

## 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.

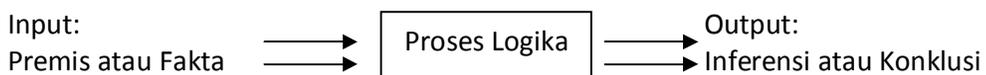
Karakteristik representasi pengetahuan

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 :

- a. Konjungsi :  $\wedge$  (and)
- b. Disjungsi :  $\vee$  (or)
- c. Negasi :  $\neg$ , (not)
- d. Implikasi :  $\rightarrow$  (if then)
- e. 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.

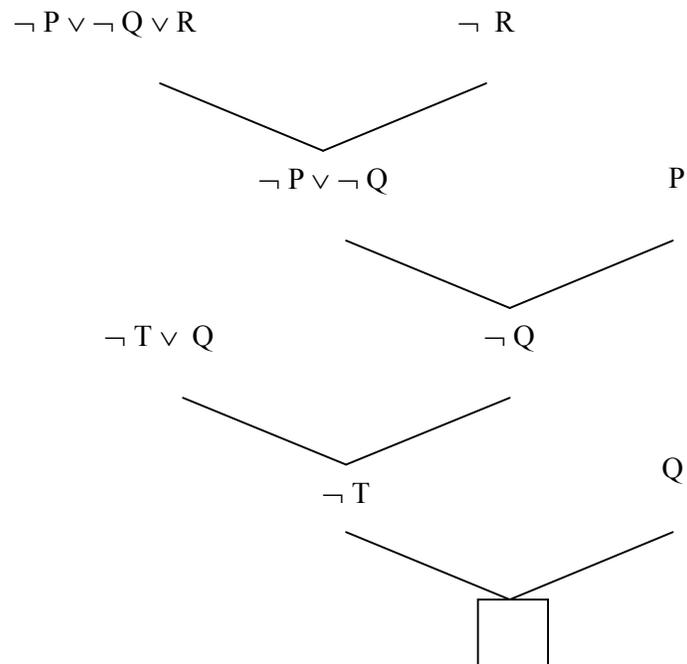
Untuk membuktikan kebenaran R dengan menggunakan resolusi, maka ubah dulu menjadi bentuk CNF.

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 :

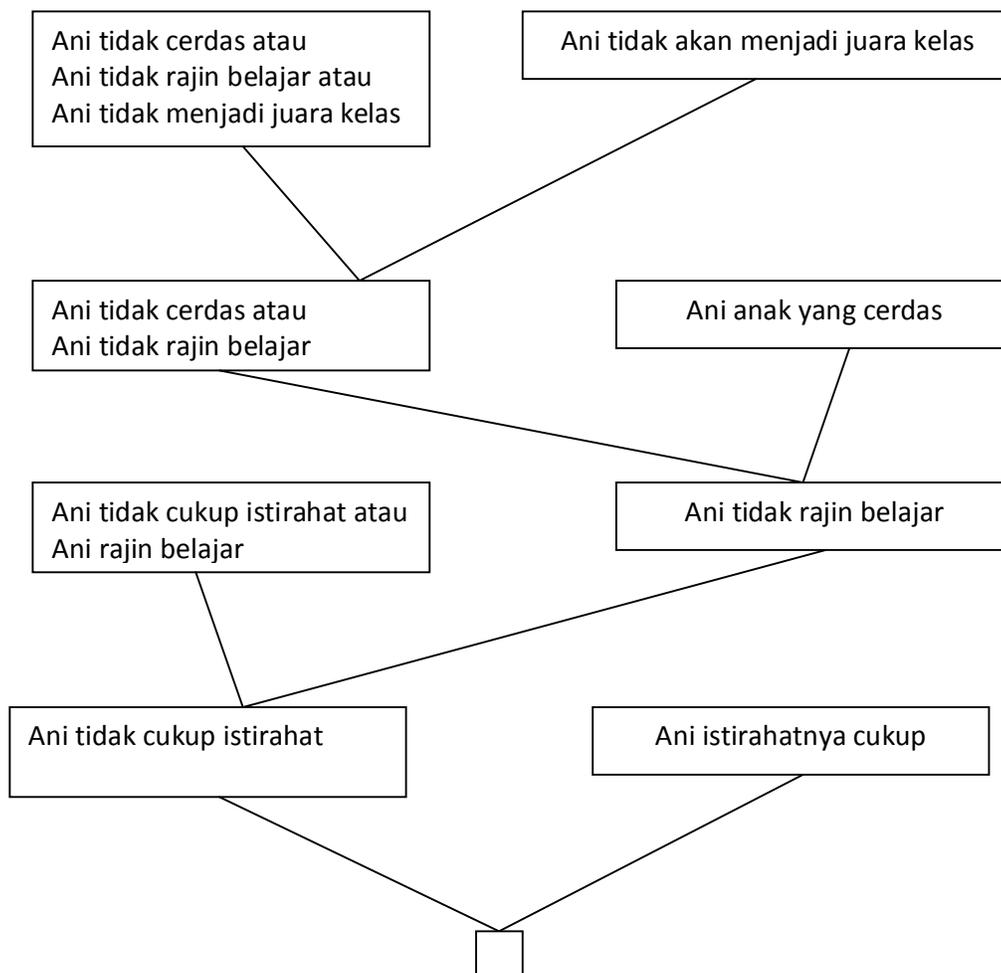
- Ani anak yang cerdas
- Jika ani anak yang cerdas dan ani rajin belajar, maka ani akan menjadi juara kelas
- Jika ani makannya banyak atau ani istirahatnya cukup, maka ani rajin belajar
- Ani istirahatnya cukup

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  
 $\neg$  cuaca(hujan, besok)  $\rightarrow$  pergi (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 : suka (x, y)  $\wedge$  suka (z, y)  $\rightarrow$  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.

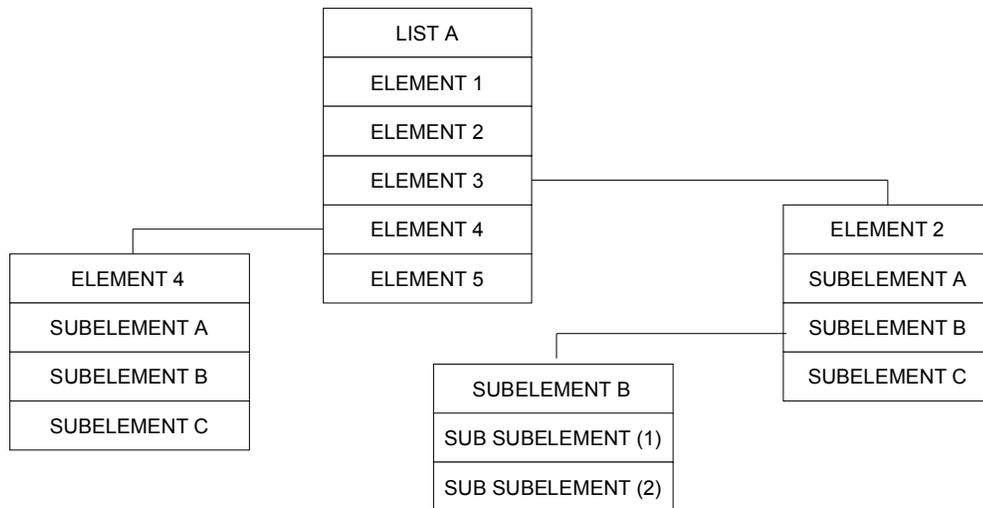
#### 4.2 LIST dan TREE

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.

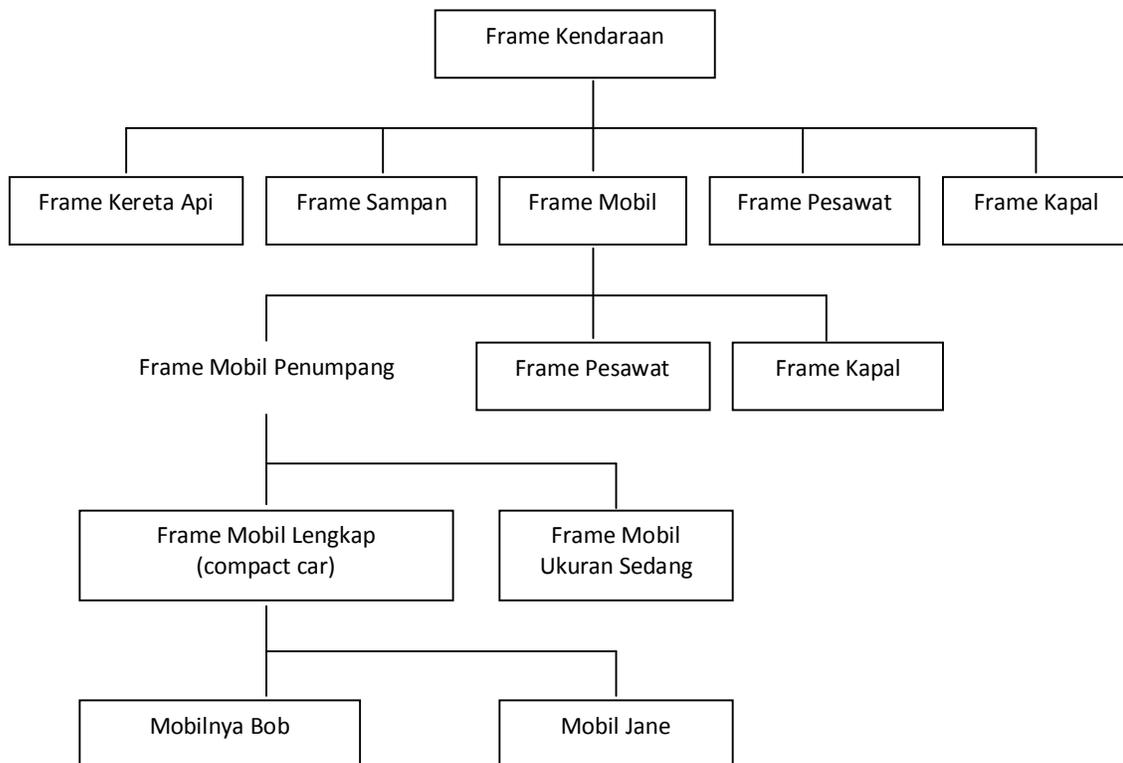
Frame Mobil
Class : Transportasi
Nama pabrik : Audi
Negara : Jerman
Model : 5000
Tipe : Sedan
Bobot : 3300 lb
Ukuran dasur roda : 105,8 inchi
Jumlah Pintu: 4 (default)
Transmisi : 3-speed otomatis
Jumlah roda : 4 (default)
Mesin : (referensi kerangka mesin)
<ul style="list-style-type: none"> <li>• Tipe : 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.

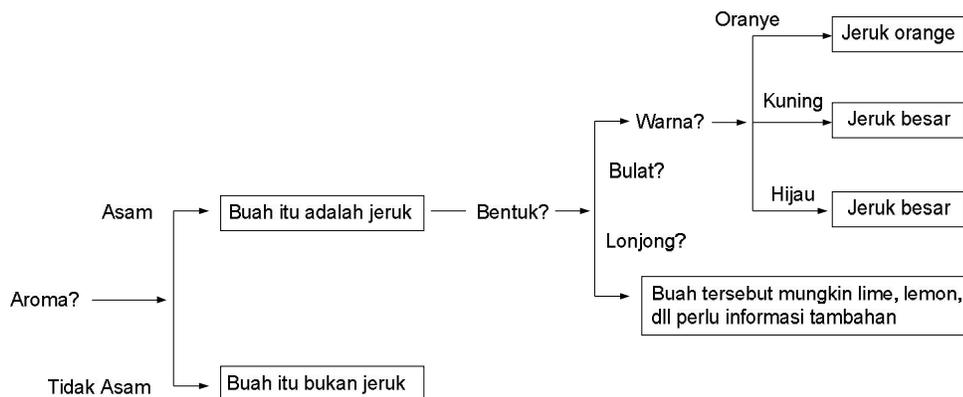
Hirarki memungkinkan pewarisan sifat-sifat. Setiap frame biasanya mewarisi sifat-sifat dari frame dengan level yang lebih tinggi. Pewarisan merupakan mekanisme untuk membentuk pengetahuan, yang menyediakan nilai slot, dari frame ke frame.

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.

