

HO-03 Kecerdasan Buatan

Representasi Pengetahuan

Pendekatan Objek

Opim S Sitompul

Erna Budhiarti Nababan





Pendahuluan

- Kemampuan terpenting dalam kecerdasan adalah kemampuan belajar (beradaptasi dengan situasi baru)
- Agar proses pembelajaran dapat berlangsung diperlukan pengetahuan terstruktur
- Kemampuan inilah yg akan diterapkan pada komputer: kemampuan belajar (dg menyediakan pengetahuan yang terstruktur)



Perwakilan Pengetahuan

- Pengetahuan dalam AI mencakup:
 - Definisi pengetahuan
 - Jenis pengetahuan dan hirarki
 - Perwakilan pengetahuan
 - Pendekatan obyek (object)
 - Pendekatan logika (logic)
 - Pendekatan aturan (rules)

Pendekatan Obyek

- Beberapa metode pendekatan objek yang digunakan untuk mewakilkan pengetahuan dalam bidang Kecerdasan Buatan (AI)
 - OAV (Object-Attribute-Value),
 - Semantic,
 - Frame dan Script.
 - Pohon keputusan



1. Teknik OAV (*Object-Attribute-Value*)

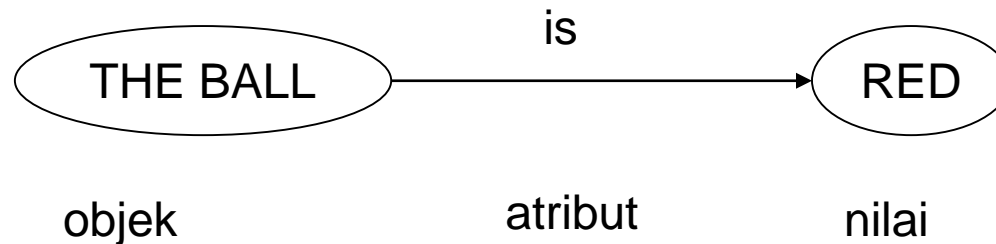
- OAV digunakan untuk mewakilkan nilai dari ciri-ciri sesuatu objek.





OAV (*Object-Attribute Value*)

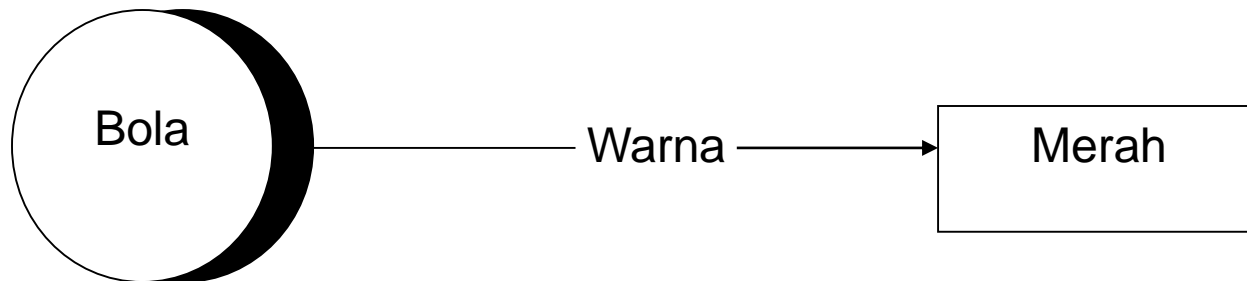
- Objek – fisik atau konsep
- Atribut – ciri-ciri objek
- Nilai – pengukuran/nilai spesifik bagi atribut



OAV (*Object-Attribute Value*)

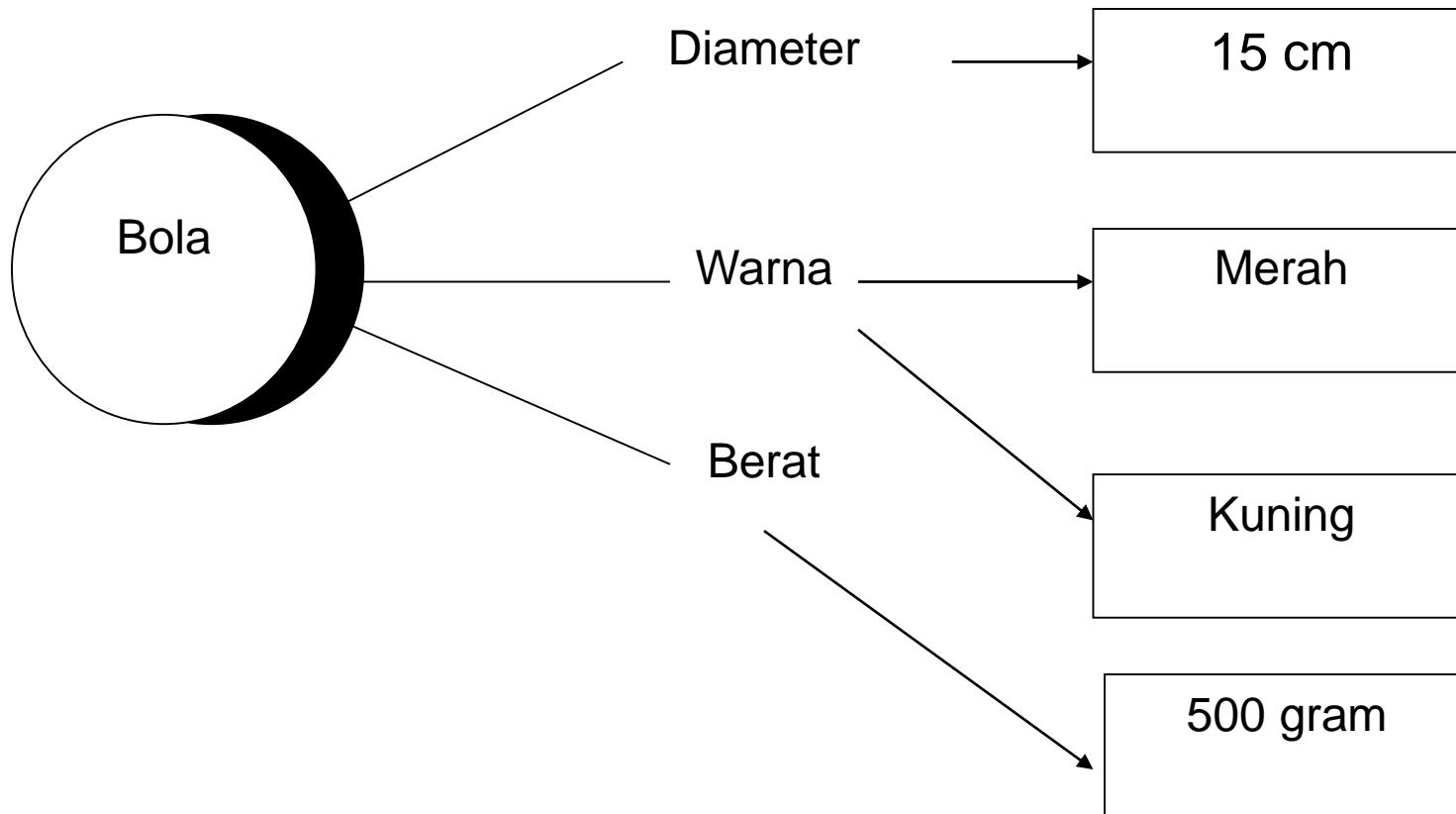


- Kalimat di atas dibagi atas 3 bagian yaitu: objek, ciri dan nilai ciri. Perwakilan OAV untuk kalimat di atas adalah seperti pada gambar:



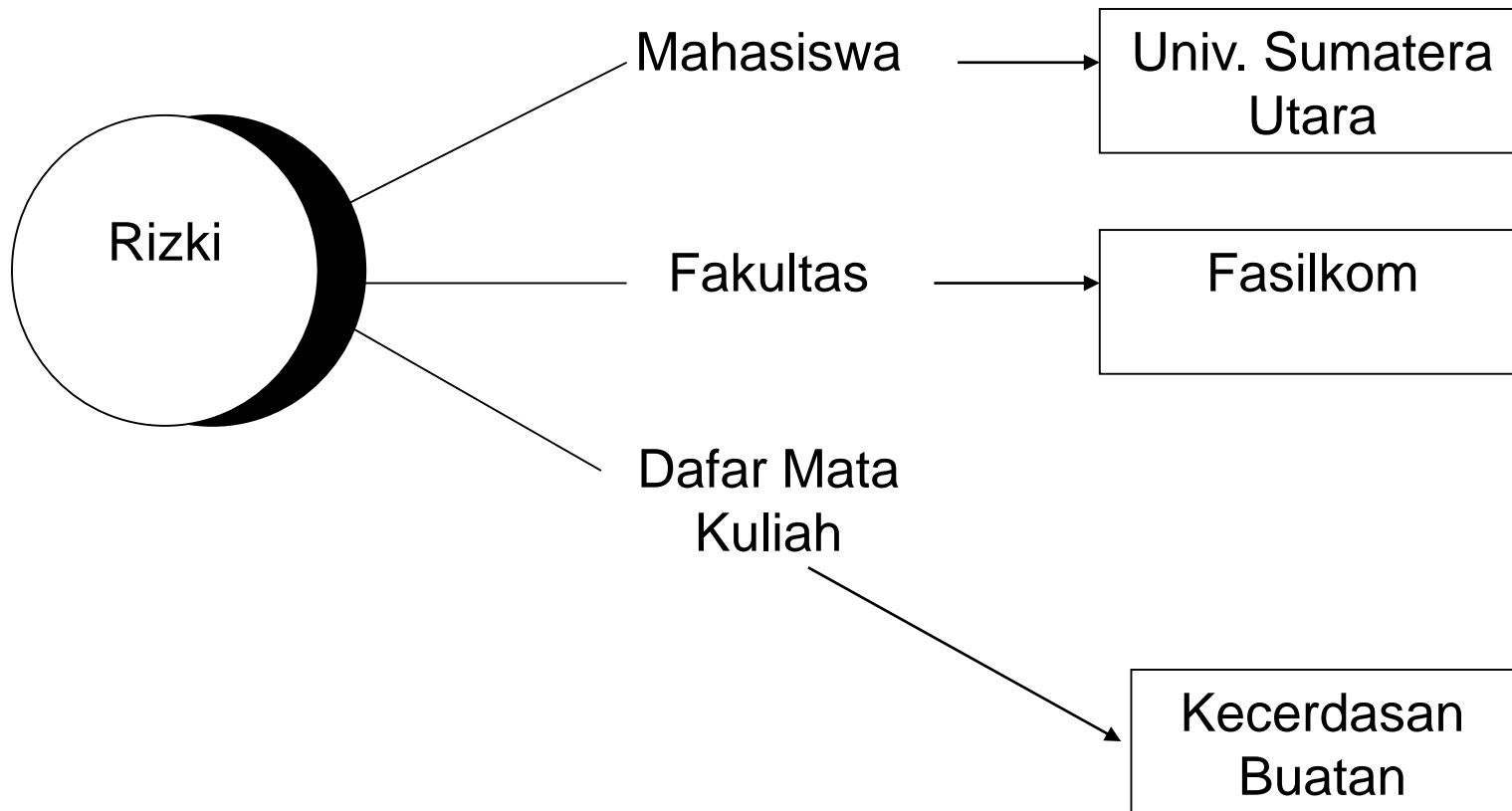
Gambar 1
Struktur OAV Satu-Ciri untuk Bola

Struktur OAV dengan Banyak Ciri



Gambar.2
Struktur OAV bola dengan banyak ciri

Struktur OAV Dengan bermacam ciri/atribut



Gb 3
Struktur OAV dg berbagai-Ciri untuk Rizki



Contoh lain perwakilan dgn OAV

Object	Attributes	Values
House	Bedrooms	2,3,4 etc
House	Color	Green, white, brown
Admission to a University	Grade Point Average	3.0, 3.5, 3.7, etc
Inventory Control	Level of Inventory	15, 20, 30, etc
Bedrooms	Size	9x10', 6x12', etc



- Contoh lain:

Object	Attribute	Value
apple	color	red
apple	type	fuji
apple	quantity	100
anggur	color	red
anggur	type	seedless
anggur	quantity	500

- OAV sangat berguna terutama dalam perwakilan fakta dan pola untuk mencocokkan fakta dalam antecedent sebuah rule.

2. Jaringan Semantik (Semantic Network)



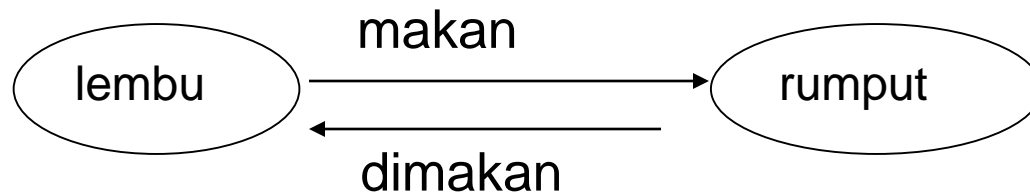
- Jaringan semantik merupakan penggambaran grafis dari pengetahuan yang melibatkan hubungan antara obyek-obyek.
- Obyek direpresentasikan sebagai simpul (*node*) pada suatu grafik berbentuk lingkaran dan hubungan antara obyek-obyek dan faktor deskriptif dinyatakan oleh garis penghubung (*link*) atau garis lengkung (*arch*) berlabel.



- Obyek dapat berupa:
 - jenis **fisik** spt. buku, mobil, meja, atau bahkan orang;
 - merupakan **pikiran**, seperti: hukum Ohm;
 - suatu **peristiwa/kejadian**, seperti: piknik, pesta
 - suatu **tindakan**, seperti: membuat rumah atau menulis buku.
 - suatu **perasaan**, seperti: marah, gembira



Node dan Arc dalam Semantic Net



NODE – mewakili unsur domain

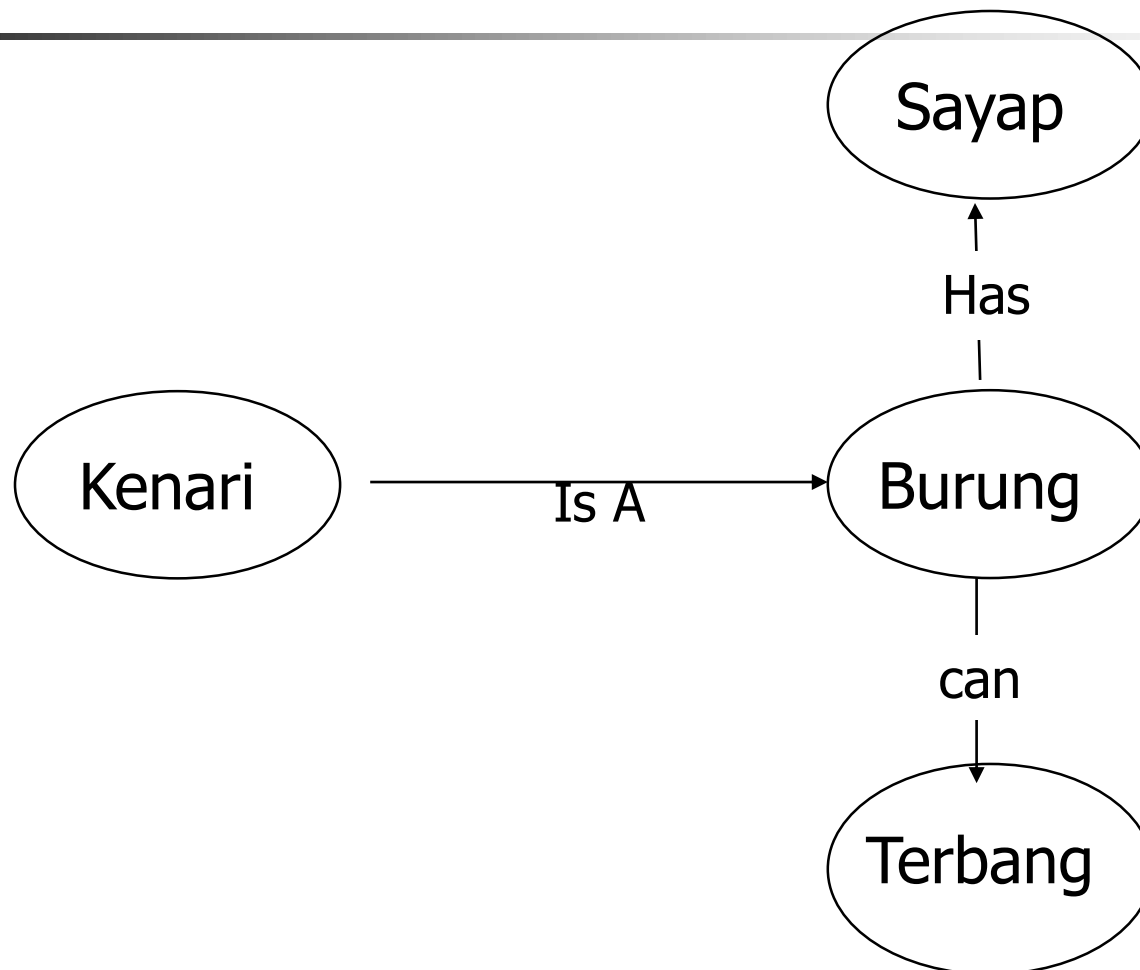
ARC/LINK – mewakili hubungan antara unsur

Hubungan yang sering digunakan

- IS-A (is an instance)
- A K O (A Kind Of)
- is_part_of (sebahagian drp)



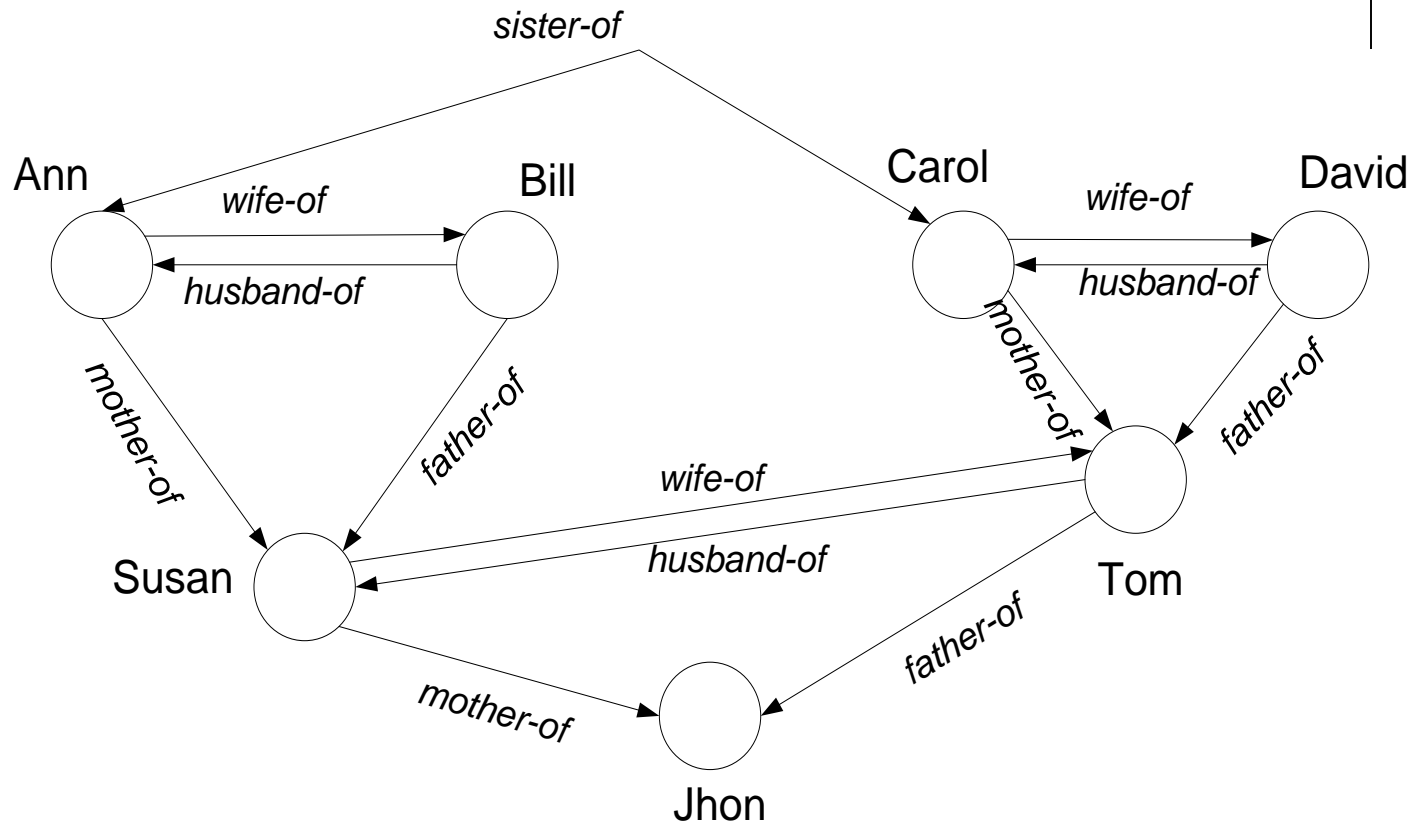
contoh:



Gb.4 Semantic Net dengan hubungan IS-A)



- Atribut objek, seperti: ukuran, warna, kelas, umur, asal-usul, atau karakteristik lainnya bisa digunakan sebagai node.
- Contoh penerapan jaringan semantik yang mudah ditemui adalah struktur kepegawaian dan garis keturunan.



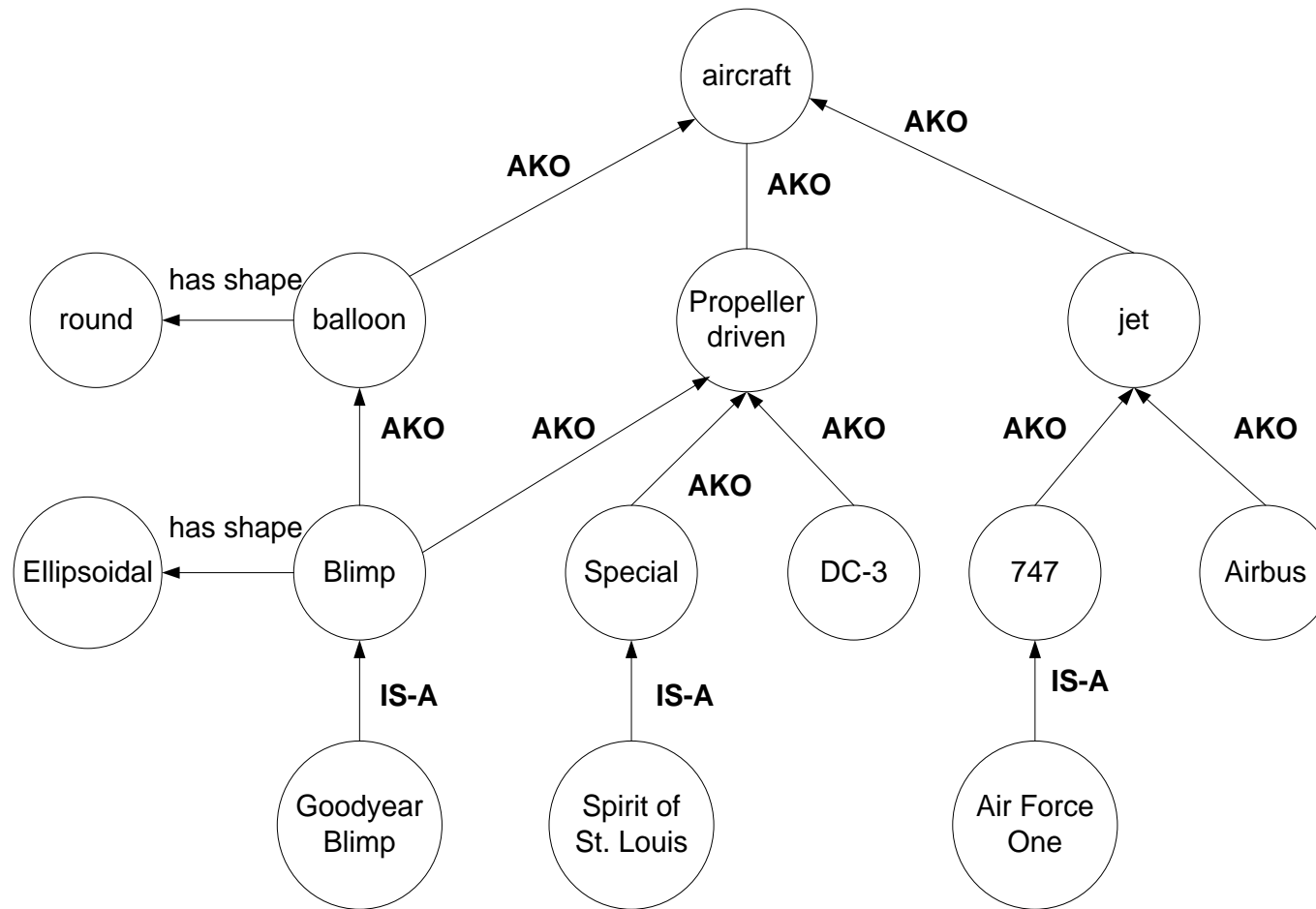
Gambar 6. Semantic Net Garis Keturunan



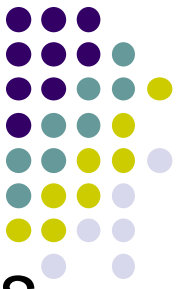
- Semantic net sering juga disebut **associative net** karena nodes berhubungan dengan node-node yang lain.
Jika node lain menerima aktivasi yang cukup, konsep akan mempercepatnya untuk menjadi ingatan sadar (conscious mind)
Contoh: meskipun anda mengetahui beribu kata, anda hanya memikirkan kata-kata tertentu dalam kalimat yang anda baca

Contoh

Semantic Net with IS-A and AKO



Gb. 7 Semantic Net dengan hubungan IS-A dan A-Kind-Of (AKO)



IS-A artinya 'is an instance of' yang merujuk kepada anggota tertentu dari suatu *class*

Class: berhubungan dengan konsep matematik *set* (himpunan) suatu obyek.

Himpunan dapat terdiri dari elemen yang berjenis apa saja, obyek dalam suatu class mempunyai hubungan satu dengan lainnya



Perhatikan himpunan berikut:

$\{3, \text{egg}, \text{blue}, \text{tires}, \text{art}\}$

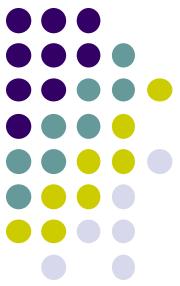
Anggota dalam himpunan ini tidak mempunyai hubungan yang umum.

Plane, train, automobiles mempunyai hubungan karena ketiganya merupakan alat transportasi.



AKO digunakan untuk menghubungkan satu *class* ke *class* lainnya.

- **AKO** tidak digunakan untuk menyatakan hubungan spesifik antar individu.
- **AKO** menghubungkan *class* individu ke *parent class* dimana individu tsb merupakan *child class* nya



- **AKO** menghubungkan **generic** node ke **generic** node lainnya

IS-A menghubungkan **hal/contoh** atau **individual** ke generic class

Dari gambar yang diberikan tadi, dapat dilihat *class* yang lebih umum (general) ada di bagian atas dan *class* yang lebih khusus berada di bagian bawah. **Class** yang lebih umum yang ditunjuk oleh **AKO** disebut **superclass**



Attributes, value, property

- Umumnya, object dalam suatu class mempunyai satu atau lebih **attributes**.
- Setiap attribute mempunyai **value**.
- Kombinasi/gabungan dari **attribute** dan **value** disebut **property**



Contoh:

- A blimp has attributes size, weight, shape and color.
- The value of the shape attributes is ellipsoidal

In other words, a blimp has the property of an ellipsoidal shape.



- Hubungan **IS-A** menyatakan **value**

Contoh:

Whatever aircraft the President is on Air Force One.

If President is on a helicopter, then Air Force One **IS-A** helicopter.



- Since the Goodyear Blimp is a blimps have ellipsoidal shape, it follows that the Goodyear blimp is ellipsoidal.
- The duplication of one's node characteristics by a descendent is called **inheritance**.
- All members of a class will inherit all the properties of their superclasses.
- Inheritance is a useful tool in knowledge representation because it eliminates the need to repeat common characteristics.



Object-Attribute-Value Triples (OAV)

Salah satu permasalahan dalam menggunakan semantic nets adalah tidak adanya definisi standar untuk nama *link*.

Contoh, sebagian buku menggunakan IS-A untuk *generic* dan *individual relation*.

Ketiga item, object, attribute dan value, sangat sering dijumpai, sehingga OAV dapat digunakan untuk menentukan karakteristik semua pengetahuan dalam semantic net.

PROLOG dan Semantic Net



Semantic net mudah ditranslasikan ke dalam PROLOG.

Contoh: (di ambil dari semantic net gb. 1)

is_a(goodyear_blimp,blimp).

is_a(spirit_of_st_louis, special).

has_shape(blimp, ellipsoidal).

has_shape(balloon, round).

Statement di atas adalah PROLOG **predicate expression** atau **predicate**.



Dalam PROLOG ekspresi predikat terdiri dari predicate name, seperti **is_a**, diikuti dengan zero atau lebih argument didalam tanda kurung (parentheses) dan dipisahkan oleh koma.

Contoh:

color (red)	; red is a color is a fact
mother(pat,ann)	; pat is the mother of ann
parents (jim, ann, tom)	; jim and ann are parents of tom
surrogate mother(pat,tom)	; pat is surrogate mother of tom

Predicate juga dapat dinyatakan dengan hubungan seperti IS-A dan HAS-A



`is_a(red,color).`

`has_a(john, father).`

`has_a(john, mother).`

`has_a(john,parents).`

Predikat HAS-A tidak menunjukkan arti yang sama seperti contoh sebelumnya.



Untuk memberi nama, maka harus diberi predikat tambahan

`is_a(tom, father).`

`is_a(susan, mother).`

`is_a(tom, parent).`

`is_a(susan, parent).`

Penambahan predikat ini tidak menunjukkan arti berbeda dg sebelumnya. John has a father dan Tom is a father tapi tdk ada informasi yg menyatakan bahwa Tom adalah ayah dari John



Program Prolog

Program Prolog terdiri dari *fact* dan *rules* untuk membentuk suatu *goal*.

$$p:-p_1, p_2, \dots, p_N.$$

Dimana p disebut the rule's head dan p_k adalah subgoal. Dinyatakan dalam kalimat Horn yang menyatakan:

head goal, p , akan dipenuhi jika dan hanya jika semua subgoal dipenuhi.



Simbol :- diterjemahkan sebagai **IF**.

Jika hanya terdapat head dan tidak ada keterangan disebelah kanannya, seperti pada:

$p.$

maka head bernilai true.

Sedangkan tanda comma yang memisahkan subgoal adalah untuk menyatakan logika AND



- Aturan-aturan PROLOG memerlukan IF karena merupakan conditional conclusion yang nilai kebenarannya tergantung dari satu atau lebih kondisi.

`parent(X,Y) :- father (X,Y).`

`parent(X,Y) :- mother(X,Y).`

Yang berarti bahwa X adalah parent dari Y jika X adalah father dari Y atau X adalah mother dari Y.

$\text{grandparent}(X, Y) \text{ :- parent } (X, Z), \text{ parent } (Z, Y)$



dan ancestor dapat dinyatakan dengan:

(1) $\text{ancestor } (X, Y) \text{ :- parent } (X, Y)$

(2) $\text{ancestor } (X, Y) \text{ :- ancestor } (X, Z), \text{ ancestor } (Z, Y)$



3. Skema: *Frame* dan *Script*

- Skema adalah metoda pengorganisasian, presentasi dan penggunaan pengetahuan tiruan (stereotype) agar komputer bisa menalar.
- Terdiri dari:
 - a. Skema Kerangka / Bingkai (Frame)
 - b. Naskah (Script)



a. *Frame*

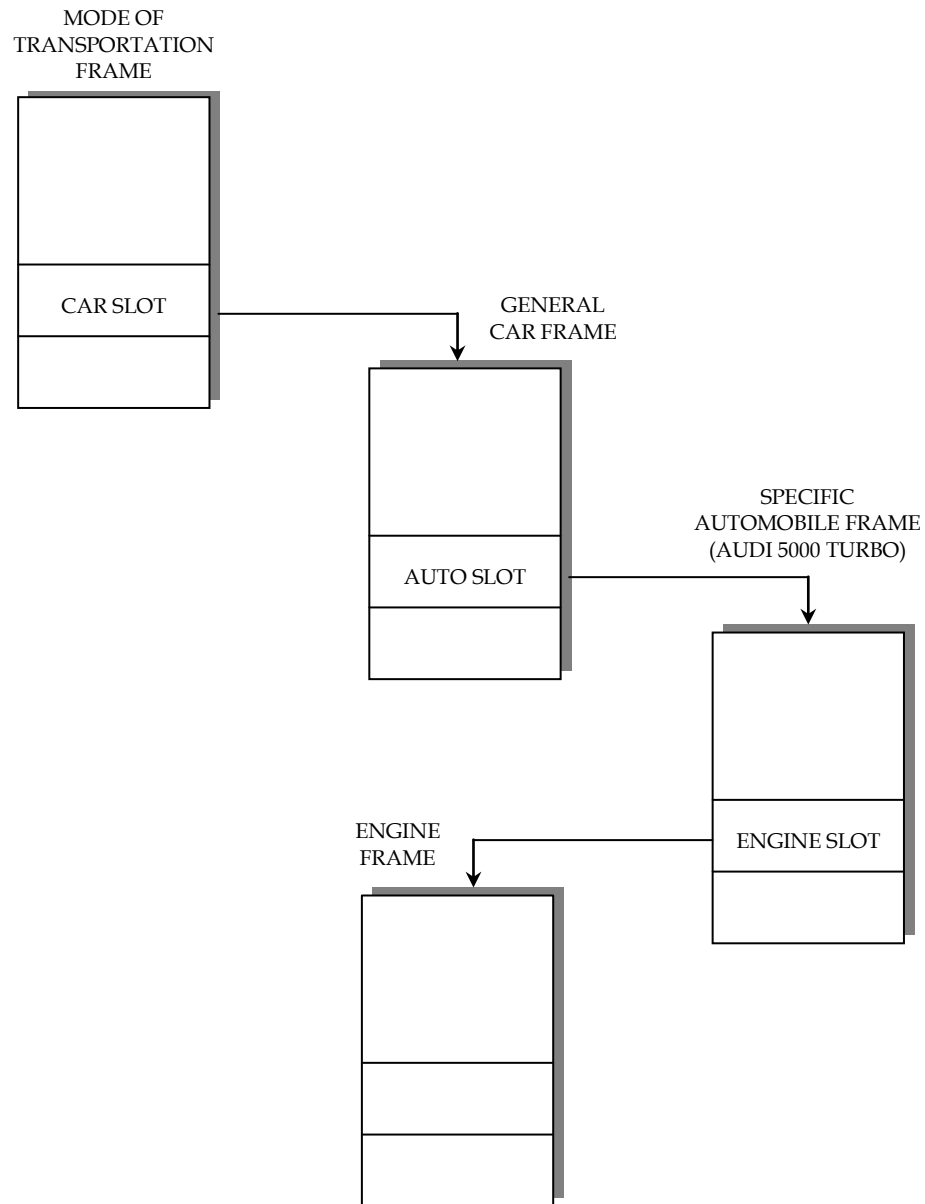
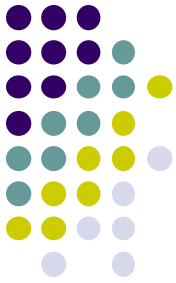
- *Frame*: blok-blok atau potongan-potongan yang berisi pengetahuan mengenai obyek-obyek khusus, kejadian, lokasi, situasi ataupun elemen-elemen lainnya dengan ukuran yang relatif besar.
- Blok-blok ini menggambarkan obyek-obyek dg sangat rinci.
- Detail diberikan dalam bentuk rak (*slot*) yg menggambarkan berbagai atribut (*attribute*) dan karakteristik dari obyek tersebut.



- Sebuah slot dapat berisi nilai default, yaitu nilai yang sudah melekat dan menjadi ciri dari suatu obyek. Misalnya: bingkai pengetahuan slot mobil sedan, memiliki nilai untuk slot jumlah ban otomatis 4 buah.
- Sebuah slot juga dapat mempunyai nilai *procedural attachment*, yaitu suatu nilai yang besarnya relatif. Akselerasi mesin misalnya, mempunyai nilai yang relatif terhadap waktu. Contohnya akselerasi suatu mesin memiliki nilai 0 – 60 km/jam dalam waktu 4 detik. Dalam waktu 2 detik, mesin tersebut memiliki akselerasi yang berbeda.



- Sebuah slot dapat berisi bingkai, dimana bingkai ini juga tersusun atas slot-slot. Misalnya slot mesin berisi sebuah bingkai mesin yang terdiri atas slot-slot rasio kompresi, sistem pengapian, besarnya daya dan besarnya torsi.





Diperoleh kesimpulan,

BINGKAI MOBIL	
Kelas	: Transportasi
Pabrik	: Audi
Negara asal	: Jerman
Model	: 5000 Turbo
Tipe	: Sedan
Berat	: 3300 lb
Jumlah roda	: 4
BINGKAI MESIN	
Ukuran Silinder	: 3.19 inch
Rasio Kompresi	: 7.9 – 1
Tenaga	: 140 HP

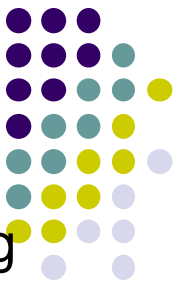


b. Script

- Naskah menggambarkan urutan peristiwa. Naskah biasanya direpresentasikan ke dalam konteks tertentu.
- Dalam menggambarkan urutan peristiwa, naskah menggunakan serangkaian slot yang berisi informasi tentang orang, obyek, dan tindakan-tindakan yang terjadi dalam suatu peristiwa.



- Beberapa elemen naskah yang tipikal meliputi: **kondisi masukan (*entry condition* / *descriptor*)**, **jalur (*track*)**, **peran (*role*)**, **pendukung (*prop*)**, dan **adegan (*scene*)**.



- *Kondisi masukan* menggambarkan situasi (dunia sekitar) yang harus dipenuhi sebelum terjadi atau berlaku suatu peristiwa yang ada dalam naskah agar skrip dapat dipanggil.
- *Jalur* mengacu kepada variasi yang mungkin terjadi dalam naskah tertentu.
- *Pendukung* mengacu kepada obyek yang digunakan dalam urutan peristiwa yang terjadi.
- *Peran* mengacu kepada orang-orang terlibat dalam naskah. Merupakan tindakan yang dilakukan oleh partisipan individual.
- *Adegan* merupakan kejadian yang menunjukkan aspek waktu dari skrip, menggambarkan urutan peristiwa aktual yang terjadi.
- Dan akhirnya, ***hasil (output)*** adalah kondisi yang ada (fakta yang benar) sesudah peristiwa dalam naskah berlangsung.



NASKAH RESTORAN

Jalur : Restoran swalayan (*fast food*)
Peran : Tamu, Pelayan
Pendukung : Counter, baki, makanan, uang, serbet, garam, merica, kecap, sedotan, dll.
Kondisi Masukan : Tamu lapar - Tamu punya uang.

Adegan_1 : Masuk

- ☺ Tamu parkir mobil
- ☺ Tamu masuk restoran
- ☺ Tamu antri
- ☺ Tamu baca menu dan mengambil keputusan tentang apa yang dipesan



Adegan_2 : Pesanan

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

Adegan_3 : Makan

- ☺ Tamu mengambil serbet, sedotan, garam, dll.
- ☺ Tamu membawa baki makanan ke meja kosong
- ☺ Tamu makan dengan cepat



Adegan_3a (option) : Membawa pulang

😊 Tamu membawa makanan dan pulang

Adegan_4 : Pulang

😊 Tamu membereskan meja

😊 Tamu membuang sampah

😊 Tamu meninggalkan restoran

😊 Tamu naik mobil dan pulang



Hasilnya

- 😊 Tamu merasa kenyang
- 😊 Uang tamu jadi berkurang
- 😊 Tamu senang
- 😊 Tamu kecewa
- 😊 Tamu sakit perut



- naskah dapat menggambarkan dengan akurat situasi yang terjadi di hampir semua restoran swalayan (cepat saji / fast food). Adegan-adegan merupakan naskah mini yang ada di dalam naskah utama yang melukiskan berbagai bagian dari seluruh proses

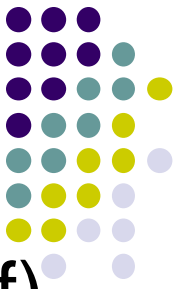


- Naskah sangat baik untuk meramalkan apa yang akan terjadi dalam situasi tertentu. Walaupun peristiwa itu tidak pernah diobservasi terlebih dahulu, tapi naskah dapat memberikan kemungkinan kepada komputer untuk meramalkan apa yang akan terjadi, kepada siapa dan kapan



4. Pohon keputusan

- Pohon keputusan adalah representasi sederhana dari contoh klasifikasi.
- Pembelajaran dengan pohon keputusan merupakan teknik ‘terbaik’ untuk pembelajaran dengan supervisi.



- **Decision tree** atau **classification tree** adalah pohon yang setiap internal (non-leaf) node dilabelkan dengan fitur input.
- Arcs yang keluar dari node dilabelkan masing-masing dengan nilai kemungkinan pada setiap fitur.
- Setiap leaf (daun) dari pohon dilabelkan dengan class or distribusi kemungkinan (probability distribution) dari kelas-kelas tersebut.



- Permasalahan klasifikasi umum:
 - Diberikan nilai atribut sebuah object, tentukan class dari object tsb.
 - Contoh:
 - Objects: Patients
 - Classes: Diabetes dan No-Diabetes
 - Attributes: Umur, ras, berat badan, tingkat kolesterol, tekanan darah dsb.



Example Training Set

Object ID	Attributes			Class
	Age	Weight	Cholesterol	
1	60	130	100	No
2	38	150	120	No
3	65	200	130	Diabetes
4	72	210	120	Diabetes
5	60	175	90	No



- Setiap internal node dihubungkan dengan kondisi yang sesuai dengan nilai atribut.
- Setiap internal node mempunyai dua cabang: Satu cabang yang memenuhi kondisi yang diinginkan, dan satu cabang lainnya yang tidak sesuai dengan kondisi.
- Setiap node daun (leaf node) dilabelkan dengan class.
- Bagaimana menggunakan decision tree?
 - Mengklasifikasi setiap object,
 - 1. Mulai dengan object pada root node.
 - 2. Turun ke level berikutnya, periksa nilai atribut objects yang tidak sesuai dengan kondisi internal node, pilih cabang yang sesuai.
 - 3. Pada saat object mencapai leaf, label class dari leaf digunakan untuk memperikakan class dari object.

Contoh



Dalam contoh ini,

Kick, Punch, Retreat dan Chase adalah class.

Skill A, Type, Strength, Player Strength and Health adalah attributes.

