesaikan masalah harus dibutuhkan pengetahuan yang cukup dan sistem juga harus memiliki kemampuan untuk menalar. Basis pengetahuan dan kemampuan untuk melakukan penalaran merupakan bagian terpenting dari sistem yang menggunakan kecerdasan buatan.

### 4.1 LOGIKA

Logika adalah bentuk representasi pengetahuan yang paling tua. Proses logika adalah proses membentuk kesimpulan atau menarik suatu inferensi berdasarkan fakta yang telah ada. Input dan proses logika berupa premis atau fakta-fakta yang diakui kebenarannya sehingga dengan melakukan penalaran pada proses logika dapat dibentuk suatu inferensi atau kesimpulan yang benar juga.



Ada 2 penalaran yang dapat dilakukan untuk mendapat konklusi :

1. Penalaran deduktif : dimulai dari prinsip umum untuk mendapatkan konklusi yang lebih khusus. Contoh :  
Premis mayor      : Jika hujan turun saya tidak akan berangkat kuliah  
Premis minor      : Hari ini hujan turun  
Konklusi            : Hari ini saya tidak akan berangkat kuliah
2. Penalaran induktif : dimulai dari fakta-fakta khusus untuk mendapatkan kesimpulan umum. Contoh :



Premis -1 : Aljabar adalah pelajaran yang sulit  
 Premis -2 : Geometri adalah pelajaran yang sulit  
 Premis -3 : Kalkulus adalah pelajaran yang sulit  
 Konklusi : Matematika adalah pelajaran yang sulit  
 Munculnya premis baru bisa mengakibatkan gugurnya konklusi yang sudah diperoleh, misal Premis -4 : Kinematika adalah pelajaran yang sulit  
 Premis tersebut menyebabkan konklusi : "Matematika adalah pelajaran yang sulit", menjadi salah, karena Kinematika bukan merupakan bagian dari Matematika, sehingga bila menggunakan penalaran induktif sangat dimungkinkan adanya ketidakpastian.

#### 4.1.1 Logika Proposisi

Proposisi adalah suatu pernyataan yang dapat bernilai Benar atau Salah. Simbol-simbol seperti P dan Q menunjukkan proposisi. Dua atau lebih proposisi dapat digabungkan dengan menggunakan operator logika :

- Konjungsi : A (and)
- Disjungsi :  $\vee$  (or)
- Negasi :  $\neg$ , (not)
- Implikasi :  $\rightarrow$  (if then)
- Ekuivalensi :  $\leftrightarrow$  (if and only if)

Not

P	Not P
B	S
S	B

And, Or, If - Then, If - and - only - if

P	Q	P and Q	P or Q	if P then Q	P if and only if Q
B	B	B	B	B	B
B	S	S	B	S	S
S	B	S	B	B	S
S	S	S	S	B	B

Untuk melakukan inferensi pada logika proposisi dapat dilakukan dengan menggunakan resolusi. Resolusi adalah suatu aturan untuk melakukan inferensi yang dapat berjalan secara efisien dalam suatu bentuk khusus yaitu conjunctive normal form (CNF), ciri – cirinya :

- setiap kalimat merupakan disjungsi literal
- semua kalimat terkonjungsi secara implicit

Langkah-langkah untuk mengubah suatu kalimat (konversi) ke bentuk CNF

- hilangkan implikasi dan ekuivalensi
  - $x \rightarrow y$  menjadi  $\neg x \vee y$
  - $x \leftrightarrow y$  menjadi  $(\neg x \vee y) \wedge (\neg y \vee x)$
- Kurangi lingkup semua negasi menjadi satu negasi saja
  - $\neg(\neg x)$  menjadi  $x$
  - $\neg(x \vee y)$  menjadi  $(\neg x \wedge \neg y)$
  - $\neg(x \wedge y)$  menjadi  $(\neg x \vee \neg y)$
- Gunakan aturan assosiatif dan distributif untuk mengkonversi menjadi conjunction of disjunction

- Asosiatif :  $(A \vee B) \vee C$  menjadi  $A \vee (B \vee C)$
- Distributif :  $(A \wedge B) \vee C$  menjadi  $(A \vee C) \wedge (B \vee C)$

- Buat satu kalimat terpisah untuk tiap-tiap konjungsi

Contoh :

Diketahui basis pengetahuan (fakta-fakta yang bernilai benar) sebagai berikut :

1. P
2.  $(P \wedge Q) \rightarrow R$
3.  $(S \vee T) \rightarrow Q$
4. T

Tentukan kebenaran R.

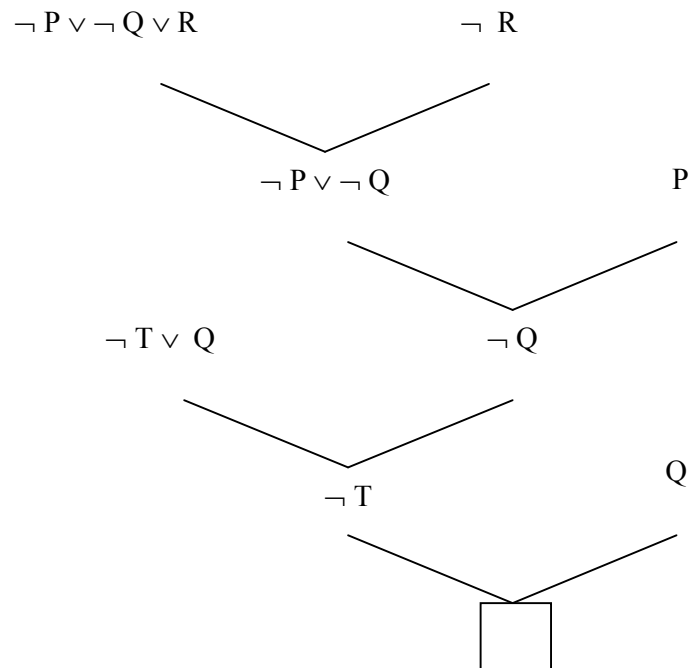
Untuk membuktikan kebenaran R dengan menggunakan resolusi, maka ubah dulu menjadi bentuk CNF.

Kalimat	Langkah-langkah	CNF
1. P	Sudah merupakan bentuk CNF	P
2. $(P \wedge Q) \rightarrow R$	<ul style="list-style-type: none"> <li>▪ Menghilangkan implikasi <math>\neg (P \vee \neg Q) \vee R</math></li> <li>▪ Mengurangi lingkup negasi <math>(\neg P \vee \neg Q) \vee R</math></li> <li>▪ Gunakan asosiatif <math>\neg P \vee \neg Q \vee R</math></li> </ul>	$\neg P \vee \neg Q \vee R$
3. $(S \vee T) \rightarrow Q$	<ul style="list-style-type: none"> <li>▪ Menghilangkan implikasi : <math>\neg (S \vee T) \vee Q</math></li> <li>▪ Mengurangi lingkup negasi <math>(\neg S \vee \neg T) \vee Q</math></li> <li>▪ Gunakan distributive <math>(\neg S \vee Q) \wedge (\neg T \vee Q)</math></li> </ul>	$(\neg S \vee Q)$ $(\neg T \vee Q)$
4, T	Sudah merupakan bentuk CNF	T

Kemudian kita tambahkan kontradiksi pada tujuannya, R menjadi  $\neg R$  sehingga fakta-fakta (dalam bentuk CNF) dapat disusun menjadi :

1. P
2.  $\neg P \vee \neg Q \vee R$
3.  $\neg S \vee Q$
4.  $\neg T \vee Q$
5. T
6.  $\neg R$

Sehingga resolusi dapat dilakukan untuk membuktikan kebenaran R, sebagai berikut :



Contoh bila diterapkan dalam kalimat

- P : Ani anak yang cerdas
- Q : Ani rajin belajar
- R : Ani akan menjadi juara kelas
- S : Ani makannya banyak
- T : Ani istirahatnya cukup

Kalimat yang terbentuk :

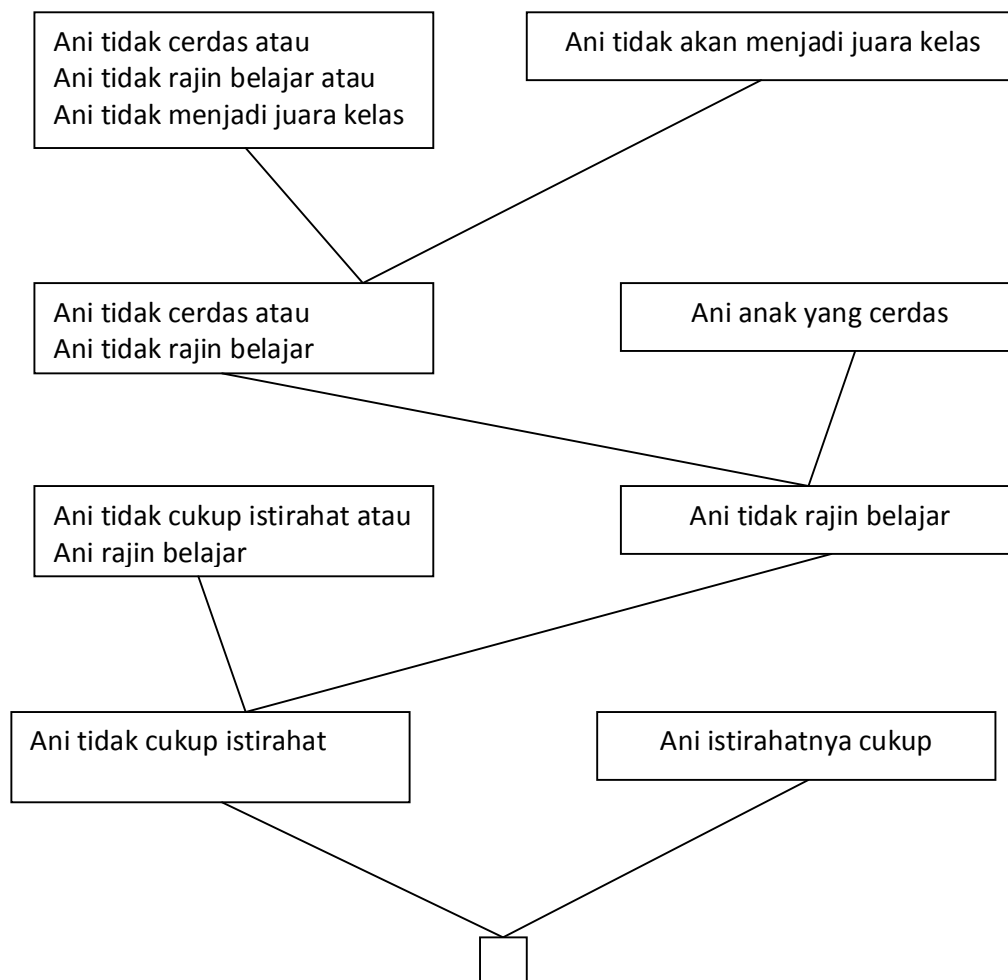
- Ani anak yang cerdas
- Jika ani anak yang cerdas dan ani rajin belajar, maka ani akan menjadi juara kelas
- Jika ani makannya banyak atau ani istirahatnya cukup, maka ani rajin belajar
- Ani istirahatnya cukup

Setelah dilakukan konversi ke bentuk CNF, didapat :

Fakta ke-2 : Ani tidak cerdas atau ani tidak rajin belajar atau ani akan menjadi juara kelas

Fakta ke-3 : Ani tidak makan banyak atau ani rajin belajar

Fakta ke-4 : Ani tidak cukup istirahat atau ani rajin belajar



#### 4.1.2 Logika Predikat

Representasi Fakta Sederhana

Misal diketahui fakta-fakta sebagai berikut :

Andi adalah seorang laki-laki : A

Ali adalah seorang laki-laki : B

Amir adalah seorang laki-laki : C

Anto adalah seorang laki-laki : D

Agus adalah seorang laki-laki : E

Jika kelima fakta tersebut dinyatakan dengan menggunakan proposisi, maka akan terjadi pemborosan, dimana beberapa pernyataan dengan predikat yang sama akan dibuat dalam proposisi yang berbeda.

Logika predikat digunakan untuk merepresentasikan hal-hal yang tidak dapat direpresentasikan dengan menggunakan logika proposisi. Pada logika predikat

kita dapat merepresentasikan fakta-fakta sebagai suatu pernyataan yang disebut dengan wff (well - formed formula). Logika predikat merupakan dasar bagi bahasa AI seperti bahasa pemrograman PROLOG

Pada contoh diatas, dapat dituliskan

laki-laki (x)

dimana x adalah variabel yang disubstitusikan dengan Andi, Ali, Amir, Anto, Agus, dan laki-laki yang lain.

Dalam logika predikat, suatu proposisi atau premis dibagi menjadi 2 bagian, yaitu argumen (objek) dan predikat (keterangan). Argumen adalah individu atau objek yang membuat keterangan. Predikat adalah keterangan yang membuat keterangan dan predikat.

Contoh :

1. Jika besok tidak hujan, Tommy pergi ke gunung  
 $\neg \text{cuaca}(\text{hujan, besok}) \rightarrow \text{pergi}(\text{tommy, gunung})$
2. Diana adalah nenek dari ibu Amir  
Nenek (Diana, ibu (Amir))
3. Mahasiswa berada di dalam, kelas  
didalam (mahasiswa, kelas)

Dari contoh diatas dapat dijabarkan sebagai berikut

di dalam = predikat (keterangan)

mahasiswa = argumen (objek)

kelas = argumen (objek)

4. Johan suka Maria  
suka (johan, maria)
5. Pintu terbuka  
Buka (pintu)
6. Johan suka Maria  
Ramon suka Maria  
Misal : Johan = x, Maria = y, Ramon = z  
Maka :  $\text{suka}(x, y) \wedge \text{suka}(z, y) \rightarrow \text{tidak suka}(x, z)$   
Dibaca: Jika Johan suka Maria dan Ramon suka Maria, maka Johan tidak suka Ramon

Misal terdapat pernyataan sebagai berikut :

1. Andi adalah seorang mahasiswa
2. Andi masuk jurusan Elektro
3. Setiap mahasiswa elektro pasti mahasiswa teknik
4. Kalkulus adalah matakuliah yang sulit
5. Setiap mahasiswa teknik pasti akan suka kalkulus atau akan membencinya
6. Setiap mahasiswa pasti akan suka terhadap suatu matakuliah
7. Mahasiswa yang tidak pernah hadir pada kuliah matakuliah sulit, maka mereka pasti tidak suka terhadap matakuliah tersebut.
8. Andi tidak pernah hadir kuliah matakuliah kalkulus

Kedelapan pernyataan diatas dapat dibawa ke bentuk logika predikat dengan menggunakan operator-operator :  $\rightarrow$ ,  $\neg$ ,  $\wedge$ ,  $\vee$ ,  $\forall$  (untuk setiap),  $\exists$  (terdapat), sebagai berikut :

1. mahasiswa(Andi)
2. elektro (Andi)
3.  $\forall x : \text{elektro}(x) \rightarrow \text{teknik}(x)$
4. sulit (kalkulus)
5.  $\forall x : \text{teknik}(x) \rightarrow \text{suka}(x, \text{kalkulus}) \vee \text{benci}(x, \text{kalkulus})$
6.  $\forall x : \exists y : \text{suka}(x, y)$
7.  $\forall x : \forall y : \text{mahasiswa}(x) \wedge \text{sulit}(y) \wedge \neg \text{hadir}(x, y) \rightarrow \neg \text{suka}(x, y)$
8.  $\neg \text{hadir}(\text{Andi}, \text{kalkulus})$

Andaikan kita akan menjawab pertanyaan :

“Apakah Andi suka matakuliah kalkulus?”

Maka dari pernyataan ke-7 kita akan membuktikan bahwa Andi tidak suka dengan matakuliah kalkulus. Dengan menggunakan penalaran backward, bisa dibuktikan bahwa :

$\neg \text{suka}(\text{Andi}, \text{kalkulus})$

Sebagai berikut :

$\neg \text{suka}(\text{Andi}, \text{kalkulus})$

↑  
(7, substitusi)

Mahasiswa (Andi)  $\wedge$  sulit (kalkulus)  $\wedge$   $\neg$  hadir (Andi, kalkulus)

↑  
(1)

Sulit (kalkulus)  $\wedge$   $\neg$  hadir (Andi, kalkulus)

↑  
(4)

$\neg$  hadir (Andi, kalkulus)

↑  
(8)



Dari penalaran tersebut dapat dibuktikan bahwa Andi tidak suka dengan matakuliah kalkulus.

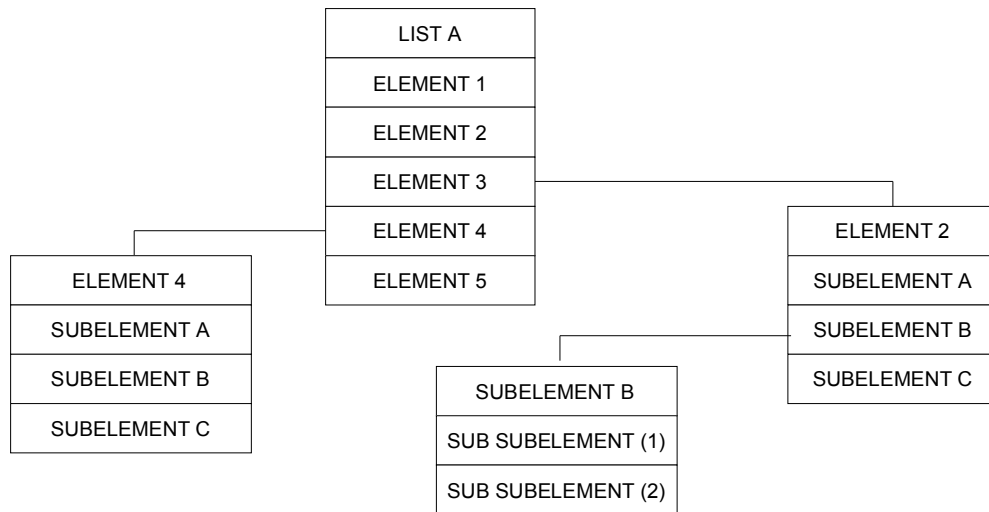
#### 4.2 LIST dan TREE

List dan Tree merupakan struktur sederhana yang digunakan dalam representasi hirarki pengetahuan.

## LIST

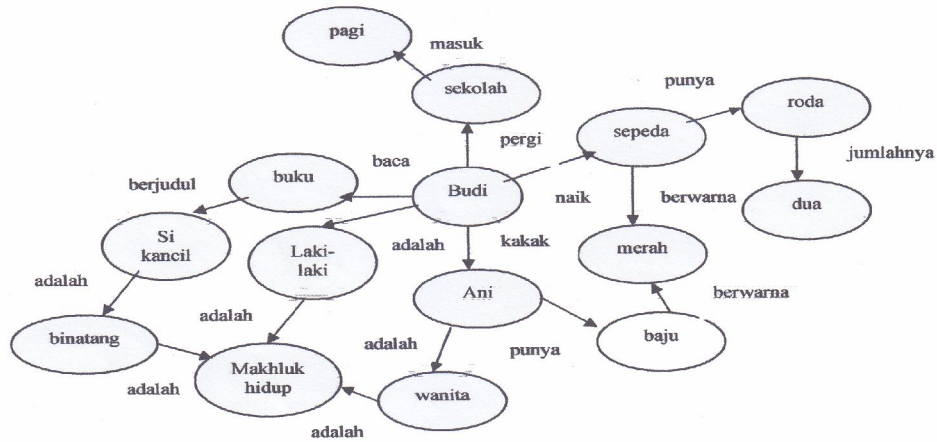
Adalah daftar dari rangkaian materi yang terkait Hal ini bisa merupakan suatu daftar (list) Hama orang yang anda kenal, barang-barang yang akan dibeli dari toko Serba, Ada, hal-hal yang akan dikerjakan minggu ini, atau produk-produk berbagai jenis barang dalam katalog, dll.

List biasanya digunakan untuk merepresentasikan hirarki pengetahuan dimana objek dikelompokkan, dikategorikan atau digabungkan sesuai dengan urutan atau hubungannya. Objek dibagi dalam kelompok atau jenis yang sama. Kemudian hubungan ditampilkan dengan menghubungkan satu sama lain.



## 4.3 JARINGAN SEMANTIK

Jaringan semantik merupakan gambaran pengetahuan grafis yang menunjukkan hubungan antar berbagai objek. Jaringan semantik terdiri dari lingkaran-lingkaran yang menunjukkan objek dan informasi tentang objek-objek tersebut. Objek disini bisa berupa benda, atau peristiwa. Antara, 2 objek dihubungkan oleh arc yang menunjukkan hubungan antar objek. Gambar berikut menunjukkan representasi pengetahuan menggunakan jaringan semantik.



#### 4.4 RAME

Frame merupakan kumpulan pengetahuan tentang suatu objek tertentu, peristiwa, lokasi, situasi, dll. Frame memiliki slot yang menggambarkan rincian (atribut) dan karakteristik objek.

Frame biasanya digunakan untuk merepresentasikan pengetahuan yang didasarkan pada karakteristik yang sudah dikenal, yang merupakan pengalaman-pengalaman.

Dengan menggunakan frame, sangat mudah untuk membuat inferensi tentang objek, peristiwa, atau situasi baru, karena frame menyediakan basis pengetahuan yang ditarik dari pengalaman.

Frame Mobil
Class : Transportasi
Nama pabrik : Audi
Negara : Jerman
Model : 5000
Tipe : Sedan
Bobot : 3300 lb
Ukuran dasur roda : 105,8 inchi
Jumlah Pintu: 4 (default)
Transmisi : 3-speed otomatis
Jumlah roda : 4 (default)
Mesin : (referensi kerangka mesin)
<ul style="list-style-type: none"> <li>• Tipe : in-line, overhead cam</li> <li>• Jumlah silinder : 5</li> </ul>
Akselerasi
0 – 60 : 40,4 detik
¼ mil : 17,1 detik, 85 mph
Jarak gas : rata-rata 22 mpg

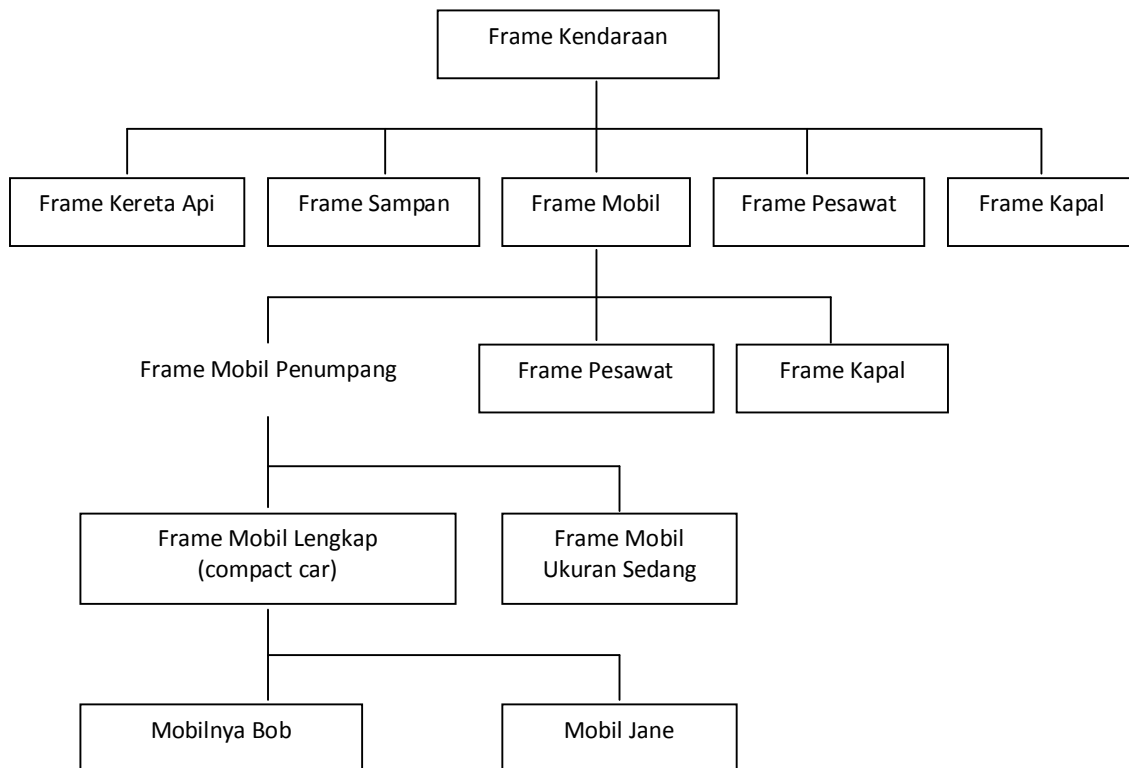
Frame silinder
Kaliber silinder : 3,19 inci



Tak silinder : 3,4 inci
Rasio kompresi : 7,8 : 1
Sistem babas bakar : injeksi dengan pertukaran turbo
Tenaga : 140 HP
Torsi : 160/ft/LB

### HIRARKI FRAME

Kebanyakan sistem AI menggunakan kumpulan frame yang saling terkait satu dengan lainnya bersama-sama. Gambar di bawah ini menunjukkan hirarki frame kendaraan, terdiri dari 5 frame yaitu frame kereta api, frame sampan, frame mobil, frame pesawat, frame kapal. Masing-masing frame masih dapat dipecah lagi menjadi beberapa frame yang rinci, misal frame mobil terdiri dari frame penumpang mobil, frame truk, frame bis.



Susunan hirarki dari frame memungkinkan pewarisan frame. Akar dari tree terletak di puncak, dimana level tertinggi dari abstraksi disajikan. Frame pada bagian dasar (bawah) disebut daun dari tree.

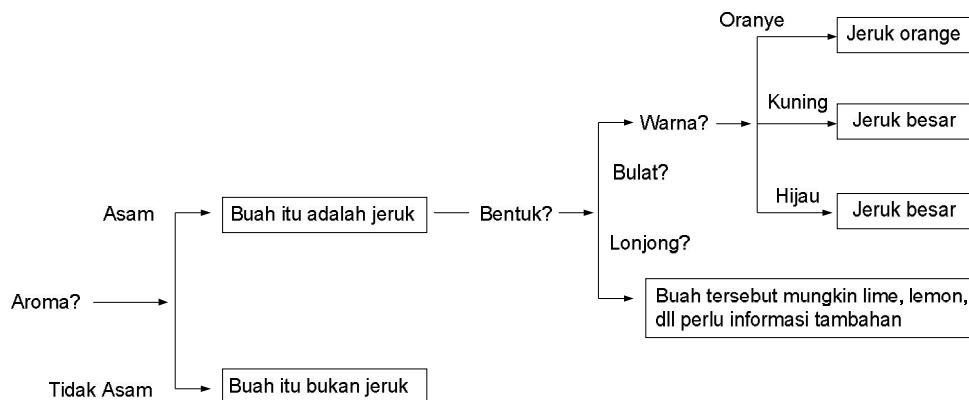
Hirarki memungkinkan pewarisan sifat-sifat. Setiap frame biasanya mewarisi sifat-sifat dari frame dengan level yang lebih tinggi. Pewarisan merupakan mekanisme untuk membentuk pengetahuan, yang menyediakan nilai slot, dari frame ke frame.

Didalam hirarki diatas, masing-masing frame dirinci hubungannya seperti hubungan antara frame orangtua (parent frame) dan anak (child frame).

Parent Frame	Nama : Compact Car
Slot	: Facets
Pemilik	: Cek daftar registrasi
Warna	: Daftar per manufaktur
No. Silinder	Daftar range : Semua manufaktur Jika dibutuhkan : Tanya pemilik
Model (tahun) :	Range : 1950 – 2001 Jika dibutuhkan : Tanya pemilik
Parent Frame	Nama : Mobilnya Jane
Slot	: Facets
Pemilik	: Jane
Warna	: Biru
No. Silinder	: 6
Buatan (tahun)	: 1992

#### 4.5. POHON KEPUTUSAN (DECISION TREE)

Keuntungan utama representasi pengetahuan dengan pohon keputusan adalah dapat menyederhanakan proses akuisisi pengetahuan dan dapat dengan mudah dikonversikan ke bentuk aturan (rule)



#### 4.6. NASKAH (SCRIPT)

Script adalah skema representasi pengetahuan yang sama dengan frame, yaitu merepresentasikan pengetahuan berdasarkan karakteristik yang sudah dikenal sebagai pengalaman-pengalaman. Perbedaannya, frame menggambarkan objek, sedangkan script menggambarkan urutan peristiwa.

Dalam menggambarkan urutan peristiwa, script menggunakan slot yang berisi informasi tentang orang, objek, dan tindakan-tindakan yang terjadi dalam suatu

peristiwa. Elemen script meliputi :

1. Kondisi input, yaitu kondisi yang harus dipenuhi sebelum terjadi atau berlaku suatu peristiwa dalam script
2. Track, yaitu variasi yang mungkin terjadi dalam suatu script
3. Prop, berisi objek-objek pendukung yang digunakan selama peristiwa terjadi
4. Role, yaitu pesan yang dimainkan oleh seseorang dalam peristiwa
5. Scene, yaitu adegan yang dimainkan yang menjadi bagian dari suatu peristiwa
6. Hasil, yaitu kondisi yang ada setelah urutan peristiwa dalam script terjadi.

Berikut ini adalah contoh script kejadian yang ada di “Ujian Akhir Semester”,

Jalur (track) : ujian tertulis matakuliah Kecerdasan Buatan  
Role (peran) : mahasiswa, pengawas  
Prop (pendukung) : lembar soal, lembar jawab, presensi, pena, dll  
Kondisi input : mahasiswa terdaftar untuk mengikuti ujian

Adegan (scene) -1 : Persiapan pengawas

- Pengawas menyiapkan lembar soal
- Pengawas menyiapkan lembar jawab
- Pengawas menyiapkan lembar presensi

Adegan-2 : Mahasiswa masuk ruangan

- Pengawas mempersilahkan mahasiswa masuk
- Pengawas membagikan lembar soal
- Pengawas membagikan lembar jawab
- Pengawas memimpin doa

Adegan - 3 : Mahasiswa mengerjakan soal ujian

- Mahasiswa menuliskan identitas di lembar jawab
- Mahasiswa menandatangani lembar jawab
- Mahasiswa mengerjakan soal
- Mahasiswa mengecek jawaban

Adegan - 4 : Mahasiswa telah selesai ujian

- Pengawas mempersilahkan mahasiswa keluar ruangan
- Mahasiswa mengumpulkan kembali lembar jawab
- Mahasiswa keluar ruangan.

Adegan - 5 : Mahasiswa mengemasi lembar jawab

- Pengawas mengurutkan lembar jawab
- Pengawas mengecek lembar jawab dan presensi
- Pengawas meninggalkan ruangan

Hasil :

- Mahasiswa merasa senang dan lega
- Mahasiswa merasa kecewa
- Mahasiswa pusing
- Mahasiswa memaki – maki
- Mahasiswa sangat bersyukur

#### **4.7 SISTEM PRODUKSI (ATURAN PRODUKSI/PRODUCTION RULES)**

Representasi pengetahuan dengan sistem produksi berupa aplikasi aturan (rule) yang berupa :

1. Antecedent, yaitu bagian yang mengekspresikan situasi atau premis (pernyataan berawalan IF)
2. Konsekuen, yaitu bagian yang menyatakan suatu tindakan tertentu atau konklusi yang diterapkan jika suatu situasi atau premis bernilai benar (pernyataan berawalan THEN)

Konsekuensi atau konklusi yang dinyatakan pada bagian THEN baru dinyatakan benar, jika bagian IF pada system tersebut juga benar dan sesuai dengan aturan tertentu.

Contoh:

IF lalulintas pagi ini padat  
THEN saya naik sepeda motor saja

Aturan dapat ditulis dalam beberapa bentuk :

1. IF premis THEN kesimpulan  
Jika pendapatan tinggi MAKA pajak yang harus dibayar juga tinggi
2. Kesimpulan IF premis  
Pajak yang harus dibayar tinggi JIKA pendapatan tinggi
3. Inclusion of ELSE  
IF pendapatan tinggi OR pengeluaran tinggi, THEN pajak yang harus dibayar tinggi ELSE pajak yang harus dibayar rendah
4. Aturan yang lebih kompleks  
IF rating kredit tinggi AND gaji lebih besar dari \$30,000 OR aset lebih dari \$75,000 AND sejarah pembayaran tidak miskin THEN pinjaman diatas \$ 10,000 disetujui dan daftar pinjaman masuk kategori "B"

Apabila pengetahuan direpresentasikan dengan aturan, maka ada 2 metode penalaran yang dapat digunakan :

1. Forward Reasoning (penalaran maju)  
Pelacakan dimulai dari keadaan awal (informasi atau fakta yang ada) dan kemudian, dicoba untuk mencocokkan dengan tujuan yang diharapkan
2. Backward Reasoning (penalaran mundur)  
Penalaran dimulai dari tujuan atau hipotesa, baru dicocokkan, dengan keadaan awal atau fakta-fakta yang ada.

Ada beberapa faktor yang mempengaruhi pemilihan backward atau forward dalam memilih metode penalaran :

- banyaknya keadaan awal dan tujuan. Jika jumlah keadaan awal lebih kecil daripada tujuan, maka digunakan penalaran forward. Sebaliknya jika jumlah tujuan lebih banyak daripada keadaan awal, maka dipilih penalaran backward
- rata-rata jumlah node yang dapat diraih langsung dari suatu node. Lebih baik dipilih yang jumlah node tiap cabangnya lebih sedikit
- apakah program butuh menanyai user untuk melakukan justifikasi terhadap proses penalaran? Jika ya, maka alangkah baiknya jika dipilih arah yang lebih memudahkan user
- bentuk kejadian yang akan memicu penyelesaian masalah. Jika kejadian itu berupa fakta baru, maka lebih baik dipilih penalaran forward. Namun jika kejadian itu berupa query, maka lebih baik digunakan penalaran backward.



## **BAB V PENALARAN**

### **5.1 KETIDAKPASTIAN**

Dalam kenyataan sehari-hari banyak masalah didunia ini tidak dapat dimodelkan secara lengkap dan konsisten. Suatu penalaran dimana adanya penambahan fakta baru mengakibatkan ketidakkonsistenan, dengan ciri-ciri penalaran sebagai berikut :

- adanya ketidakpastian
- adanya perubahan pada pengetahuan
- adanya penambahan fakta baru dapat mengubah konklusi yang sudah terbentuk

contoh :

Premis -1 : Aljabar adalah pelajaran yang sulit

Premis -2 : Geometri adalah pelajaran yang sulit

Premis -3 : Kalkulus adalah pelajaran yang sulit

Konklusi : Matematika adalah pelajaran yang sulit

Munculnya premis baru bisa mengakibatkan gugurnya konklusi yang sudah diperoleh, misal

Premis -4 : Kinematika adalah pelajaran yang sulit

Premis tersebut menyebabkan konklusi : "Matematika adalah pelajaran yang sulit, menjadi salah karena Kinematika bukan merupakan bagian dari Matematika, sehingga bila menggunakan penalaran induktif sangat dimungkinkan adanya ketidakpastian.

Untuk mengatasi ketidakpastian maka digunakan penalaran statistik.

### **5.2 PROBABILITAS & TEOREMA BAYES**

#### **PROBABILITAS**

Probabilitas menunjukkan kemungkinan sesuatu akan terjadi atau tidak.

$$p(x) = \frac{\text{Jumlah kejadian berhasil}}{\text{Jumlah semua kejadian}}$$

Misal dari 10 orang sarjana, 3 orang menguasai cisco, sehingga peluang untuk memilih sarjana yang menguasai cisco adalah :

$$p(\text{cisco}) = 3/10 = 0.3$$

## TEOREMA BAYES

$$p(H_i|E) = \frac{p(E|H_i) * (p(H_i))}{\sum_{k=1}^n p(E|H_k) * (p(H_k))}$$

dengan:

- $p(H_i|E)$  = probabilitas hipotesis  $H_i$ , benar jika diberikan evidence (fakta) E
- $p(E|H_i)$  = probabilitas munculnya evidence (fakta) E jika diketahui hipotesis  $H_i$  benar
- $p(H_i)$  = probabilitas hipotesis  $H_i$ , (menurut hasil sebelumnya) tanpa memandang evidence (fakta) apapun
- n = jumlah hipotesis yang mungkin

Contoh :

Asih mengalami gejala ada bintik-bintik di wajahnya. Dokter menduga bahwa Asih terkena cacar dengan :

- probabilitas munculnya bintik-bintik di wajah, jika Asih terkena cacar →  $p(\text{bintik} | \text{cacar}) = 0.8$
- probabilitas Asih terkena cacar tanpa memandang gejala apapun →  $p(\text{cacar}) = 0.4$
- probabilitas munculnya bintik-bintik di wajah, jika Asih terkena alergi →  $p(\text{bintik} | \text{alergi}) = 0.3$
- probabilitas Asih terkena alergi tanpa memandang gejala apapun →  $p(\text{alergi}) = 0.7$
- probabilitas munculnya bintik-bintik di wajah, jika Asih jerawat →  $p(\text{bintik} | \text{jerawat}) = 0.9$
- probabilitas Asih jerawat tanpa memandang gejala apapun →  $p(\text{jerawat}) = 0.5$

Matra :

- probabilitas Asih terkena cacar karena ada bintik-bintik di wajahnya :

$$p(H_i|E) = \frac{p(E|H_i) * (p(H_i))}{\sum_{k=1}^n p(E|H_k) * (p(H_k))}$$

$$p(\text{cacar} | \text{bintik}) = \frac{p(\text{bintik} | \text{cacar}) * p(\text{cacar})}{p(\text{bintik} | \text{cacar}) * p(\text{cacar}) + p(\text{bintik} | \text{alergi}) * p(\text{alergi}) + p(\text{bintik} | \text{jerawat}) * p(\text{jerawat})}$$

$$p(\text{cacar} | \text{bintik}) = \frac{(0.8) * (0.4)}{(0.8) * (0.4) + (0.3) * (0.7) + (0.9) * (0.5)} = \frac{0.32}{0.98} = 0.327$$



- probabilitas Asih terkena alergi karena ada bintik-bintik di wajahnya :

$$p(\text{alergi}|\text{bintik}) = \frac{p(\text{bintik}|\text{alergi}) * p(\text{alergi})}{p(\text{bintik}|\text{cacar}) * p(\text{cacar}) + p(\text{bintik}|\text{alergi}) * p(\text{alergi}) + p(\text{bintik}|\text{jerawat}) * p(\text{jerawat})}$$

$$p(\text{alergi}|\text{bintik}) = \frac{(0.3) * (0.7)}{(0.8) * (0.4) + (0.3) * (0.7) + (0.9) * (0.5)} = \frac{0.21}{0.98} = 0.214$$

- probabilitas Asih jerawat karena ada bintik-bintik di wajahnya :

$$p(\text{jerawat}|\text{bintik}) = \frac{p(\text{bintik}|\text{jerawat}) * p(\text{jerawat})}{p(\text{bintik}|\text{cacar}) * p(\text{cacar}) + p(\text{bintik}|\text{alergi}) * p(\text{alergi}) + p(\text{bintik}|\text{jerawat}) * p(\text{jerawat})}$$

$$p(\text{alergi}|\text{bintik}) = \frac{(0.9) * (0.5)}{(0.8) * (0.4) + (0.3) * (0.7) + (0.9) * (0.5)} = \frac{0.45}{0.98} = 0.459$$

Jika setelah dilakukan pengujian terhadap hipotesis muncul satu atau lebih evidence (fakta) atau observasi baru maka :

$$p(H|E, e) = \frac{p(H|E) * p(e|E, H)}{p(e|E)}$$

dengan:

e = evidence lama

E = evidence atau observasi baru

$p(H|E, e)$  = probabilitas hipotesis H benar jika muncul evidence baru E dari evidence lama e

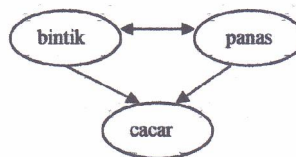
$p(H|E)$  = probabilitas hipotesis H benar jika diberikan evidence E

$p(e|E, H)$  = kaitan antara e dan E jika hipotesis H benar

$p(e|E)$  = kaitan antara e dan E tanpa memandang hipotesis apapun

Misal:

Adanya bintik-bintik di wajah merupakan gejala seseorang terkena cacar. Observasi baru menunjukkan bahwa selain bintik-bintik di wajah, panas badan juga merupakan gejala orang kena cacar. Jadi antara munculnya bintik-bintik di wajah dan panas badan juga memiliki keterkaitan satu sama lain.



Asih ada bintik-bintik di wajahnya. Dokter menduga bahwa Asih terkena cacar dengan probabilitas terkena cacar bila ada bintik-bintik di wajah →  $p(\text{cacar}|\text{bintik}) = 0.8$

Ada observasi bahwa orang terkena cacar pasti mengalami panas badan. Jika

diketahui probabilitas orang terkena cacar bila panas badan  $\rightarrow p(\text{cacar} \mid \text{panas}) = 0.5$

Keterkaitan antara adanya bintik-bintik di wajah dan panas badan bila seseorang terkena cacar  $\rightarrow p(\text{bintik} \mid \text{panas, cacar}) = 0,4$

Keterkaitan antara adanya bintik-bintik di wajah dan panas badan  $\rightarrow p(\text{bintik} \mid \text{panas}) = 0,6$

Maka :

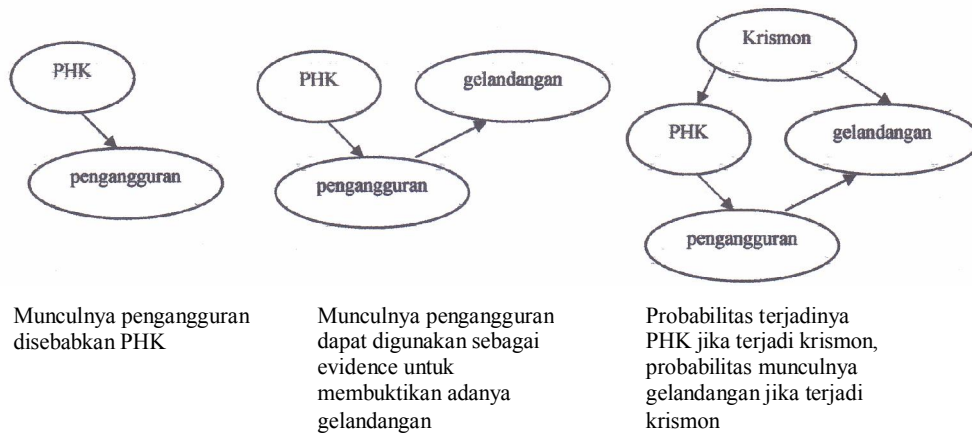
$$p(H|E, e) = \frac{p(H|E) * p(e|E, H)}{p(e|E)}$$

$$p(\text{cacar} \mid \text{panas, bintik}) = p(\text{cacar} \mid \text{panas}) \frac{p(\text{bintik} \mid \text{panas, cacar})}{p(\text{bintik} \mid \text{panas})}$$

$$p(\text{cacar} \mid \text{panas, bintik}) = (0.5) * \frac{(0.4)}{(0.6)} = 0.33$$

Pengembangan lebih jauh dari Teorema Bayes adalah Jaringan Bayes.

Contoh : hubungan antara krismon, PHK, pengangguran, gelandangan dalam suatu jaringan



Probabilitas untuk jaringan bayes

Atribut	Prob	Keterangan
$p(\text{pengangguran} \mid \text{PHK, gelandangan})$	0.95	Keterkaitan antara pengangguran & PHK, jika muncul gelandangan
$p(\text{pengangguran} \mid \text{PHK, } \sim \text{gelandangan})$	0.20	Keterkaitan antara pengangguran & PHK jika tidak ada gelandangan
$p(\text{pengangguran} \mid \sim \text{PHK, gelandangan})$	0.75	Keterkaitan antara pengangguran & tidak ada PHK,

		jika muncul gelandangan
p(pengangguran   ~PHK, ~gelandangan)	0.40	Keterkaitan antara pengangguran & tidak ada PHK, jika tidak ada gelandangan
p(PHK krismon)	0,50	Probabilitas orang di PHK jika terjadi krismon
p(PHK   ~krismon)	0.10	Probabilitas orang diPHK jika tidak terjadi krismon
p(pengangguran   krismon)	0.90	Probabilitas muncul pengangguran jika terjadi krismon
p(pengangguran   ~krismon)	0.30	Probabilitas muncul pengangguran jika tidak terjadi krismon
p(krismon)	0.80	

### 5.3 FAKTOR KEPASTIAN (CERTAINTY FACTOR)

Certainty Factor (CF) menunjukkan ukuran kepastian terhadap suatu fakta atau aturan.

$$CF[h,e] = MB[h,e] - MD[h,e]$$

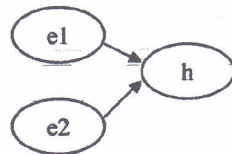
CF[h,e] = faktor kepastian

MB[h,e] = ukuran kepercayaan/tingkat keyakinan terhadap hipotesis h, jika diberikan/dipengaruhi evidence e (antara 0 dan 1)

MD[h,e] = ukuran ketidakpercayaan / tingkat ketidakyakinan terhadap hipotesis h, jika diberikan / dipengaruhi evidence e (antara 0 dan 1)

3 hal yang mungkin terjadi :

1. Beberapa evidence dikombinasikan untuk menentukan CF dari suatu hipotesis. Jika e1 dan e2 adalah observasi, maka :



$$MB[h, e1 \wedge e2] = \begin{cases} 0 & \text{Jika } MD[h, e1 \wedge e2] = 1 \\ MB[h, e1] + MB[h, e2] * (1 - MB[h, e1]) & \text{lainnya} \end{cases}$$

$$MD[h, e1 \wedge e2] = \begin{cases} 0 & \text{Jika } MB[h, e1 \wedge e2] = 1 \\ MD[h, e1] + MD[h, e2] * (1 - MD[h, e1]) & \text{lainnya} \end{cases}$$

Contoh :

Misal suatu observasi memberikan kepercayaan terhadap h dengan  $MB[h,e1]=0,3$  dan  $MD[h1, e] = 0.2$  maka :

$$CF[H,e1] = 0,3 - 0 = 0,3$$

Jika ada observasi baru dengan  $MB[H,e2] = 0,2$  dan  $MD[h,e2] = 0$ , maka

$$MB[h,e1 \wedge e2] = 0,3 + 0,2 (1- 0,3) = 0,44$$

$$MD[h,e1 \wedge e2] = 0$$

$$CF[h,e1 \wedge e2] = 0,44 - 0 = 0,44$$

Asih menderita bintik-bintik di wajahnya. Dokter memperkirakan Asih terkena cacar dengan kepercayaan  $MB[cacar,bintik] = 0,80$  dan  $MD [cacar,bintik] = 0,01$  maka:

$$CF[cacar,bintik] = 0,80 - 0,01=0,79$$

Jika ada observasi baru bahwa Asih juga panas badan dengan kepercayaan,

$MD[cacar,panas]=0,7$  dan  $MD[cacar,panas] = 0,08$  maka :

$$MB[cacar,bintik \wedge panas] = 0,8 + 0,7 * (1- 0,8)=0,94$$

$$MD[cacar,bintik \wedge panas] = 0,01 + 0,08 * (1- 0,01) = 0,0892$$

$$CF[cacar,bintik \wedge panas] = 0,94 - 0,0892 = 0,8508$$

## 2. CF dihitung dari kombinasi beberapa hipotesis

Jika  $h1$  dan  $h2$  adalah hipotesis maka :



$$MB[h1 \wedge h2,e] = \min (MB[h1,e], MB[h2,e])$$

$$MD[h1 \wedge h2,e] = \max (MB[h1,e], MB[h2,e])$$

$$MD[h1 \wedge h2,e] = \min (MD[h1,e], MD[h2,e])$$

$$MD[h1 \wedge h2,e] = \max (MD[h1,e], MD[h2,e])$$

Contoh :

Misal suatu observasi memberikan kepercayaan terhadap  $h1$  dengan

$MB[h1,e]=0,5$  dan  $MD[h1,e] = 0,2$  maka :

$$CF[h1,e] = 0,5 - 0,2 = 0,3$$

Jika observasi tersebut juga memberikan kepercayaan terhadap  $h2$  dengan

$MB[h2,e]=0,8$  dan  $MD[h2,e]=0,1$ , maka:

$$CF[h2,e] = 0,8 - 0,1= 0,7$$

Untuk mencari  $CF[h1 \wedge h2,e]$  diperoleh dari

$$MB[h1 \wedge h2,e] = \min (0,5 ; 0,8) = 0,5$$

$$MD[h1 \wedge h2,e] = \min (0,2 ; 0,1) = 0,1$$

$$CF[h1 \wedge h2,e] = 0,5 - 0,1 = 0,4$$

Untuk mencari  $CF[h1 \vee h2,e]$  diperoleh dari

$$MB[h1 \vee h2,e] = \max (0,5 ; 0,8) = 0,8$$

$$MD[h1 \vee h2,e] = \max (0,2 ; 0,1) = 0,2$$

$$CF[h1 \vee h2,e] = 0,8 - 0,2 = 0,6$$

Asih menderita bintik-bintik di wajahnya. Dokter memperkirakan Asih terkena cacar dengan kepercayaan  $MB[\text{cacar}, \text{bintik}] = 0,80$  dan  $MD[\text{cacar}, \text{bintik}] = 0,01$  maka

$$CF[\text{cacar}, \text{bintik}] = 0,80 - 0,01 = 0,79$$

Jika observasi tersebut juga memberikan kepercayaan bahwa Asih mungkin juga terkena alergi dengan kepercayaan  $MB[\text{alergi}, \text{bintik}] = 0,4$  dan  $MD[\text{alergi}, \text{bintik}] = 0,3$  maka

$$CF[\text{alergi}, \text{bintik}] = 0,4 - 0,3 = 0,1$$

Untuk mencari  $CF[\text{cacar} \wedge \text{alergi}, \text{bintik}]$  diperoleh dari

$$MB[\text{cacar} \wedge \text{alergi}, \text{bintik}] = \min(0,8; 0,4) = 0,4$$

$$MD[\text{cacar} \wedge \text{alergi}, \text{bintik}] = \min(0,01; 0,3) = 0,01$$

$$CF[\text{cacar} \wedge \text{alergi}, \text{bintik}] = 0,4 - 0,01 = 0,39$$

Untuk mencari  $CF[\text{cacar} \vee \text{alergi}, \text{bintik}]$  diperoleh dari

$$MB[\text{cacar} \vee \text{alergi}, \text{bintik}] = \max(0,8; 0,4) = 0,8$$

$$MD[\text{cacar} \vee \text{alergi}, \text{bintik}] = \max(0,01; 0,3) = 0,3$$

$$CF[\text{cacar} \vee \text{alergi}, \text{bintik}] = 0,8 - 0,3 = 0,5$$

Kesimpulan : semula faktor kepercayaan bahwa Asih terkena cacar dan gejala munculnya bintik-bintik di wajahnya adalah 0,79. demikian pula faktor kepercayaan bahwa Ani terkena alergi dari gejala munculnya bintik-bintik di wajah adalah 0,1. Dengan adanya gejala yang sama mempengaruhi 2 hipotesis yang berbeda ini memberikan faktor kepercayaan :

$$\text{Asih menderita cacar dan alergi} = 0,39$$

$$\text{Asih menderita cacar atau alergi} = 0,5$$

Pertengahan tahun 2002, ada indikasi bahwa turunnya devisa Indonesia disebabkan oleh permasalahan TKI di Malaysia. Apabila, diketahui  $MB[\text{devisaturun}, \text{TKI}] = 0,8$  dan  $MD[\text{devisaturun}, \text{TKI}] = 0,3$  maka  $CF[\text{devisaturun}, \text{TKI}]$  :

$$CF[\text{devisaturun}, \text{TKI}] = MB[\text{devisaturun}, \text{TKI}] - MD[\text{devisaturun}, \text{TKI}] \\ 0,8 - 0,3 = 0,5$$

## **BAB VI**

### **SISTEM PAKAR**

#### **6.1 Pendahuluan**

Sistem pakar (expert system) adalah sistem yang berusaha mengapdosikan pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para ahli.

Jadi sistem pakar adalah kepakaran ditransfer dari seorang pakar (atau sumber kepakaran yang lain) ke komputer, pengetahuan yang ada disimpan dalam komputer, dan pengguna dapat berkonsultasi pada komputer itu untuk suatu nasehat, lalu komputer dapat mengambil inferensi (menyimpulkan, mendeduksi, dll) seperti layaknya seorang pakar, kemudian menjelaskannya, ke pengguna tersebut, bila perlu dengan alasan-alasannya.

Sistem Pakar terkadang lebih baik unjuk kerjanya daripada seorang pakar manusia. Dengan sistem pakar, orang awam pun dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli. Bagi para ahli, sistem pakar juga akan membantu aktivitasnya sebagai asisten yang sangat berpengalaman. Sistem pakar dikembangkan pertama kali tahun 1960.

Sistem pakar yang terkenal : MYCIN yaitu Sistem pakar medical yang bisa mendiagnosa penyakit infeksi dan merekomendasi pengobatan; DENDRAL yaitu Mengidentifikasi struktur molokular campuran kimia yang tak dikenal; XCON merupakan sistem pakar untuk membantu konfigurasi sistem komputer besar, membantu melayani order langganan sistem komputer DEC VAX 11/780 ke dalam sistem spesifikasi final yang lengkap; XSEL; PROSPECTOR dan lain sebagainya

#### **6.2 Keuntungan dan Kelemahan Sistem Pakar**

KEUNTUNGAN SISTEM PAKAR :

1. Memungkinkan orang swam bisa, mengeijakan peker aan para ahli
2. Bisa melakukan proses secara berulang secara otomatis
3. Menyimpan pengetahuan dan keahlian para pakar
4. Mampu mengambil dan melestarikan keahlian para pakar (terutama yang termasuk keahlian langka)
5. Mampu beroperasi dalam lingkungan yang berbahaya,
6. Memiliki kemampuan untuk bekerja dengan informasi yang tidak lengkap dan mengandung ketidakpastian. Pengguna bisa merespon dengan jawaban 'tidak tahu' atau 'tidak yakin' pada satu atau lebih pertanyaan selama konsultasi dan sistem pakar tetap akan memberikan jawaban.
7. Tidak memerlukan biaya, saat tidak digunakan, sedangkan pada pakar manusia memerlukan biaya sehari-hari.

8. Dapat digandakan (diperbanyak) sesuai kebutuhan dengan waktu yang minimal dan sedikit biaya,
9. Dapat memecahkan masalah lebih cepat daripada kemampuan manusia dengan catatan menggunakan data yang sama.
10. Menghemat waktu dalam pengambilan keputusan
11. Meningkatkan kualitas dan produktivitas karena dapat memberi nasehat yang konsisten dan mengurangi kesalahan
12. Meningkatkan kapabilitas sistem terkomputerisasi yang lain. Integrasi Sistem Pakar dengan sistem komputer lain membuat lebih efektif dan bisa mencakup lebih banyak aplikasi.
13. Mampu menyediakan pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman. Fasilitas penjelas dapat berfungsi sebagai guru.

#### KELEMAHAN SISTEM PAKAR

1. Biaya yang diperlukan untuk membuat, memelihara, dan mengembangkannya sangat mahal
2. Sulit dikembangkan, hal ini erat kaitannya dengan ketersediaan pakar di bidangnya dan kepakaran sangat sulit diekstrak dari manusia karena sangat sulit bagi seorang pakar untuk menjelaskan langkah mereka dalam menangani masalah.
3. Sistem pakar tidak 100% benar karena seseorang yang terlibat dalam pembuatan sistem pakar tidak selalu benar. Oleh karena itu perlu diuji ulang secara teliti sebelum digunakan.
4. Pendekatan oleh setiap pakar untuk suatu situasi atau problem bisa berbeda-beda, meskipun sama-sama benar.
5. Transfer pengetahuan dapat bersifat subjektif dan bias
6. Kurangnya rasa percaya pengguna dapat menghalangi pemakaian sistem pakar.

### 6.3 Konsep Dasar Sistem Pakar

Konsep dasar sistem pakar mengandung keahlian, ahli/pakar, pengalihan keahlian, mengambil keputusan, aturan, kemampuan menjelaskan.

#### Keahlian

Keahlian bersifat luas dan merupakan penguasaan pengetahuan dalam bidang khusus yang diperoleh dari pelatihan, membaca atau pengalaman. Contoh bentuk pengetahuan yang termasuk keahlian

- Teori, fakta, aturan-aturan pada lingkup permasalahan tertentu
- Strategi global untuk menyelesaikan masalah

#### Ahli / Pakar

Seorang ahli adalah seseorang yang mampu menjelaskan suatu tanggapan,

mempelajari hal-hal baru seputar topik permasalahan, menyusun kembali pengetahuan jika dipandang perlu, memecahkan masalah dengan cepat dan tepat

#### Pengalihan keahlian

Tujuan dari sistem pakar adalah untuk mentransfer keahlian dari seorang pakar ke dalam komputer kemudian ke masyarakat. Proses ini meliputi 4 kegiatan, yaitu perolehan pengetahuan (dari para ahli atau sumber-sumber lainnya), representasi pengetahuan ke komputer, kesimpulan dari pengetahuan dan pengalihan pengetahuan ke pengguna

#### Mengambil keputusan

Hal yang unik dan sistem pakar adalah kemampuan untuk menjelaskan dimana keahlian tersimpan dalam basis pengetahuan. Kemampuan komputer untuk mengambil kesimpulan dilakukan oleh komponen yang dikenal dengan mesin inferensi yaitu meliputi prosedur tentang pemecahan masalah.

#### Aturan

Sistem pakar yang dibuat merupakan sistem yang berdasarkan pada aturan – aturan dimana program disimpan dalam bentuk aturan-stores sebagai prosedur pemecahan masalah. Aturan tersebut biasanya berbentuk IF – THEN.

#### Kemampuan menjelaskan

Keunikan lain dari sistem pakar adalah kemampuan dalam menjelaskan atau memberi saran/rekomendasi serta juga menjelaskan mengapa beberapa tindakan/saran tidak direkomendasikan.

#### PERBEDAAN SISTEM KONVENSIONAL DENGAN SISTEM PAKAR

Sistem Konvensional	Sistem Pakar
Informasi dan pemrosesannya biasanya jadi satu dengan program	Basis pengetahuan merupakan bagian terpisah dari mekanisme inferensi
Program tidak pernah salah (kecuali pemrogramnya yang salah)	Program bisa saja melakukan kesalahan
Biasanya tidak bisa menjelaskan mengapa suatu input data itu dibutuhkan atau bagaimana output itu diperoleh	Penjelasan adalah bagian terpenting dari system pakar
Pengubahan program cukup sulit dan merepotkan	Pengubahan pada aturan/kaidah dapat dilakukan dengan mudah
Sistem hanya akan bekerja jika sistem tersebut sudah lengkap	Sistem dapat bekerja hanya dengan beberapa aturan
Eksekusi dilakukan langkah demi	Eksekusi dilakukan pada keseluruhan



langkah secara algoritmik	basis pengetahuan secara heuristik dan logis
Menggunakan data	Menggunakan pengetahuan
Tujuan utamanya, adalah efisiensi	Tujuan utamanya adalah efektivitas

#### ELEMEN MANUSIA YANG TERKAIT DALAM PENGGUNAAN DAN PENGEMBANGAN SISTEM PAKAR

##### 1. Pakar

Pakar adalah orang yang memiliki pengetahuan khusus, pendapat, pengalaman dan metode, serta kemampuan untuk mengaplikasikan keahliannya tersebut guna menyelesaikan masalah.

##### 2. Perekayasa pengetahuan.

Perekayasa pengetahuan adalah orang yang membantu pakar dalam menyusun area permasalahan dengan menginterpretasikan dan mengintegrasikan jawaban-jawaban pakar atas pertanyaan yang diajukan, menggambarkan analogi, mengajukan counter example dan menerangkan kesulitan-kesulitan konseptual.

##### 3. Pemakai

- Pemakai awam : dalam hal ini sistem pakar bertindak sebagai konsultan untuk memberikan saran dan solusi kepada pemakai
- Pelajar yang ingin belajar : sistem pakar bertindak sebagai instruktur
- Pembuat sistem pakar : sistem pakar sebagai partner dalam pengembangan basis pengetahuan.
- Pakar : sistem pakar berindak sebagai mitra kerja/asisten

#### AREA PERMASALAHAN APLIKASI SISTEM PAKAR

##### 1. Interpretasi

Yaitu pengambilan keputusan dari hasil observasi, diantaranya : pengawasan, pengenalan ucapan, analisis citra, interpretasi sinyal, dan beberapa analisis kecerdasan

##### 2. Prediksi

Memprediksi akibat-akibat yang dimungkinkan dari situasi-situasi tertentu, diantaranya peramalan, prediksi demografis, peralihan ekonomi, prediksi lalu lintas, estimasi hasil, militer, pem&wan, atau peramalan keuangan.

##### 3. Diagnosis

Menentukan sebab malfungsi dalam situasi kompleks yang didasarkan pada gejala-gejala yang teramati, diantaranya : medis, elektronis, mekanis, dan diagnosis perangkat lunak

##### 4. Desain

Menentukan konfigurasi komponen-komponen sistem yang cocok dengan

tujuan-tujuan kinerja tertentu dan kendala-kendala tertentu, diantaranya :  
layout sirkuit, perancangan bangunan

5. Perencanaan  
Merencanakan serangkaian tindakan yang akan dapat mencapai sejumlah tujuan dengan kondisi awal tertentu, diantaranya : perencanaan keuangan, komunikasi, militer, pengembangan politik, routing dan manajemen proyek.
6. Monitoring  
Membandingkan tingkah laku suatu sistem yang teramati dengan tingkah laku yang diharapkan darinya, diantaranya: Computer Aided Monitoring System
7. Debugging dan repair  
Menentukan dan mengimplementasikan cara-cara untuk mengatasi malfungsi, diantaranya memberikan resep obat terhadap suatu kegagalan.
8. Instruksi  
Melakukan instruksi untuk diagnosis, debugging dan perbaikan kinerja.
9. Kontrol  
Mengatur tingkah laku suatu environment yang kompleks seperti kontrol terhadap interpretasi-interpretasi, prediksi, perbaikan, dan monitoring kelakuan sistem
10. Seleksi  
Mengidentifikasi pilihan terbaik dari sekumpulan (list) kemungkinan.
11. Simulasi  
Pemodelan interaksi antara komponen-komponen sistem.

#### **6.4 Bentuk dan Struktur Sistem**

##### **BENTUK / TIPE SISTEM PAKAR**

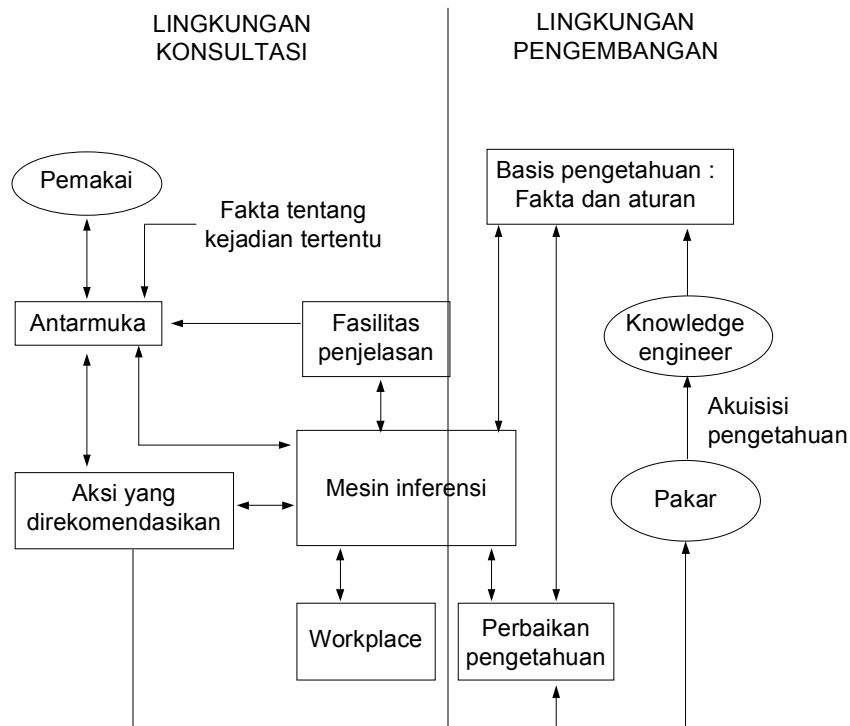
1. Mandiri : sistem pakar yang murni berdiri sendiri, tidak digabung dengan software lain, bisa dijalankan pada komputer pribadi, mainframe.
2. Terkait/Tergabung : dalam bentuk ini sistem pakar hanya merupakan bagian dari program yang lebih besar. Program tersebut biasanya menggunakan teknik algoritma konvensional tapi bisa mengakses sistem pakar yang ditempatkan sebagai subrutin yang bisa dimanfaatkan setiap kali dibutuhkan.
3. Terhubung : merupakan sistem pakar yang berhubungan dengan software lain, misal spreadsheet, DBMS, program grafik. Pada saat proses inferensi, sistem pakar bisa mengakses data dalam spreadsheet atau DBMS atau program grafik bisa dipanggil untuk menayangkan output visual.
4. Sistem Mengabdikan  
Merupakan bagian dari komputer khusus, yang diabdikan kepada fungsi tunggal. Sistem tersebut bisa membantu analisa data radar dalam pesawat tempur atau membuat keputusan intelejen tentang bagaimana memodifikasi pembangunan kimiawi, dll.

## STRUKTUR SISTEM PAKAR

2 bagian utama sistem pakar :

- lingkungan pengembangan (development environment)  
digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan sistem pakar
- lingkungan konsultasi (consultation environment)  
digunakan oleh pengguna yang bukan pakar untuk memperoleh pengetahuan pakar

arsitektur system pakar



Komponen-komponen yang terdapat dalam arsitektur/shuktur sistem pakar

### 1. Antarmuka Pengguna (User Interface)

Merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Antarmuka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selama itu antarmuka menerima dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai

### 2. Basis Pengetahuan

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi dan penyelesaian masalah. Komponen sistem pakar ini disusun atas 2 elemen dasar, yaitu

- **Fakta** : informasi tentang obyek dalam area permasalahan tertentu

- Aturan : informasi tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui.
3. Akuisisi Pengetahuan (Knowledge Acquisition)
- Akuisisi pengetahuan adalah akumulasi, transfer, dan transformasi keahlian dalam menyelesaikan masalah dan sumber pengetahuan ke dalam program komputer. Dalam tahap ini knowledge engineer berusaha menyerap pengetahuan untuk selanjutnya ditransfer ke dalam basis pengetahuan. Pengetahuan diperoleh dan pakar, dilengkapi dengan buku, basis data, laporan penelitian dan pengalaman pemakai.
- Metode akuisisi pengetahuan
- Wawancara  
Metode yang paling banyak digunakan, yang melibatkan pembicaraan dengan pakar secara langsung dalam suatu wawancara
  - Analisis protokol  
Dalam metode ini pakar diminta untuk melakukan suatu pekerjaan dan mengungkapkan proses pemikirannya dengan menggunakan kata-kata. Pekerjaan tersebut direkam, dituliskan, dan dianalisis.
  - Observasi pada pekerjaan pakar  
Pekerjaan dalam bidang tertentu yang dilakukan pakar direkam dan diobservasi
  - Induksi aturan dan contoh  
Induksi adalah suatu proses penalaran dari khusus ke umum. Suatu sistem induksi aturan diberi contoh-contoh dari suatu masalah yang hasilnya telah diketahui. Setelah diberikan beberapa contoh, sistem induksi aturan tersebut dapat membuat aturan yang benar untuk kasus-kasus contoh. Selanjutnya aturan dapat digunakan untuk menilai kasus lain yang hasilnya tidak diketahui.
4. Mesin/Motor Inferensi (inference engine)
- Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Mesin inferensi adalah program komputer yang membenkan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam workplace, dan untuk memformulasikan kesimpulan.
5. Workplace / Blackboard
- Workplace merupakan area dan sekumpulan memori kerja (working memory), digunakan untuk merekam kejadian yang sedang berlangsung termasuk keputusan sementara. Ada 3 keputusan yang dapat direkam :
- Rencana : bagaimana menghadapi masalah
  - Agenda : aksi-aksi yang potensial yang sedang menunggu untuk dieksekusi
  - Solusi : calon aksi yang akan dibangkitkan

## 6. Fasilitas Penjelasan

Adalah komponen tambahan yang akan meningkatkan kemampuan sistem pakar. Digunakan untuk melacak respon dan memberikan penjelasan tentang kelakuan sistem pakar secara interaktif melalui pertanyaan:

- mengapa suatu pertanyaan ditanyakan oleh sistem pakar ?
- bagaimana konklusi dicapai ?
- mengapa ada alternatif yang dibatalkan ?
- rencana apa yang digunakan untuk mendapatkan solusi ?

## 7. Perbaikan Pengetahuan

Pakar memiliki kemampuan untuk menganalisis dan meningkatkan kinerjanya serta kemampuan untuk belajar dari kinerjanya. Kemampuan tersebut adalah penting dalam pembelajaran terkomputerisasi, sehingga program akan mampu menganalisis penyebab kesuksesan dan kegagalan yang dialaminya dan juga mengevaluasi apakah pengetahuan-pengetahuan yang ada masih cocok untuk digunakan di masa mendatang

### 6.5 Basis Pengetahuan (Knowledge Base)

Basis pengetahuan berisi pengetahuan-pengetahuan dalam penyelesaian masalah. Ada 2 bentuk pendekatan basis pengetahuan :

#### a. Penalaran berbasis aturan (rule-based reasoning)

Pada penalaran berbasis aturan, pengetahuan direpresentasikan dengan menggunakan aturan berbentuk IF-THEN. Bentuk ini digunakan apabila kita memiliki sejumlah pengetahuan pakar pada suatu permasalahan tertentu, dan si pakar dapat menyelesaikan masalah tersebut secara berurutan. Disamping itu, bentuk ini juga digunakan apabila dibutuhkan penjelasan tentang jejak (langkah-langkah) pencapaian solusi.

Contoh : aturan identifikasi hewan

Rule 1 : IF hewan berambut dan menyusui THEN hewan mamalia,

Rule 2: IF hewan mempunyai sayap dan bertelur THEN hewan jenis burung

Rule 3: IF hewan mamalia, dan memakan daging THEN hewan karnivora Dst...

#### b. Penalaran berbasis kasus (case-based reasoning)

Pada penalaran berbasis kasus, basis pengetahuan akan berisi solusi-solusi yang telah dicapai sebelumnya, kemudian akan diturunkan suatu solusi untuk keadaan yang terjadi sekarang (fakta yang ada). Bentuk ini digunakan apabila user menginginkan untuk tahu lebih banyak lagi pada kasus-kasus yang hampir sama (mirip). Selain itu bentuk ini juga digunakan bila kita telah memiliki sejumlah situasi atau kasus tertentu dalam basis pengetahuan.

## 6.6 Mesin Inferensi (Inference Engine)

Ada 2 cara, penalaran yang dapat dikerjakan dalam melakukan inferensi

- a. Forward Chaining  
Pencocokan fakta atau pernyataan dimulai dan bagian sebelah kiri dulu (IF dulu). Dengan kata lain penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis.
  
- b. Backward Chaining  
Pencocokan fakta atau pernyataan dimulai dan bagian sebelah kanan (THEN dulu). Dengan kata lain penalaran dimulai dari hipotesis terlebih dahulu, dan untuk menguji kebenaran hipotesis tersebut harus dicari fakta-fakta yang ada dalam basis pengetahuan.

Contoh :

R1 : IF suku bunga turun THEN harga obligasi naik

R2 : IF suku bunga naik THEN harga obligasi turun

R3 : IF suku bunga tidak berubah THEN harga, obligasi tidak berubah

R4 : IF dolar naik THEN suku bunga turun

R5 : IF dolar turun THEN suku bunga naik

R6 : IF harga obligasi turun THEN beli obligasi

Apabila diketahui bahwa dolar turun, maka untuk memutuskan apakah akan membeli obligasi atau tidak dapat ditunjukkan sebagai berikut :

Forward Chaining

Dari fakta dolar turun, berdasarkan Rule 5, diperoleh konklusi suku bunga naik Dan Rule 2 suku bunga naik menyebabkan harga obligasi turun. Dengan Rule 6, jika harga, obligasi turun, maka kesimpulan yang diambil adalah membeli obligasi.

Backward Chaining

Dari solusi yaitu membeli obligasi, dengan menggunakan Rule 6 diperoleh anteseden harga obligasi turun. Dari Rule 2 dibuktikan harga obligasi turun bernilai benar jika suku bunga naik bernilai benar. Dari Rule 5 suku bunga naik bernilai memang bernilai benar karena diketahui fakta dolar turun.

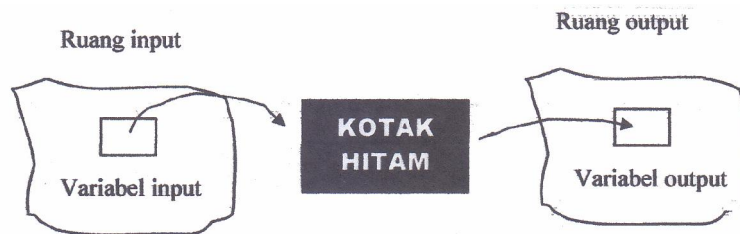
### LANGKAH-LANGKAH PEMBUATAN SISTEM PAKAR

1. Mengidentifikasi masalah dan kebutuhan
2. Menentukan problema yang cocok
3. mempertimbangkan alternative
4. menghitung pengembalian investasi
5. Memilih alat pengembangan
6. merekayasa pengetahuan

7. merancang system
8. melengkapi pengembangan
9. menguji dan mencari kesalahan system
10. memelihara sistem

## BAB VII LOGIKA FUZZY

Logika fuzzy adalah suatu cara untuk memetakan suatu ruang input ke dalam suatu ruang output. Skema logika fuzzy :



Antara input dan output terdapat suatu kotak hitam yang harus memetakan input ke output yang sesuai. Misal :



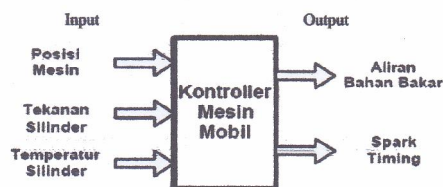
Pemetaan input-output pada masalah produksi : “diberikan data persediaan barang, berapa jumlah barang yang harus diproduksi ?”

Ada beberapa cara/metode yang mampu bekerja di kotak hitam tersebut, misal : sistem fuzzy, jaringan syaraf tiruan, sistem linear, sistem pakar, persamaan diferensial, dll.

Namun menurut Prof. Lotfi A. Zadeh, penemu teori logika fuzzy di tahun 1960-an:



“pada hampir semua kasus kita dapat menghasilkan suatu produk tanpa menggunakan logika fuzzy, namun menggunakan fuzzy akan lebih cepat dan lebih murah”





## 7.1 ALASAN MENGGUNAKAN FUZZY

1. Konsep logika fuzzy mudah dimengerti
2. Logika fuzzy sangat fleksibel
3. Memiliki toleransi terhadap data-data yang tidak tepat
4. Dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan
5. Logika fuzzy didasarkan pada bahasa alami

## APLIKASI LOGIKA FUZZY

1. Tahun 1990 pertama kali mesin cuci dengan logika fuzzy di Jepang (Matsushita Electric Industrial Company). Sistem fuzzy digunakan untuk menentukan putaran yang tepat secara otomatis berdasarkan jenis dan banyaknya kotoran serta jumlah yang akan dicuci. Input yang digunakan : seberapa kotor, jenis kotoran, banyaknya yang dicuci. Mesin ini menggunakan sensor optik, mengeluarkan cahaya ke air dan mengukur bagaimana cahaya tersebut sampai ke ujung lainnya. Makin kotor, maka sum yang sampai makin redup. Sistem juga mampu menentukan jenis kotoran tersebut daki/minyak.
2. Transmisi otomatis pada mobil Nissan, menghemat bensin 12 – 17 %
3. Kereta bawah tanah Sendai mengontrol pemberhentian otomatis pada area tertentu
4. Ilmu kedokteran dan biologi, seperti sistem diagnosis kanker
5. Manajemen dan pengambilan keputusan, misal tata letak pabrik berdasarkan logika fuzzy, pembuatan games berdasarkan logika fuzzy, dll.
6. Ilmu lingkungan, misal kendali kualitas air, prediksi cuaca.
7. Teknik, misal perancangan jaringan komputer, prediksi adanya gempa bumi, dll.
8. dsb

## 7.2 HIMPUNAN DAN FUNGSI KEANGGOTAAN

### HIMPUNAN TEGAS (CRISP)

= nilai keanggotaan suatu item  $x$  dalam suatu himpunan  $A$ , yang sering ditulis dengan  $\mu_A[x]$ , memiliki 2 kemungkinan, yaitu :

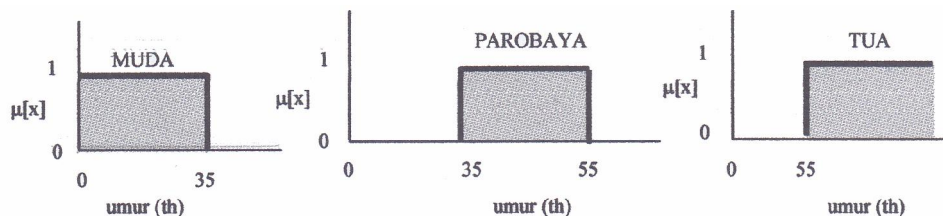
- 1, yang berarti bahwa item tersebut ( $x$ ) anggota himpunan  $A$
- 0, yang berarti bahwa, item tersebut ( $x$ ) bukan anggota himpunan  $A$

contoh :

- $S = [1,2,3,4,5,6]$  adalah semesta, pembicaraan  
 $A = [1,2,3]$   
 $B = [3,4,5]$   
Jadi :  
Nilai keanggotaan 2 pada himpunan  $A \rightarrow \mu_A[2] = 1$  karena  $2 \in A$   
Nilai keanggotaan 3 pada himpunan  $A \rightarrow \mu_A[3] = 1$  karena  $3 \in A$

Nilai keanggotaan 4 pada himpunan B  $\rightarrow \mu_A[4] = 0$  karena  $4 \in A$   
 Nilai keanggotaan 2 pada himpunan B  $\rightarrow \mu_A[2] = 0$  karena  $2 \in B$   
 Nilai keanggotaan 3 pada himpunan A  $\rightarrow \mu_A[3] = 1$  karena  $2 \in B$

- misal variabel umur dibagi menjadi 3 kategori, yaitu  
 MUDA umur < 35 tahun  
 PAROBAYA  $35 \leq \text{umur} \leq 55$  tahun  
 TUA umur > 55 tahun  
 Nilai keanggotaan secara grafis, himpunan MUDA, PAROBAYA, TUA :

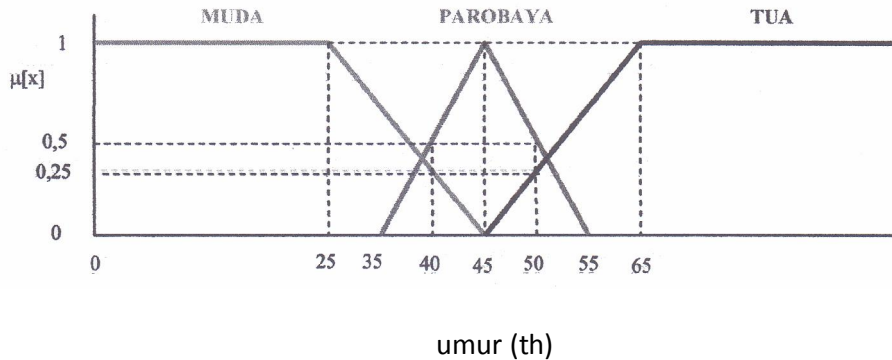


usia 34 tahun maka dikatakan MUDA  $\rightarrow \mu_{MUDA}[34] = 1$   
 usia 35 tahun maka dikatakan TIDAKMUDA  $\rightarrow \mu_{MUDA}[35] = 0$   
 usia 35 tahun maka dikatakan PAROBAYA  $\rightarrow \mu_{PAROBAYA}[35] = 1$   
 usia 34 tahun maka dikatakan TIDAKPAROBAYA  $\rightarrow \mu_{PAROBAYA}[34] = 0$   
 usia 35 tahun kurang 1 hari maka dikatakan TIDAKPAROBAYA  $\rightarrow \mu_{PAROBAYA} [35$   
 $\text{th} - 1 \text{ hari}] = 0$   
 usia 35 tahun lebih 1 hari maka dikatakan TIDAKMUDA  $\rightarrow \mu_{MUDA}[35 \text{ th} + 1 \text{ hari}]$   
 $= 0$

Himpunan crisp untuk menyatakan umur bisa tidak adil karena adanya perubahan kecil saja pada suatu nilai mengakibatkan perbedaan kategori yang cukup signifikan.

#### HIMPUNAN FUZZY

Himpunan fuzzy digunakan untuk mengantisipasi hal tersebut diatas. Seseorang dapat masuk dalam 2 himpunan yang berbeda, MUDA dan PAROBAYA, PAROBAYA dan TUA, dsb. Seberapa besar eksistensinya dalam himpunan tersebut dapat dilihat pada nilai/derajat keanggotaannya. Himpunan fuzzy untuk variabel UMUR :



usia 40 tahun termasuk dalam himpunan MUDA dengan  $\mu_{MUDA}[40] = 0,25$   
 termasuk juga dalam himpunan PAROBAYA dengan  $\mu_{PAROBAYA}[40] = 0,5$   
 usia 50 tahun termasuk dalam himpunan = TUA dengan  $\mu_{TUA}[50] = 0,25$   
 termasuk juga dalam himpunan PAROBAYA dengan  $\mu_{PAROBAYA}[50] = 0,5$

Himpunan crisp, nilai keanggotaan hanya 0 dan 1.  
 Himpunan fuzzy, derajat/nilai keanggotaan terletak pada rentang 0 sampai 1 sehingga :  
 Bila x memiliki derajat keanggotaan fuzzy  $\mu_A[x] = 0 \rightarrow x$  bukan anggota himpunan A  
 Bila x memiliki derajat keanggotaan fuzzy  $\mu_A[x] = 1 \rightarrow x$  anggota penuh himpunan A

#### FUNGSI KEANGGOTAAN (Membership function)

= suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai/derajat yang memiliki interval antara 0 sampai 1.

Contoh grafik diatas adalah suatu fungsi keanggotaan untuk variabel UMUR yang dibagi menjadi 3 kategori atau 3 himpunan fuzzy yaitu MUDA, PAROBAYA, TUA, dimana dapat direpresentasikan sebagai berikut :

$$\mu_{MUDA}[x] = \begin{cases} 1, & x \leq 25 \\ \frac{45-x}{45-25}, & 25 < x < 45 \\ 0, & x \geq 45 \end{cases} \quad \mu_{TUA}[x] = \begin{cases} 0, & x \leq 45 \\ \frac{x-45}{65-45}, & 45 < x < 65 \\ 1, & x \geq 65 \end{cases}$$

$$\mu_{PAROBAYA}[x] = \begin{cases} 0, & x \leq 35 \\ \frac{x-35}{45-35}, & 35 < x < 45 \\ \frac{55-x}{55-45}, & 45 \leq x \leq 55 \end{cases}$$

## WATAK KEKABURAN

Perhatikan pernyataan dibawah ini

Mesin yang digunakan terus-menerus akan cepat panas

→ kita tidak dapat menentukan dengan tepat batasan terus-menerus, cepat, dan panas

Jika air pancuran terlalu panas maka naikkan aliran air dingin perlahan-lahan

→ kita tidak dapat menentukan dengan tepat batasan terlalu panas, menaikkan, air yang dingin, dan perlahan-lahan



Maka solusinya dengan menggunakan LOGIKA FUZZY (logika samar)

## VARIABEL LINGUISTIK

- Variabel linguistik = sebuah variabel yang memiliki nilai berupa kata-kata dalam bahasa alamiah bukan angka.
- Mengapa menggunakan kata/kalimat daripada angka ? → karena peranan linguistik memang kurang spesifik dibandingkan angka, namun informasi yang disampaikan lebih informatif.  
Contoh, jika "KECEPATAN" adalah variabel linguistik, maka nilai linguistik untuk variabel kecepatan adalah, misalnya "LAMBAT", "SEDANG", "CEPAT". Hal ini sesuai dengan kebiasaan manusia sehari-hari dalam menilai sesuatu, misalnya : "la mengendarai mobil dengan cepat", tanpa memberikan nilai berapa kecepatannya.
- Setiap variabel linguistik berkaitan dengan sebuah fungsi keanggotaan.
- Menurut Wang (1997) definisi formal dari variabel linguistik diberikan sebagai berikut. Sebuah variabel linguistik dikarakteristik oleh  $(X, T(x), U, M)$ , dimana :
  - X : Nama variabel (variabel linguistik) yang menjadi objek
  - T(x) : Himpunan semua istilah (nilai-nilai) linguistik yang terkait dengan (nama) variabel (X) yang menggambarkan objek tersebut
  - U : Domain fisik aktual/ruang lingkup dimana variabel linguistik X mengambil nilai-nilai kuantitatifnya/nilai numeric (crisp) → himpunan semesta
  - M : Suatu aturan semantik yang menghubungkan setiap nilai linguistik dalam T dengan suatu himpunan fuzzy dalam U.

Dari contoh diatas, maka diperoleh:

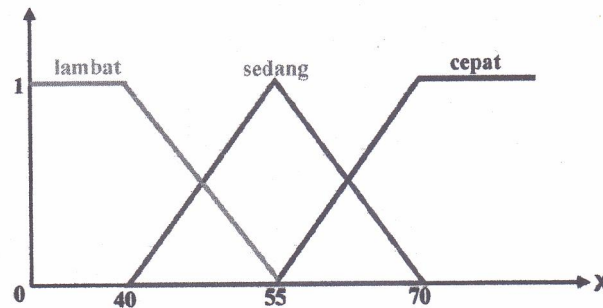
X = kecepatan

U = [0 , 100] → maksudnya domain/ruang lingkup kecepatan misal dari 0 sampai 100 km/jam

T(kecepatan) = {lambat, sedang, cepat} → maksudnya variabel kecepatan terbagi menjadi 3 himpunan fuzzy yaitu lambat, sedang, cepat

Maka M untuk setiap X, M(x) adalah: M(lambat), M(sedang), M(cepat)  
M(lambat) = himpunan fuzzynya "kecepatan dibawah 40 Km/jam" dengan fungsi keanggotaan  $\mu_{lambat}$   
M(sedang) = himpunan fuzzynya "kecepatan mendekati 55 Km/jam" dengan fungsi keanggotaan  $\mu_{sedang}$   
M(cepat) = himpunan fiizzynya. "kecepatan diatas 70 Km/Jam" dengan fungsi keanggotaan  $\mu_{cepat}$

Gambar grafik fungsi keanggotaanya sebagai berikut :



Grafik fungsi keanggotaan kecepatan

Sehingga himpunan fuzzy untuk.

$M(lambat) = \{(0,1), (1,1),(2,1), \dots, (40,1), \dots, (47,0.533), \dots, (55,0),(56,0)\dots (100,0)\}$

$M(sedang) = \{(0,0), (1,0),(2,0), \dots, (40,0), \dots, (47,0.533), \dots, (55,1),(56,0.933)\dots (100,0)\}$

$M(cepat) = \{(0,0), (1,1),(2,1), \dots, (40,1), \dots, (47,0), \dots, (55,0),(56,0.866)\dots (100,1)\}$

### 7.3 OPERASI DASAR HIMPUNAN FUZZY (Operator Zadeh)

Digunakan untuk mengkombinasi dan memodifikasi himpunan fuzzy. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan disebut fire strength atau a predikat.

Operator	Operasi	Fungsi Keanggotaan
AND	Intersection	$\mu_{(A \cap B)}(x) = \min [\mu A(x), \mu B(x)]$
OR	Union	$\mu_{(A \cup B)}(x) = \max [\mu A(x), \mu B(x)]$
NOT	Complement	$\mu_{A^c}(x) = 1 - \mu A(x)$

Contoh :

- $U = \{1,2,3,4,5,6\}$   
 $A = \{(1,0), (2,0,2), (3,0,6), (4,0,9), (5,1), (6,0,8)\}$   
 $B = \{(1,0,8), (2,1), (3,0,7), (4,0,4), (5,0,1), (6,0)\}$

Maka  $\alpha$  predikat untuk :

$A^c = \{(1,1), (2,0,8), (3,0,3), (4,0,1), (5,0), (6,0,2)\}$

$$B^c = \{(1,0.2), (2,0), (3,0.3), (4,0.6), (5,0.9), (6,1)\}$$

$$A \cap B = \{(1,0), (2,0.2), (3,0.6), (4,0.4), (5,0.1), (6,0)\}$$

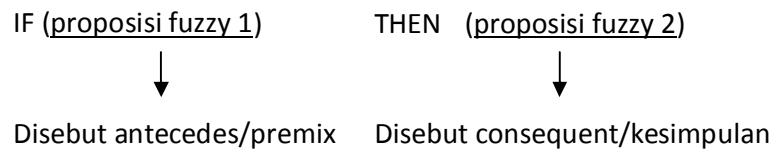
$$A \cup B = \{(1,0.8), (2,1), (3,0.7), (4,0.9), (5,1), (6,0.8)\}$$

- Misal derajat keanggotaan 27 tahun pada himpunan MUDA adalah 0.6 ( $\mu_{MUDA}[27] = 0.6$ )  
Derajat keanggotaan Rp. 2 juta pada himpunan penghasilan TINGGI adalah 0.8 ( $\mu_{GAJITINGGI}[2juta] = 0.8$ ) maka  $\alpha$  predikat untuk usia MUDA dan berpenghasilan TINGGI :  
$$\mu_{MUDA} \cap \mu_{GAJITINGGI} = \min(\mu_{MUDA}[27], \mu_{GAJITINGGI}[2juta])$$
$$= \min(0.6, 0.8) = 0.6$$

#### 7.4 PENALARAN MONOTON

##### ATURAN (RULE) IF-THEN FUZZY

- Aturan IF-THEN fuzzy adalah pernyataan IF-THEN dimana beberapa kata-kata dalam pernyataan tersebut ditentulcm oleh fungsi keanggotaan.
- Aturan produksi fuzzy adalah relax; fuzzy antara dua proposisi fuzzy. Aturan tersebut dinyatakan dalam bentuk:



- Proposisi fuzzy adalah memiliki derajat kebenaran yang dinyatakan dalam suatu bilangan dalam bentuk interval  $[0, 1]$ , dimana benar dinyatakan oleh nilai 1 dan salah dinyatakan oleh nilai 0.
- Premis dari aturan fuzzy dapat memiliki lebih dari satu bagian (premis1, premis2, ...dst), semua bagian dari premix dihitung socam simultan dan diselesaikan untuk sebuah nilai tunggal dengan menggunakan operator fuzzy dalam himpunan fuzzy.

IF premis 1 AND premis 2 THEN kesimpulan 1 AND kesimpulan 2

Dimana :      AND adalah operator fuzzy  
                  Premis 1 dan premis 2 berupa variabel masukan  
                  Kesimpulan 1 dan kesimpulan 2 berupa variabel keluaran

Contoh:

IF permintaan turun AND persediaan banyak THEN produksi barang berkurang IF  
permintaan naik AND persediaan sedikit THEN produksi barang bertambah

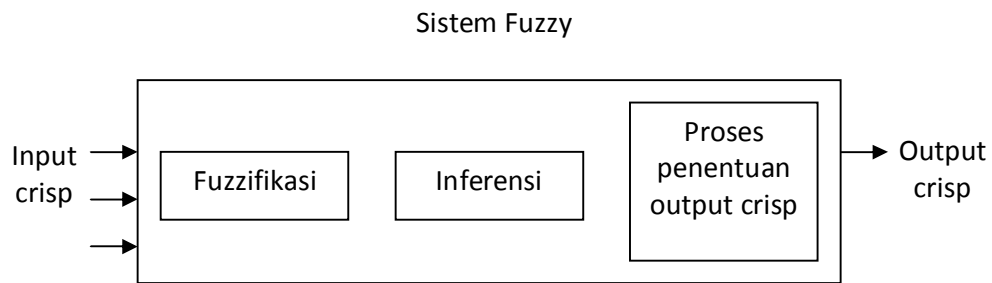
Dimana :

Permintaan, persediaan : variabel masukan  
produksi barang : variabel keluaran  
Turun, naik : kategori himpunan fuzzy dari permintaan  
Banyak, sedikit : kategori himpunan fuzzy dari persediaan  
Berkurang, bertambah : kategori himpunan fuzzy dari produksi barang

## 7.5 FUNGSI IMPLIKASI

### TAHAPAN MEMBANGUN SISTEM FUZZY

Tahapan membangun sistem fuzzy tergantung metode yang digunakan, karena banyak teori/metode untuk membangun sistem fuzzy. Namun secara garis besar dapat disimpulkan sebagai berikut:



#### Fuzzifikasi

= mengambil masukan nilai crisp dan menentukan derajat dimana nilai-nilai tersebut menjadi anggota dan setiap himpunan fuzzy yang sesuai → membuat fungsi keanggotaan

Contoh : masukan crisp 75 derajat ditransformasikan sebagai panas dalam bentuk fuzzy dengan derajat keanggotaan 0.80.

#### Inferensi

- mengaplikasikan aturan pada masukan fuzzy yang dihasilkan dalam proses fuzzyfikasi
- mengevaluasi tiap aturan dengan masukan yang dihasilkan dari proses fuzzyfikasi dengan mengevaluasi hubungan atau derajat keanggotaan antecedent/premis setiap aturan.
- Derajat keanggotaan/nilai kebenaran dan premis digunakan untuk menentukan nilai kebenaran bagian consequent/kesimpulan

Proses penentuan Output Crisp Tergantung teori/metode yang digunakan.

## 7.6 METODE INFERENSI FUZZY

### METODE ISUKAMOTO

Suatu perusahaan makanan kaleng akan memproduksi makanan jenis ABC.

Dada data 1 bulan terakhir, permintaan terbesar mencapai 5000 kemasan/hari, dan permintaan terkecil, sampai 1000 kemasan/hari.

Persediaan barang di gudang terbanyak sampai 600 kemasan/hari, dan terkecil pernah 100 kemasan/hari.

Dengan segala keterbatasannya, sampai saat ini, perusahaan baru mampu memproduksi barang maksimum 7000 kemasan/hari, serta demi efisiensi mesin dan SDM perusahaan memproduksi paling tidak 2000 kemasan.

Apabila proses produksi perusahaan tersebut menggunakan 4 aturan fuzzy :

R1 : IF permintaan turun AND persediaan banyak THEN produksi barang berkurang

R2 : IF permintaan turun AND persediaan sedikit THEN produksi barang berkurang

R3 : IF permintaan naik AND persediaan banyak THEN produksi barang bertambah

R4 : IF permintaan naik AND persediaan sedikit THEN produksi barang bertambah

Berapa kemasan maimmn jenis ABC yang harus diproduksi, jika jumlah permintaan sebanyak 4000 kemasan dan persediaan di gudang masih 300 kemasan?

Solusi :

FUZZIFIKASI → membuat fungsi keanggotaan

Ada 3 variabel fuzzy yang akan dimodelkan, yaitu :

#### 1. PERMINTAAN

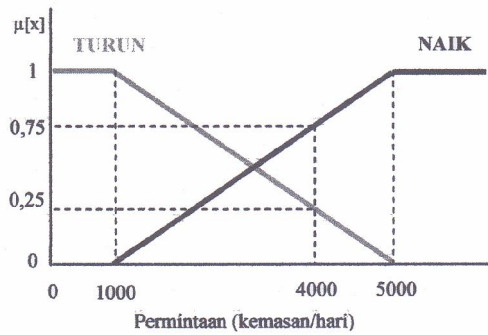
Terdiri atas 2 himpunan fuzzy yaitu NAIK dan TURUN

$$\mu_{\text{per min taan TURUN}}[x] = \begin{cases} 1, & x \leq 1000 \\ \frac{5000 - x}{5000 - 1000} & 1000 < x < 5000 \\ 0, & x \geq 5000 \end{cases}$$

$$\mu_{\text{per min taan NAIK}}[x] = \begin{cases} 0, & x \leq 1000 \\ \frac{x - 1000}{5000 - 1000} & 1000 < x < 5000 \\ 1, & x \geq 5000 \end{cases}$$

Derajat/nilai keanggotaan : $\mu_{\text{permintaan TURUN}}[4000]$ $= (5000 - 4000)/4000$ $= 0,25$
$\mu_{\text{permintaan NAIK}}[4000]$ $= (4000 - 1000)/4000$ $= 0,75$





## 2. PERSEDIAAN

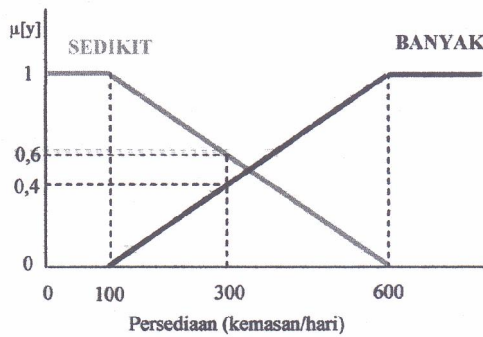
Terdiri atas 2 himpunan fuzzy yaitu SEDIKIT dan BANYAK

$$\mu_{\text{persediaanSEDIKIT}}[x] = \begin{cases} 1, & y \leq 1000 \\ \frac{600 - y}{600 - 100} & 100 < x < 600 \\ 0, & y \geq 600 \end{cases}$$

$$\mu_{\text{persediaanBANYAK}}[x] = \begin{cases} 0, & y \leq 1000 \\ \frac{Y - 100}{600 - 100} & 100 < x < 600 \\ 1, & y \geq 600 \end{cases}$$

Derajat/nilai keanggotaan :  
 $\mu_{\text{permintaan SEDIKIT}}[300]$   
 $= (600 - 300)/5000$   
 $= 0,6$

$\mu_{\text{persediaan BANYAK}}[300]$   
 $= (300 - 100)/500$   
 $= 0,4$



### 3. PRODUKSI BARANG

Terdiri atas 2 himpunan fuzzy yaitu BERKURANG dan BERTAMBAH

$$\mu_{produksiBERKURANG}[z] = \begin{cases} 1, & z \leq 2000 \\ \frac{7000 - z}{7000 - 2000} & 2000 < z < 7000 \\ 0, & z \geq 7000 \end{cases}$$
$$\mu_{produksiBERTAMBAH}[x] = \begin{cases} 0, & z \leq 2000 \\ \frac{z - 2000}{7000 - 2000}, & 2000 < z < 7000 \\ 1, & z \geq 7000 \end{cases}$$

#### MENENTUKAN OUTPUT CRISP

Pada metode TSUKAMOTO untuk menentukan output crisp menggunakan rata-rata berbobot yaitu

$$z = \frac{\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3 + \alpha_4 z_4}{\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4}$$
$$= \frac{0,25 * 5750 + 0,25 * 5750 + 0,4 * 4000 + 0,6 * 5000}{0,25 + 0,25 + 0,4 + 0,6}$$
$$= 7475 / 1,5 = 4983$$

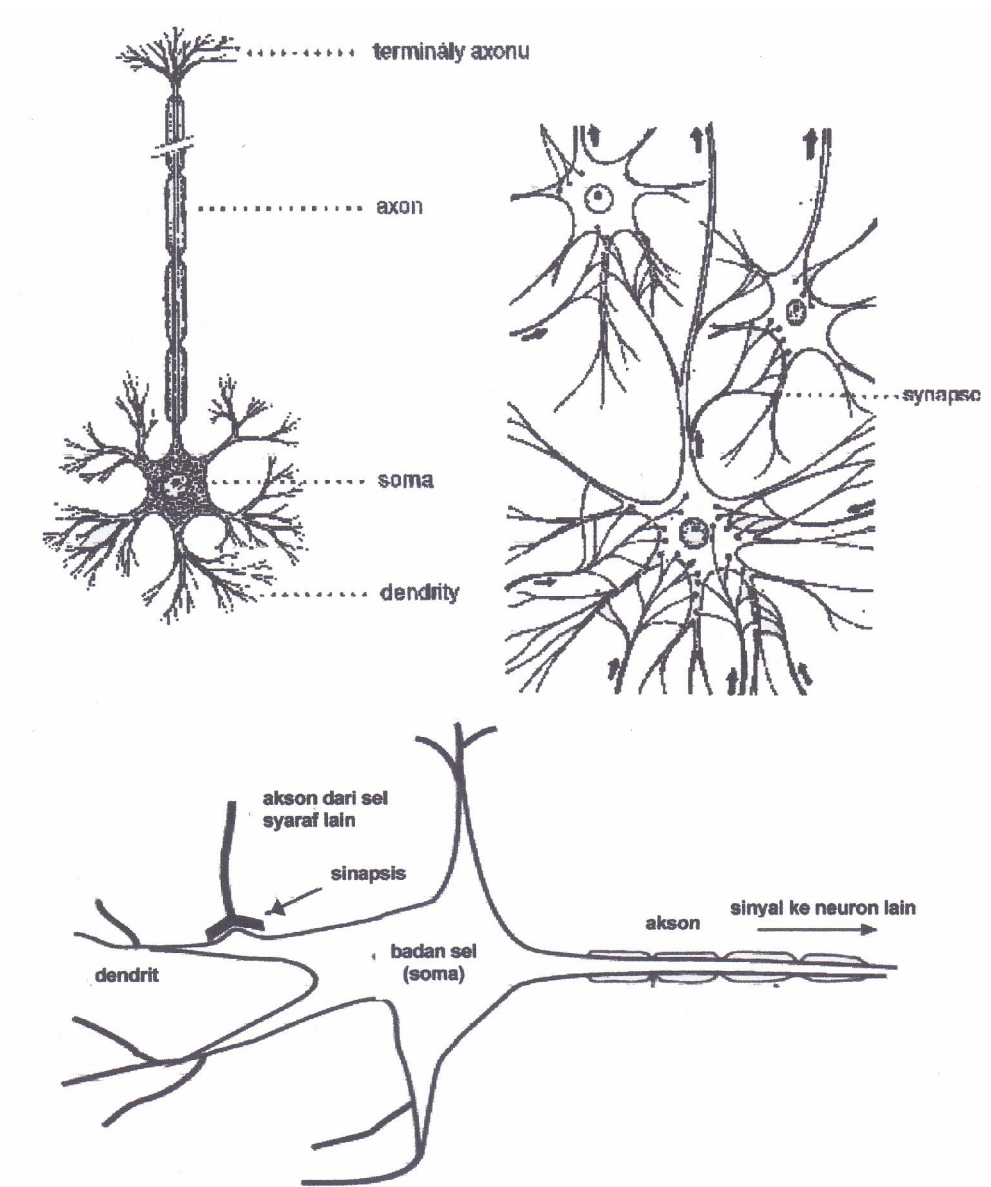
Jadi jumlah makanan kaleng yang harus diproduksi 4983 kemasan

**BAB VIII**  
**PENGANTAR JARINGAN SYARAF TIRUAN (JST)**

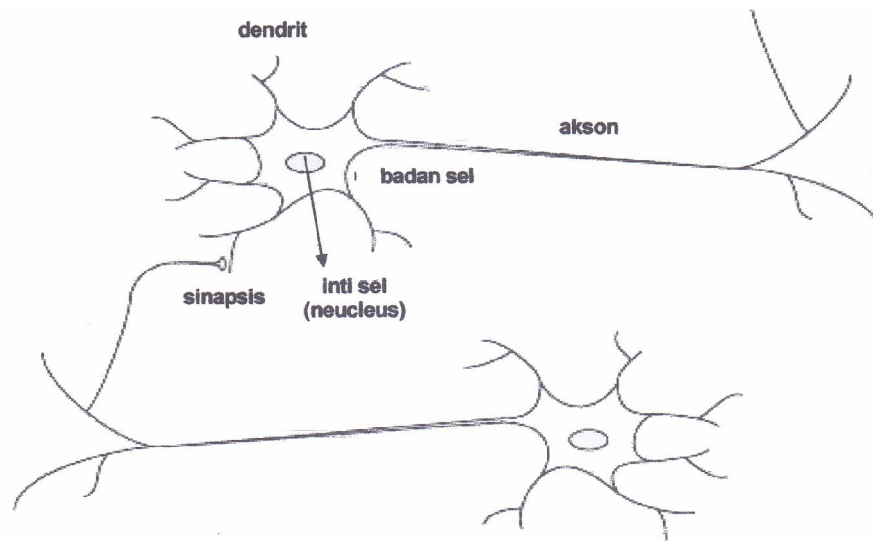
**8.1 Komponen Jaringan Syaraf**

JARINGAN SYARAF BIOLOGIS (JSB)

- Otak manusia berisi sekitar  $10^{11}$  sel syaraf (neuron) yang bertugas untuk memproses informasi yang masuk. Tiap sel syaraf dihubungkan dengan sel syaraf lain hingga sekitar  $10^4$  sinapsis. Tiap sel bekerja seperti suatu prosesor sederhana. Masing-masing sel tersebut saling berinteraksi sehingga mendukung kemampuan kerja otak manusia.



**Gambar NEURON**



Komponen utama neuron dapat dikelompokkan menjadi 3 bagian :

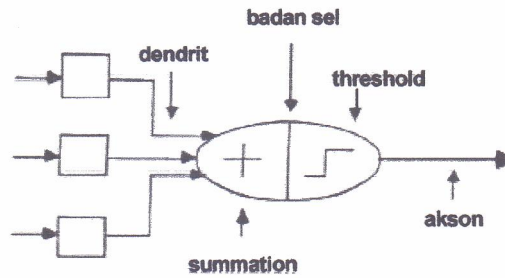
1. Dendrit = bertugas menerima informasi = jalur input bagi soma
2. Badan sel (soma) = tempat pengolahan informasi
3. Akson = bertugas mengirimkan impuls-impuls sinyal ke sel syaraf lain = jalur output bagi soma

Perhatikan gambar-gambar diatas:

- Sebuah neuron menerima impuls-impuls sinyal (informasi) dari neuron lain melalui dendrit dan mengirimkan sinyal yang dibangkitkan (hasil penjumlahan) oleh badan sel melalui akson.
- Akson dari sel syaraf ini bercabang-cabang dan berhubungan dengan dendrit dari sel syaraf lain dengan cara mengirimkan impuls melalui sinapsis
- Sinapsis adalah unit fungsional antara 2 buah sel syaraf, misal A dan B, dimana yang satu adalah serabut akson dari neuron A dan satunya lagi adalah dendrit dan neuron B.
- Kekuatan sinapsis bisa menurun / meningkat tergantung seberapa besar tingkat propagasi (penyiaran) sinyal yang diterimanya.
- Impuls-impuls sinyal (informasi) akan diterima oleh neuron lain jika memenuhi batasan tertentu, yang sering disebut dengan nilai ambang (threshold).

## 8.2 Arsitektur Jaringan

Model Struktur NEURON JSB



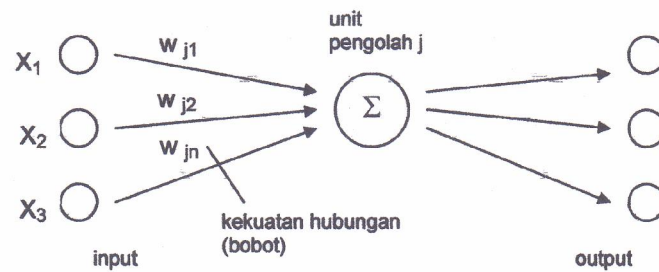
### JARINGAN SYARAF TIRUAN (JST)

- JST didefinisikan sebagai suatu sistem pemrosesan informasi yang mempunyai karakteristik menyerupai jaringan syaraf manusia (JSB)
- JST tercipta sebagai suatu generalisasi model matematis dari penvibaman manusia (human cognition) yang didasarkan atas asumsi sebagai berikut :
  1. Pemrosesan informasi terjadi pads elemen sederhana yang disebut neuron
  2. Sinyal mengalir diantara sel saraf/neuron melalui suatu sambungan penghubung
  3. Setiap sambungan penghubung memiliki bobot yang bersesuaian. Bobot ini akan digunakan untuk menggandakan / mengalikan sinyal yang dikirim melaluinya.
  4. Setiap sel syaraf akan menerapkan fungsi aktivasi terhadap sinyal hasil penjumlahan berbobot yang masuk kepadanya untuk menentukan sinyal keluarannya.
- Perbandingan kemampuan otak manusia dengan CPU

Parameter	Otak manusia	CPU
Elemen pengolah	$10^{11}$ sinapsis	$10^8$ transistor
Ukuran elemen	$10^{-6}$ m	$10^{-6}$ m
Energi yang digunakan	30 W	30 W (CPU)
Kecepatan pengolah	100 Hz	$10^9$ Hz
Bentuk komputasi	Paralel terdistribusi	Serial terpusat
Fault tolerant	Ya	Tidak
Proses belajar	Ya	Tidak
Kepandaian	selalu,	Tidak (kadang-kadang)

### Model Struktur NEURON JST

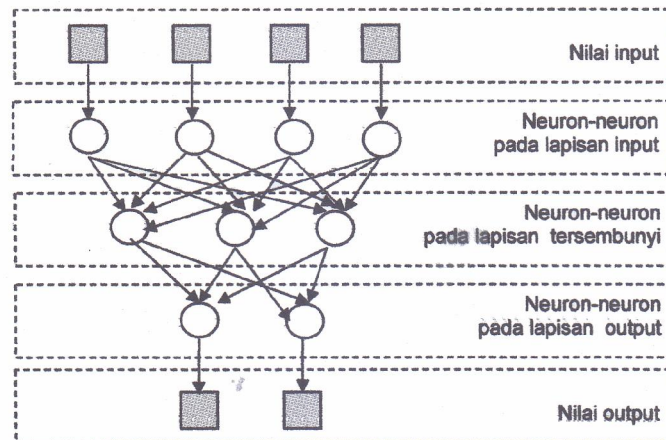




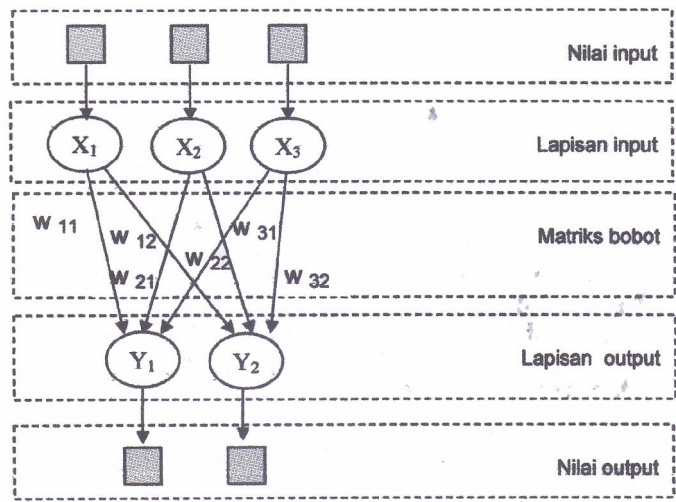
- Jaringan syaraf tiruan dapat belajar dari pengalaman, melakukan generalisasi atas contoh-contoh yang diperolehnya dan mengabstraksi karakteristik esensial input bahkan untuk data yang tidak relevan.
- Algoritma untuk JST beroperasi secara langsung dengan angka sehingga data yang tidak numerik harus diubah menjadi data numerik.
- JST tidak diprogram untuk menghasilkan keluaran tertentu. Semua, keluaran atau kesimpulan yang ditarik oleh jaringan didasarkan pada pengalamannya selama mengikuti proses pembelajaran. Pada proses pembelajaran, ke dalam JST dimasukkan pola-pola input (dan output) lalu jaringan akan diajari untuk memberikan jawaban yang bisa diterima.
- Pada dasarnya karakteristik JST ditentukan oleh :
  1. Pole hubungan antar neuron (disebut arsitektur jaringan)
  2. Metode penentuan bobot bobot sambungan (disebut dengan pelatihan atau proses belajar jaringan)
  3. Fungsi aktivasi

#### ARSITEKTUR JST

- Pada JST, neuron-neuron akan dikumpulkan dalam lapisan-lapisan (layer) yang disebut dengan lapisan neuron (neuron layers).
- Neuron-neuron pada satu lapisan akan dihubungkan dengan lapisan-lapisan sebelum dan sesudahnya.
- Informasi yang diberikan pada jaringan syaraf akan dirambatkan lapisan ke lapisan, mulai dari lapisan input sampai ke lapisan output melalui lapisan tersembunyi (hidden layer).
- Gambar berikut ini jaringan syaraf dengan 3 lapisan dan bukanlah struktur umum jaringan syaraf karena beberapa jaringan syaraf ada yang tidak memiliki lapisan tersembunyi.

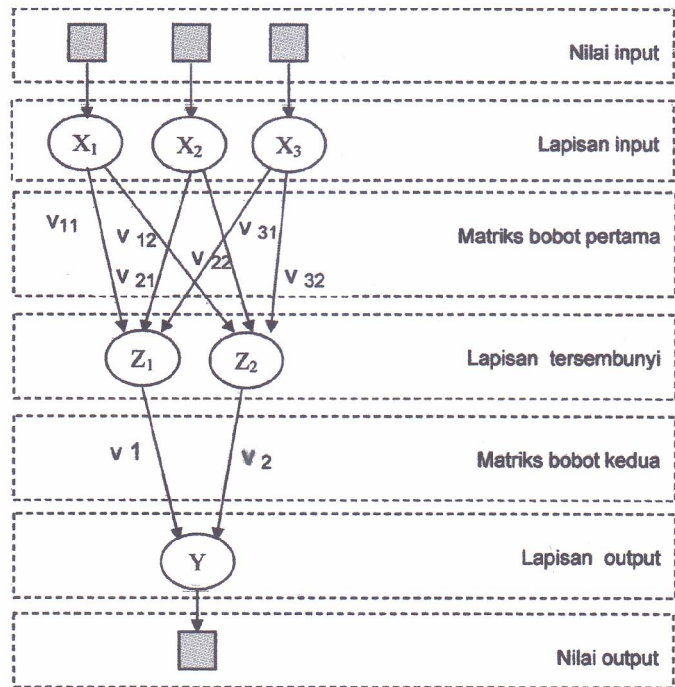


- Faktor terpenting dalam menentukan kelakuan suatu neuron adalah fungsi aktivasi dan pola bobotnya.
- Umumnya neuron-neuron yang terletak pada lapisan yang sama akan memiliki keadaan yang sama sehingga pada setiap lapisan yang sama neuron-neuron memiliki fungsi aktivasi yang sama.
- Bila neuron-neuron pada suatu lapisan (misal lapisan tersembunyi) akan dihubungkan dengan neuron-neuron pada lapisan lain (misal lapisan output) maka setiap neuron pada lapisan tersebut (lapisan tersembunyi) juga harus dihubungkan dengan setiap neuron pada lapisan lainnya, (lapisan output)
- Macam arsitektur JST ada 3 :
  1. Jaringan dengan lapisan tunggal (single layer net)  
 Hanya memiliki 1 lapisan dengan bobot-bobot terhubung. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi output tanpa harus melalui lapisan tersembunyi. Pada gambar berikut neuron-neuron pada kedua lapisan saling berhubungan. Seberapa besar hubungan antara 2 neuron ditentukan oleh bobot yang bersesuaian. Semua unit input akan dihubungkan dengan setiap unit output.



2. Jaringan dengan banyak lapisan (multilayer net)

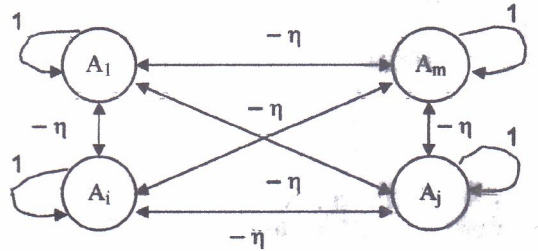
Memiliki 1 atau lebih lapisan yang terletak di antara lapisan input dan lapisan output. Umumnya ada lapisan bobot-bobot yang terletak antara 2 lapisan yang bersebelahan. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit daripada lapisan tunggal, tentu saja dengan pembelajaran yang lebih rumit. Pada banyak kasus, pembelajaran pada jaringan dengan banyak lapisan ini lebih sukses dalam menyelesaikan masalah.





3. Jaringan dengan lapisan kompetitif (competitive layer net)

Pada jaringan ini sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif. Umumnya hubungan antar neuron pada lapisan kompetitif ini tidak diperlihatkan pada diagram arsitektur. Gambar berikut menunjukkan salah satu contoh arsitektur jaringan dengan lapisan kompetitif yang memiliki bobot  $-\eta$ .



8.3 FUNGSI AKTIVASI

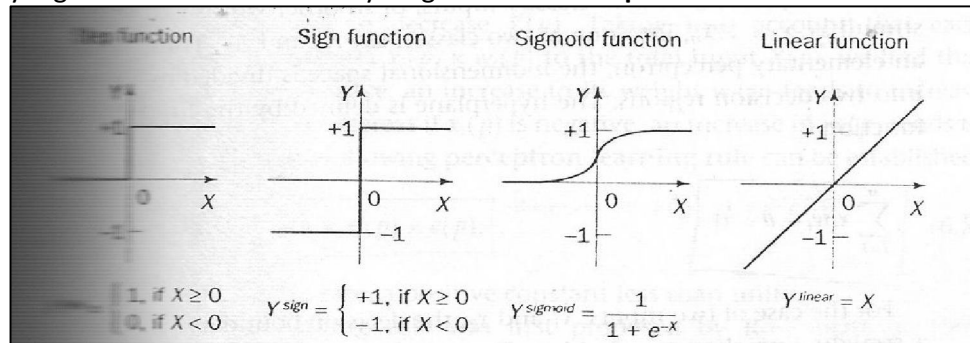
Selain Fungsi Tanda, terdapat beberapa fungsi lain yang praktis untuk digunakan oleh neuron, yaitu:

- Step Function
- Linear Function
- Sigmoid Function

Fungsi aktivasi **Step** dan **Sign** disebut juga *hard limit function*, sering digunakan oleh neuron untuk tugas-tugas **pengambilan keputusan (decision-making) dalam pengklasifikasian dan pengenalan pola.**

Fungsi **Sigmoid** mentransformasi input, yang bisa memiliki nilai di antara positif dan negatif tak terhingga, menjadi nilai yang dapat dinalar diantara 0 dan 1. Biasanya untuk jaringan **back-propagation.**

Fungsi Aktivasi **Linear** menyediakan output yang setara dengan input neuron yang diberikan bobot. Biasanya digunakan untuk **aproksimasi linear.**



8.4 PROSES PEMBELAJARAN JARINGAN

- Cara belajar JST :  
Ke dalam JST diinputkan informasi sebelumnya telah diketahui hasil keluarannya. Penginputan informasi ini dilakukan lewat node-node atau

unit-unit input. Bobot-bobot antarkoneksi dalam suatu arsitektur diberi nilai awal dan kemudian JST dijalankan. Bobot-bobot ini bagi jaringan digunakan untuk belajar dan mengingat suatu informasi. Pengaturan bobot dilakukan secara terus-menerus dan dengan menggunakan kriteria tertentu sampai diperoleh keluaran yang diharapkan.

- Hal yang ingin dicapai dengan melatih/mengajari JST adalah untuk mencapai keseimbangan antara kemampuan memorisasi dan generalisasi.
- Kemampuan memorisasi = kemampuan JST untuk memanggil kembali secara sempurna sebuah pola yang telah dipelajari.
- Kemampuan generalisasi = adalah kemampuan JST untuk menghasilkan respon yang bisa diterima terhadap pola-pola input yang serupa. (namun tidak identik) dengan pola-pola yang sebelumnya telah dipelajari.
- Hal ini sangat bermanfaat bila pada suatu saat ke dalam JST diinputkan informasi baru yang belum pernah dipelajari, maka JST masih akan tetap dapat memberikan tanggapan yang baik, memberikan keluaran yang paling mendekati.
- Paradigma/metode pembelajaran/pelatihan JST :
  1. Pembelajaran terawasi (supervised learning)

Pada pembelajaran ini kumpulan input yang digunakan, output-outputnya telah diketahui. Perbedaan antara output-output aktual dengan output-output yang diinginkan digunakan untuk mengoreksi bobot JST agar JST dapat menghasilkan jawaban sedekat (semirip) mungkin dengan jawaban yang benar yang telah diketahui oleh JST.
  2. Pembelajaran tak terawasi (unsupervised learning / pembelajaran tanpa guru)

Pada pembelajaran ini, JST mengorganisasi dirinya sendiri untuk membentuk vektor-vektor input yang serupa tanpa menggunakan data atau contoh-contoh pelatihan. Struktur menggunakan dasar data atau korelasi antara, pola-pola data yang dieksplorasi. Paradigma pembelajaran ini mengorganisasi pola-pola ke dalam kategori-kategori berdasarkan korelasi yang ada.
  3. Gabungan pembelajaran terawasi dan tak terawasi (hybrid)

Merupakan kombinasi dari kedua pembelajaran tersebut. Sebagian dari bobot-bobotnya ditentukan melalui pembelajaran terawasi dan sebagian lainnya melalui pembelajaran tak terawasi.

#### FUNGSI AKTIVASI

- Dipakai untuk menentukan keluaran susu neuron
- Merupakan fungsi yang menggambarkan hubungan antara tingkat aktivasi internal (summation function) yang mungkin berbentuk linier atau nonlinear. Beberapa fungsi aktivasi JST diantaranya hard limit, purelin, dan sigmoid. Yang populer digunakan adalah fungsi sigmoid yang memiliki beberapa varian : sigmoid logaritma, sigmoid biner, sigmoid bipolar, sigmoid tangen.
- Hard limit memberikan batasan tegas 0 atau 1, purelin memisahkan

secara linier, sigmoid berupa fungsi smooth bernilai antara 0 sampai dengan 1 (bila biner) atau antara -1 sampai 1 (bila bipolar)

#### SUMMATION FUNCTION

- Fungsi yang digunakan untuk mencari rata-rata bobot dari semua elemen input
- Bentuk sederhananya adalah dengan mengalikan setiap nilai input ( $X_j$ ) dengan bobotnya ( $W_{ij}$ ) dan menjumlahkannya (disebut penjumlahan berbobot atau  $S_i$ )

$$S_i = \sum_{j=i}^N W_{ij} * X_j$$

- Diibaratkan dengan sebuah neuron yang memonitor sinyal yang datang dari neuron-neuron lain. Neuron ini menghitung penjumlahan berbobotnya dan kemudian menentukan sinyal untuk dikirim ke neuron-neuron lain

#### SUM SQUARE ERROR dan ROOT MEAN SQUARE ERROR

- Perhitungan kesalahan merupakan pengukuran bagaimana jaringan dapat belajar dengan baik sehingga jika dibandingkan dengan pola yang baru akan dengan mudah dikenali
- Kesalahan pada keluaran jaringan merupakan selisih antara keluaran sebenarnya (current output) dan keluaran yang diinginkan (desired output)
- Selisih yang dihasilkan antara keduanya biasanya ditentukan dengan cara dihitung menggunakan suatu persamaan
- Sum Square Error (SSE) :
  1. Hitung keluaran jaringan syaraf untuk masukan pertama
  2. Hitung selisih antara nilai keluaran jaringan syaraf dan nilai target/yang diinginkan untuk setiap keluaran
  3. Kuadratkan setiap keluaran kemudian dihitung seluruhnya

$$SSE = \sum_p \sum_j (T_{jp} - X_{jp})^2$$

$T_{jp}$  : nilai keluaran jaringan syaraf

$X_{jp}$  : nilai target yang diinginkan untuk setiap keluaran

- Root Mean Square Error (RMS Error) :
  1. Hitung SSE
  2. Hasilnya dibagi dengan perkalian antara banyaknya data pada pelatihan dan banyaknya keluaran, kemudian ditakarkan

$$RMSE_{Error} = \sqrt{\frac{\sum_p \sum_j (T_{jp} - X_{jp})^2}{n_p n_o}}$$

- $T_{jp}$  : nilai keluaran jaringan syaraf  
 $X_{jp}$  : nilai target yang diinginkan untuk setiap keluaran  
 $n_p$  : jumlah seluruh pola  
 $n_o$  : jumlah keluaran

keberhasilan suatu proses belajar JST ditunjukkan dengan besarnya error yang minimum. Pada kondisi inilah JST tersebut dapat digunakan. Ketika ada hal baru yang harus diketahui oleh JST maka proses belajar harus diulang kembali dengan menggunakan informasi-informasi yang lama ditambah dengan informasi-informasi baru.

#### APLIKASI JARINGAN SYARAT TIRUAN

- Aerospace  
Autopilot pesawat terbang, simulasi jalur penerbangan, system kendali pesawat, perbaikan autopilot, simulasi komponen pesawat
- Otomotif : system kendali otomatis mobil
- Keuangan dan perbankan  
Pendeteksian uang palsu, evaluator aplikasi kredit, pengidentifikasian pola-pola data pasar saham
- Militer  
Pengendali senjata, pendeteksi bom, penelusuran target, pembedaan objek, pengendali sensor, sonar, radar dan pengolahan sinyal citra yang meliputi kompresi data, ekstraksi bagian istimewa dan penghilangan derau, pengenalan sinyal citra.
- Elektronik  
Pembuatan perangkat keras yang bias mengimplementasikan JST secara efisien, machine vision, pengontrol gerakan dan penglihatan robot, sintesis suara
- Broadcast : pencarian klip berita melalui pengenalan wajah
- Keamanan : JST digunakan untuk mengenali mobil dan mengenali wajah oknum
- Medis : analisis sel kanker
- Pengenalan suara : pengenalan percakapan, klasifikasi suara
- Pengenalan tulisan : pengenalan tulisan tangan, penerjemahan tulisan kedalam tulisan latin
- Matematika : alat pemodelan masalah dimana bentuk eksplisit dari hubungan antara variabel-variabel tertentu tidak diketahui
- Pengenalan benda bergerak  
Selain pola dari citra diam, JST juga bisa digunakan untuk mendeteksi citra bergerak dari video seperti citra orang yang bergerak, dll
- JST digunakan sebagai detektor virus komputer, penginderaan bau, dll



## BAB IX

### ALGORITMA GENETIK

#### 69.1 Pendahuluan

Genetic Algorithm di usulkan pertama kali oleh John Holland dan teman-temannya di universitas Michigan untuk aplikasi seluler otomatis. Teknik ini menjadi populer diantara saintis dan rekayasawan di seluruh dunia untuk memecahkan masalah optimasi mereka.

Aplikasinya antara lain adalah :

- job shop scheduling,
- pembelajaran pengendali neuro-fuzzy,
- pemrosesan citra dan optimasi kombinatorial

#### **Konsep berfikir:**

GA mensimulasikan proses yang terjadi pada populasi alamiah yang merupakan hal penting dalam proses evolusi. Di alam, individu di populasi saling bersaing untuk memperoleh sumber daya seperti makanan, baju, tempat tinggal dan pekerjaan. Individu yang berhasil akan bertahan hidup sedangkan individu yang tidak, akan mati dan punah.

GA secara khusus dapat di terapkan untuk memecahkan masalah optimasi yang kompleks. Karena itu GA baik untuk aplikasi yang memerlukan strategi pemecahan masalah secara adaptif. GA secara inheren paralel karena pencarian pemecahan yang terbaik dilakukan melalui struktur genetik yang menyatakan sejumlah kemungkinan penyelesaian.

GA adalah salah satu teknik optimasi yang terkenal. Secara khusus ada 3 golongan teknik optimasi yaitu:

- Berdasarkan pada kalkulus
- Berdasarkan pada Enumeratif/Perulangan
- Berdasarkan pada Pencarian Acak Terarah

Setelah Holland memaparkan idenya tentang GA maka banyak peneliti yang mengusulkan berbagai variasi. Namun secara umum GA adalah teknik penanganan populasi.

*Ide awal GA mengadopsi kebijakan penggantian umum dimana keseluruhan populasi diganti setiap generasi. Di pihak lain kebijakan keadaan tunak (Steady state policy) menerapkan pergantian seleksi bagi populasi.* Hal ini lebih alami karena di alam sudah biasa orang tua dan anak hidup berdampingan pada saat yang bersamaan

Penyajian individu dari GA biasanya menggunakan sekumpulan bilangan biner. Tetapi penyajian biner ini memiliki beberapa kelemahan bila diaplikasikan pada masalah multidimensional dan persamaa numerik yang persisi.

Contoh:

Bila suatu masalah memiliki 100 variabel dengan ranah (-500 sd 500) memerlukan ketelitian 6 angka di belakang koma, maka akan diperlukan vektor solusi biner 3000. Itu berarti ada  $10^{1000}$  ruang pencarian yang diperlukan. Untuk persoalan tersebut GA memiliki kinerja yang buruk

Algoritma ini bekerja dengan sebuah populasi yang terdiri dari individu-individu yang masing-masing individu mempresentasikan sebuah solusi yang mungkin bagi persoalan yang ada. Dalam kaitan ini sebuah individu dilambangkan dengan nilai "FITNESS" yang akan digunakan untuk mencari nilai solusi yang terbaik dari persoalan yang ada.

Dalam teori genetika sebuah individu akan mengalami perkembang biakan, dimana pada saat perkembang biakan tersebut terdapat penurunan sifat kepada keturunannya (offspring). Keturunan ini dapat memiliki sifat gabungan dari kedua Parentnya. Sifat ini dapat dimiliki dengan mutasi atau crossover dari parentnya. Pada saat penurunan sifat maka akan terdapat individu baru yang akan di seleksi alam. Bila turunan tersebut mampu baik maka akan mampu bertahan, sebaliknya pula bila tidak maka akan musnah. Pada akhirnya kita akan mendapat keturunan yang terbaik. Nah individu yang terbaik tersebut adalah solusi optimal bagi permasalahan kita.

#### **Algoritma Dari Genetic Algorithm**

1. Membangun sebuah populasi yang terdiri atas beberapa gen atau direpresentasikan dengan String.
2. Evaluasi masing masing String (Fitness Value)
3. Proses Seleksi agar didapatkan string yang baik
4. Manupulasi genetika untuk mendapatkan individu yang baru dari string

## 9.2 Sejarah Genetic Algorithm

Teori darwin yang sempat membuat orang-orang berfikir bahwa manusia berasal dari kera, bahkan lebih rendah telah membuat membutakan kita sekitar abad 19 hingga beberapa tahun belakangan ini. Pada abad ke-19 banyak ilmuwan yang mencoba untuk membuktikan dan mensimulasikannya. Neo darwinisme yang menyebutkan bahwa sejarah kehidupan makhluk hidup adalah melalui suatu mekanisme proses statistika yang terjadi antara populasi dan spesies, yang dikenal dengan proses manipulasi genetika. Proses ini masing-masing adalah **reproduksi, mutasi, kompetisi dan pemilihan**.

Cikal bakal penggunaan GA untuk pencarian dalam sistem buatan di prakarsai oleh beberapa ahli biologi yang menggunakan komputer digital untuk mengerjakan simulasi dari sistem genetika. Diantara para ahli tersebut adalah:

1. Baricelli, N.A pada tahun 1957 melakukan penelitian tentang proses evolusi simbiogenetik yang direalisasikan dengan sistem artificial.
2. Baricelli, N.A pada tahun 1962 mengajukan teori evolusi dan analisis numeriknya
3. Fraser, A.S pada tahun 1960 menyimulasikan sistem genetika dengan komputer, yang meliputi aspek-aspek S-linkage, dominasi dan epistasis.

Meskipun penelitian –penelitian tersebut bertujuan untuk meneliti gejala alam namun yang mereka kerjakan secara kebetulan memiliki pemikiran paralel yang memunculkan ide tentang Genetic Algorithm.

Fraser mensimulasikan evolusi dari 15 bit Biner sebagai string generasi dan menghitung presentasi dari individu-individu yang terpilih oleh fenotip dengan generasi-generasi yang berurutan. Pada saat itu Fraser tidak menyebutkan dalam laporannya bahwa algoritma pencarian dalam gejala alam akan berguna dalam sistem buatan, namun ternyata hasil dari penemuannya ternyata menyerupai optimasi fungsi.

Hal itulah yang memberikan inspirasi bagi John Holland dan murid-muridnya untuk mengaplikasikan proses genetika ini pada sistem buatan. Holland menancapkan pondasi dalam karya tulisnya pada teori sistem adaptif yaitu:

1. Concern efficient adaptive systems(1962)
2. Information processing and processing systems(1962)
3. Outline for a logical theory of adaptive systems(1962)

Tahun 1962-1965 Holland mengajar tentang teori of adaptive sistem dan sering memberikan seminar-seminar tentang ini. Dalam masa itu penyempurnaan GA makin jelas. Selanjutnya dibuatlah rumus standar untuk GA ini.



### 9.3 Komponen Algoritma Genetik

#### Parameter Genetic Algorithm

Dalam GA ada beberapa parameter yang menjadi acuan yang standar dalam menentukan proses optimasi. Beberapa diantaranya adalah:

1. Ukuran populasi
2. Probabilitas CrossOver
3. Probabilitas Mutasi

#### Teorema Skema

##### Tahap1

Teorema skema merupakan dasar teori yang menjelaskan bagaimana Genetic algorithm bekerja. Skema adalah keserupaan pola dalam mendeskripsikan suatu himpunan bagian dari beberapa string yang mempunyai kesamaan pada posisi tertentu. Sebuah skema dibentuk dengan menambahkan sebuah simbol spesial, yaitu sebuah simbol \* (don't care) dalam representasi biner(0 atau 1).

Contoh: \*01 adalah 101 atau 001 yaitu 5 atau 1

##### Tahap2

Tingkatan dari sebuah skema  $S$  (dinotasikan dengan  $o(S)$ ) menunjukkan jumlah dari posisi angka 0 atau 1 yang sudah tetap (Bukan posisi don't care) yang ada dalam skema. Tingkatan ini menunjukkan spesialisasi dari sebuah skema. Contoh:

$$S1 = (* * * 0 0 1 * 1 1 0)$$

$$S2 = (* * * * 0 0 * * 0 *)$$

$$S3 = (1 1 1 0 1 * * 0 0 1)$$

Dimana

$$o(S1) = 6$$

$$o(S2) = 3$$

$$o(S3) = 8$$

##### Tahap3

Batasan panjang dari skema S (dinotasikan dengan  $\delta(S)$ ) adalah jarak antara posisi angka 0 atau 1 yang pertama hingga terakhir. Angka ini menunjukkan kerapatan informasi yang ada dalam sebuah skema. Contoh:

$$\delta(S_1) = 10 - 4 = 6;$$

$$\delta(S_2) = 9 - 5 = 4;$$

$$\delta(S_3) = 10 - 1 = 9;$$

### **Mekanisme Genetic Algorithm**

Mekanisme yang ada dalam Genetic Algorithm adalah sangat sederhana. Yaitu hanya melibatkan penyalinan string dan pertukaran bagian string.

Siklus perkembangbiakan GA diawali dengan pembuatan himpunan solusi secara acak yang dinamakan populasi. Dimana didalamnya terdapat individu-individu yang dinamakan kromosom. Kromosom ini secara lambat laun mengalami iterasi 'Perkembangbiakan' dalam sebuah generasi. Selama dalam sebuah generasi kromosom-kromosom ini di evaluasi dengan menggunakan rumus yang telah ditentukan dengan rumus fitness.

Untuk menciptakan generasi berikutnya dengan kromosom yang baru (offspring) dapat dilakukan dengan menggabungkan dua potongan kromosom yang telah didapatkan dengan operator crossover atau mutasi. Sebuah generasi baru sebelum dievaluasi lagi, maka dia melalui proses seleksi berdasarkan fungsi fitnessnya. Dari kreasi ini kromosom-kromosom yang paling fit mempunyai kemungkinan besar untuk diseleksi.

Setelah beberapa generasi maka algoritma ini akan mengalami konvergen pada kromosom terbaik yang merupakan nilai optimum dari permasalahan yang diselesaikan.

### **9.4 PROSES GENETIC ALGORITHM**

Langkah pertama dalam pemecahan masalah dengan GA adalah pengkodean terhadap masalah yang akan kita pecahkan. Pada algoritma ini dalam mengasumsikan sebuah solusi untuk sebuah persoalan dimungkinkan dengan diwakili oleh satu set parameter. Parameter ini dinamakan gen. Berisi nilai-nilai yang bersatu membentuk string(Kromosom).

Selanjutnya kromosom yang berkumpul membentuk populasi. Dari sebuah populasi inilah GA memulai untuk pencarian.

Dalam GA fungsi Fitness harus dirancang sesuai dengan masing-masing masalah yang akan diselesaikan. Misalnya saja untuk masalah mencari rute terpendek dari suatu perjalanan, maka fungsi Fitnessnya adalah jarak tempuh. Sedangkan untuk optimasi penjadwalan kuliah fungsi Fitnessnya melibatkan faktor-faktor seperti adanya dua kuliah atau lebih yang akan diajarkan dalam satu kelas, kepadatan dosen mengajar setiap hari, penempatan kuliah dan praktikum tidak semestinya.

Sebagai contoh untuk pengkodean jika kita mempunyai masalah yaitu mencari maximum sebuah fungsi dari 3 variabel( $x,y,z$ ) dan akan menampilkan masing-masing variabel dengan 6 bit. Dari contoh ini kita melakukan pengkodean dengan membentuk kromosom yang terdiri dari 3 gen yang masing-masing gen terdiri dari 6 bit sehingga sebuah kromosom terdiri dari 18 bit.

Suatu hal yang sangat mendasar dari GA adalah bahwa algoritma ini bekerja pada daerah pengkodean dan daerah solusi. Operasi genetika (CrossOver dan Mutasi) bekerja pada daerah pengkodean, sedang proses evaluasi dan proses seleksi bekerja pada daerah solusi.

Dalam representasi nonBiner, ada dua hal yang perlu diperhatikan untuk pengkodean baik dalam fenotip ataupun genetik. Yaitu:

1. **Feasibilitas dan legalitas dari sebuah kromosom**
2. **Pemetaan**

Feasibilitas adalah apakah solusi/kromosom berada pada daerah yang feasible dari domain permasalahan. InFeasibilitas kromosom menghilangkan kealamian batasan-batasan dalam permasalahan optimasi. Legalitas adalah apakah suatu solusi/kromosom dapat menyelesaikan masalah

Untuk beberapa permasalahan kasus metode pinalti telah diajukan untuk mengatasi kasus kromosom yang tidak dimungkinkan. Dalam permasalahan kasus optimasi nilai optimum secara tipikal akan berada di wilayah antara daerah yang mungkin ataupun yang tidak mungkin. Penerapan metode pinalti akan mengeksploitasi pencarian GA pada nilai optimum di dua daerah tersebut.

Kromosom-kromosom yang tidak legal menghilangkan kealamian dari teknik encoding. Untuk beberapa permasalahan kombinatorial, beberapa cara digunakan untuk menghindari keturunan yang tidak legal karena tidak dapat mewakili suatu solusi, dalam arti kromosom tersebut tidak dapat di evaluasi. Salah satu cara, misalnya PMX operator. Metode ini digunakan untuk mencegah dihasilkannya kromosom yang tidak mungkin dan tidak ilegal.

Pemetaan dari kromosom-kromosom ke solusi memungkinkan satu dari tiga kasus ini terjadi:

1. Pemetaan dari 1 ke  $-n$
2. Pemetaan dari 1 ke 1
3. Pemetaan dari  $-n$  ke 1

Pemetaan 1 ke 1 merupakan pemetaan terbaik, karena dari pemetaan ini satu kromosom dapat dipasangkan dengan satu solusi. Pada pemetaan  $-n$  ke 1, sejumlah  $-n$  kromosom mewakili satu solusi yang sama, sedangkan pemetaan 1 ke  $-n$  adalah pemetaan yang tidak diinginkan karena merelasikan satu kromosom dengan banyak solusi, yang menjelaskan banyak solusi sehingga menjadi tidak jelas solusi mana yang sebenarnya yang ditunjukkan.

### **9.5 OPERASI DALAM GENETIC ALGORITHM**

Dengan mekanisme GA seperti di jelaskan pada artikel sebelumnya, setelah dilakukan pendefinisian terhadap kode GA selanjutnya adalah dilakukan operasi-operasi tertentu yang melibatkan 2 faktor utama yaitu:

1. Operator seleksi yang melibatkan proses seleksi didalamnya.
2. Operasi genetika yang melibatkan Crossover dan mutasi

#### **Fungsi Evaluasi**

Disamping representasi, fungsi evaluasi juga merupakan masalah yang penting dalam dalam GA. Fungsi Evaluasi yang baik harus mampu memberikan nilai fitness yang sesuai dengan kinerja kromosom.

Pada permukaan optimasi, biasanya nilai fitness masing-masing individu masih mempunyai rentang yang besar. Seiring dengan bertambahnya generasi, beberapa kromosom mendominasi populasi dan mengakibatkan konvergensi dini.

Problem klasik dalam GA adalah beberapa kromosom dengan nilai fitness yang tinggi (tetapi bukan nilai optimum) mendominasi populasi dan mengakibatkan GA konvergen pada local optima. Ketika populasi konvergen, Kemampuan GA untuk mencari solusi yang baik hilang, Crossover antara kromosom yang hampir identik menghasilkan offspring yang identik. Dalam kondisi ini hanya operasi mutasi yang mampu menghasilkan kromosom yang relatif baru dan merupakan cara untuk menghindari kromosom super untuk mendominasi populasi.

### **Fungsi Seleksi**

Pada proses evaluasi GA, keragaman populasi dan selection Pressure memegang peranan penting. Keduanya berkaitan erat. Meningkatnya tekanan seleksi akan berakibat pada minimnya keragaman populasi, sebaliknya tekanan seleksi yang terlalu longgar akan mengakibatkan banyaknya ragam dari kromosom sehingga pencarian tidak efisien.

---