

Kecerdasan Buatan

Reasoning, Semantic Network dan Frame

Oleh Politeknik Elektronika Negeri Surabaya
2017



Politeknik Elektronika Negeri Surabaya
Departemen Teknik Informatika dan Komputer

Konten

- Reasoning
- Semantic Network
- Frame

Tujuan Instruksi Umum

Mahasiswa memahami filosofi Kecerdasan Buatan dan mampu menerapkan beberapa metode Kecerdasan Komputasional dalam menyelesaikan sebuah permasalahan, baik secara individu maupun berkelompok/kerjasama tim.

Tujuan Instruksi Khusus

- Mengetahui cara representasi Pengetahuan reasoning, semantic network, frame
- Mengetahui permasalahan direpresentasikan dengan RP tersebut

Reasoning

- Mengamati permasalahan untuk mendapatkan jawaban dari suatu pertanyaan yang didasarkan pada data mengenai fakta sederhana

Contoh 1

1. Marcus was a man
2. Marcus was a Pompeian
3. Marcus was born in 40 A.D.
4. All men are mortal
5. All Pompeian died when the volcano erupted in 79 A.D.
6. No mortal lives longer than 150 years
7. It is now 2002 A.D.

Contoh 1

Apakah Marcus telah meninggal?

Contoh 1

```
man(marcus).  
pompeian(marcus).  
birth(marcus,1940).  
mortal(X):-man(X).  
age(X,AGE):-birth(X,BIRTH), year(Y), AGE is(Y-BIRTH).  
dead(X):-mortal(X), age(X,AGE), AGE > 150.  
dead(X):-pompeian(X), year(Y), Y >= 1979.  
year(2002).
```

Contoh 1

?- dead(marcus).

YES

Reason 1

1. Marcus was a Pompeian
2. All Pompeian died when the volcano erupted in 79 A.D.
3. It is now 2002 A.D.

Reason 2

1. Marcus was a man
2. All men are mortal
3. No mortal lives longer than 150 years
4. Marcus was born in 40 A.D.
5. It is now 2002 A.D.

Contoh 2

- (1) Agus adalah seorang lelaki
- (2) Agus adalah seorang berkebangsaan x
- (3) Agus dilahirkan pada tahun 1908
- (4) Semua manusia adalah fana
- (5) Semua orang berkebangsaan x meninggal karena adanya bencana banjir tahun 1995
- (6) Tidak ada manusia yang dapat hidup lebih lama dari 80 tahun
- (7) Sekarang tahun 2006



Contoh 2

- lelaki(agus). %(1)
- kebangsaanx(agus). %(2)
- lahir(agus,1908). %(3)
- fana(X):-lelaki(X). %(4)
- age(X,AGE):-lahir(X,BIRTH), tahun(Y), AGE is (Y-BIRTH).
- mati(X):-kebangsaanx(X), age(X,UMUR), UMUR > 80. %(5)
- mati(X):-kebangsaanx(X), tahun(Y), Y > 1995. %(6)
- tahun(2006). %(7).



Semantic Network

- adalah representasi yang mengekspresikan solusi permasalahan dengan menggunakan network (graph berarah)
- Pertama kali diperkenalkan oleh Quillian back di akhir tahun 60-an
- Semantic Network adalah skema representasi sederhana yang menggunakan graph berarah yang terdiri dari
 - Node → objects, concepts, events
 - Link berarah → hubungan antar node

Representasi Semantic Network

- **Lexically**, di dalamnya terdapat node (simpul), link dan batasan-batasan khusus dari permasalahan.
- **Structurally**, masing-masing link akan terkoneksi dari simpul yang paling depan (head node) sampai simpul yang paling belakang (tail node).
- **Semantically**, semua simpul dan link merepresentasikan permasalahan tetap berada dalam batasannya.



Semantik network: relation

Tidak ada standart untuk penamaan relasi pada semantik network, tapi relasi yang digunakan pada umumnya adalah sbb:

INSTANCE: X adalah INSTANCE/objek dari Y

Contoh : Ani adalah objek dari class Manusia

ISA: X adalah Y jika Class X adalah subclass dari Class Y

Contoh : Nuri adalah burung

HASPART: X mempunyai bagian Y atau Y adalah bagian dari X

Contoh : Sapi mempunyai ekor.

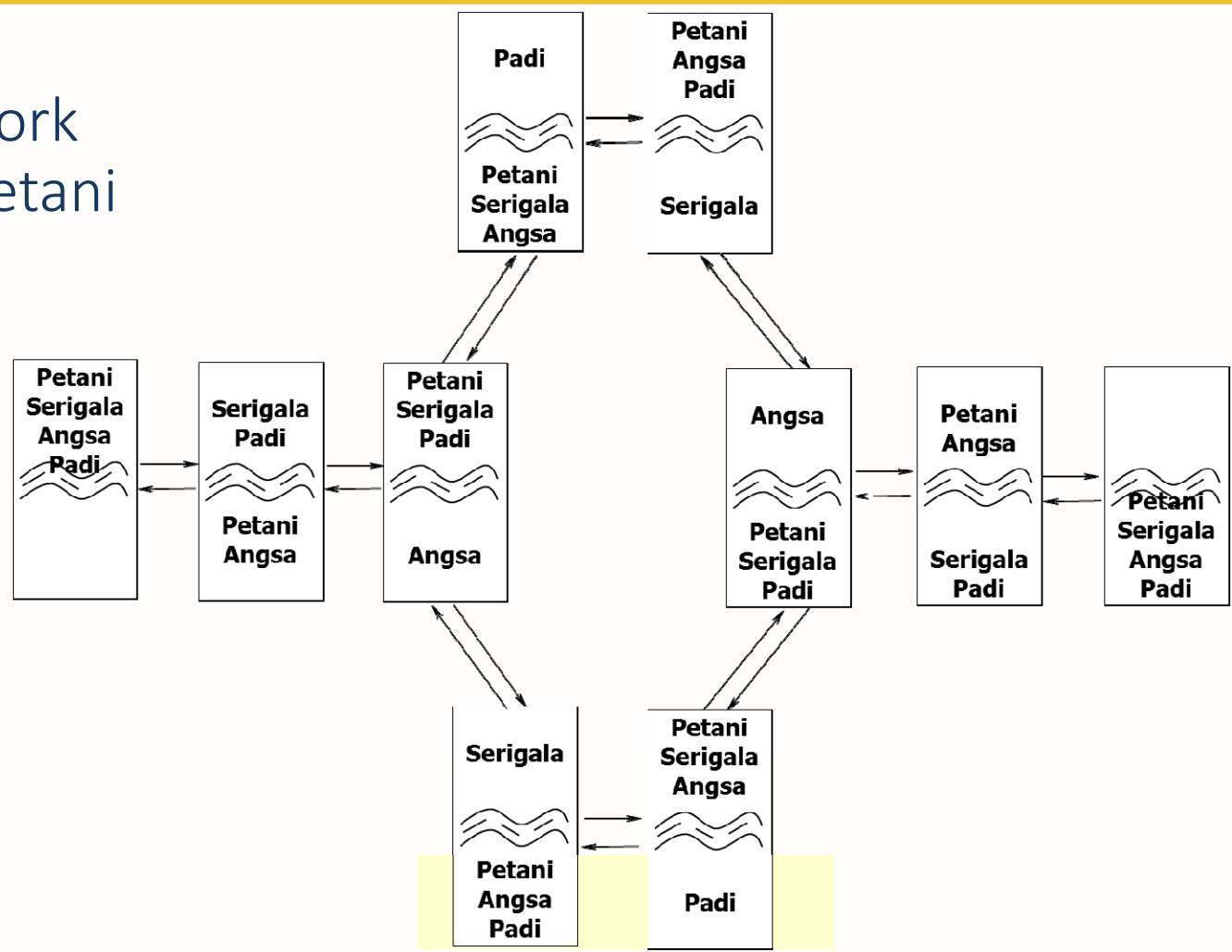


Domain-specific links: "has_disease", "father_of"

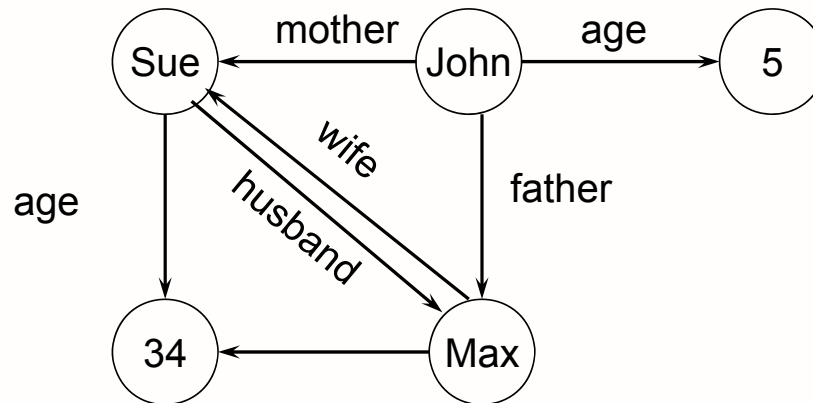
Contoh 1: permasalahan petani

- Seorang petani ingin menyeberangi sungai untuk memindah:
 - dirinya sendiri,
 - seekor serigala,
 - seekor angsa gemuk, dan
 - seikat padi
- Sayangnya, perahunya sangat terbatas;
- Petani hanya dapat membawa satu objek lain dalam satu penyeberangan.
- Dan lagi, dia tidak bisa meninggalkan serigala dan angsa dalam satu tempat, karena **serigala akan memangsa angsa**.
- Demikian pula dia tidak bisa meninggalkan **angsa dengan padi** dalam satu tempat.

Semantic network permasalahan petani



Contoh 2: Family relationship



`mother (john, sue)`
`age (john, 5)`
`wife (sue, max)`
`age (max, 34)`
...

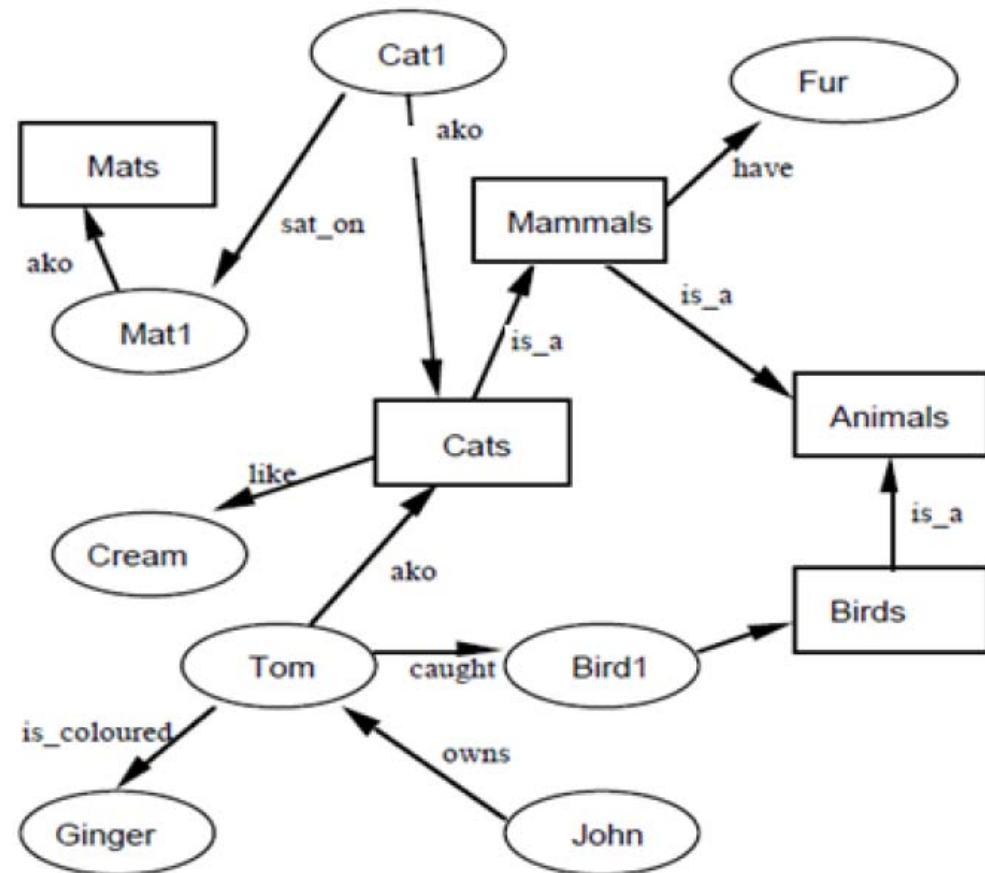
Contoh 3: Animals

- *Tom is a cat*
- *Tom caught a bird*
- *Tom is owned by John*
- *Tom is ginger in colour*
- *Cats like cream*
- *The cat sat on the mat*
- *A cat is a mammal*
- *A bird is an animal*
- *All mammals are animals*
- *Mammals have fur*

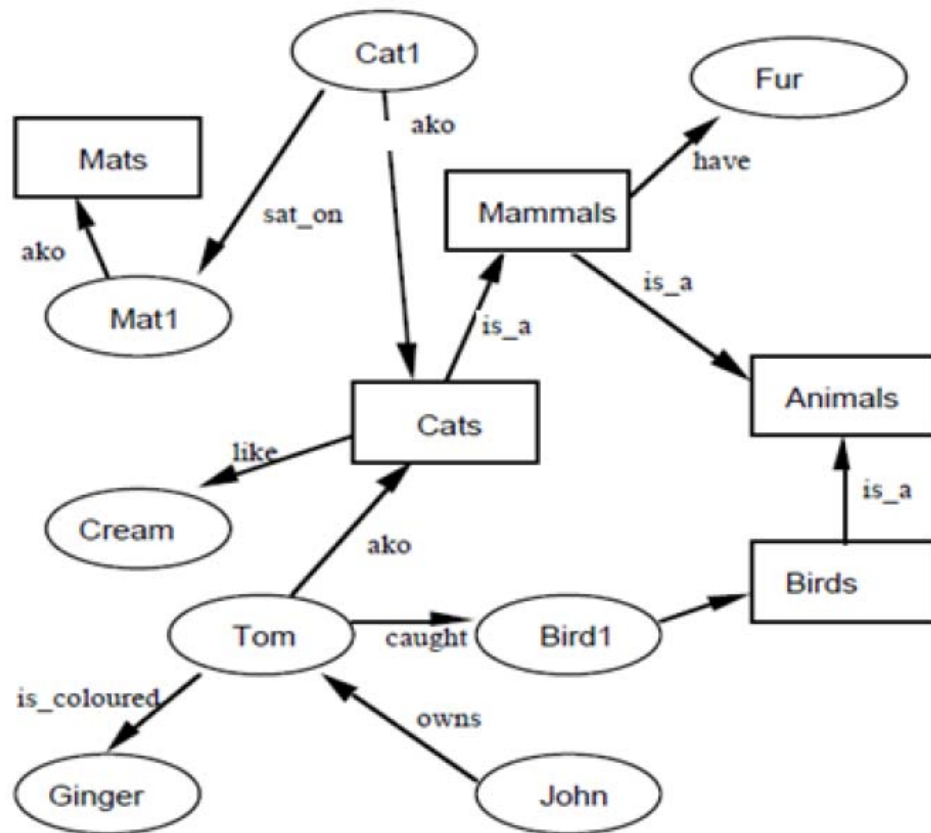


Representasi Semantic Network: Animals

- *Tom is a cat*
- *Tom caught a bird*
- *Tom is owned by John*
- *Tom is ginger in colour*
- *Cats like cream*
- *The cat sat on the mat*
- *A cat is a mammal*
- *A bird is an animal*
- *All mammals are animals*
- *Mammals have fur*



Representasi Semantic Network → Representasi Logika



```

cat(tom).
cat(cat1).
mat(mat1).
sat_on(cat1,mat1).
bird(bird1).
caught(tom,bird1).
like(X,cream) :- cat(X).
mammal(X) :- cat(X).
has(X,fur) :- mammal(X).
animal(X) :- mammal(X).
animal(X) :- bird(X).
owns(john,tom).
is_coloured(tom,ginger).
    
```

Query permasalahan animals

Siapa yang menyukai cream ?

?- like(X,cream).

Siapa saja yang merupakan mamalia ?

?- mammal(X).

Siapa saja yang mempunyai bulu ?

?- has(X,fur).

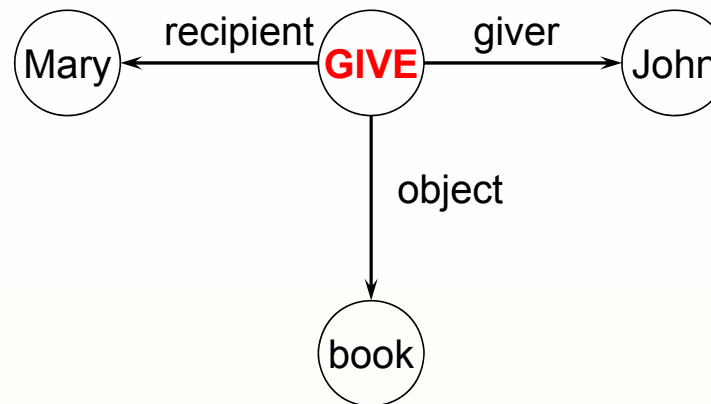
Siapa saja yang merupakan animal?

?- animal(X).

Non-binary relations

Semantic Network dapat merepresetasikan suatu kejadian yang terdiri dari:

- Pelaku
- Penerima
- Obyek

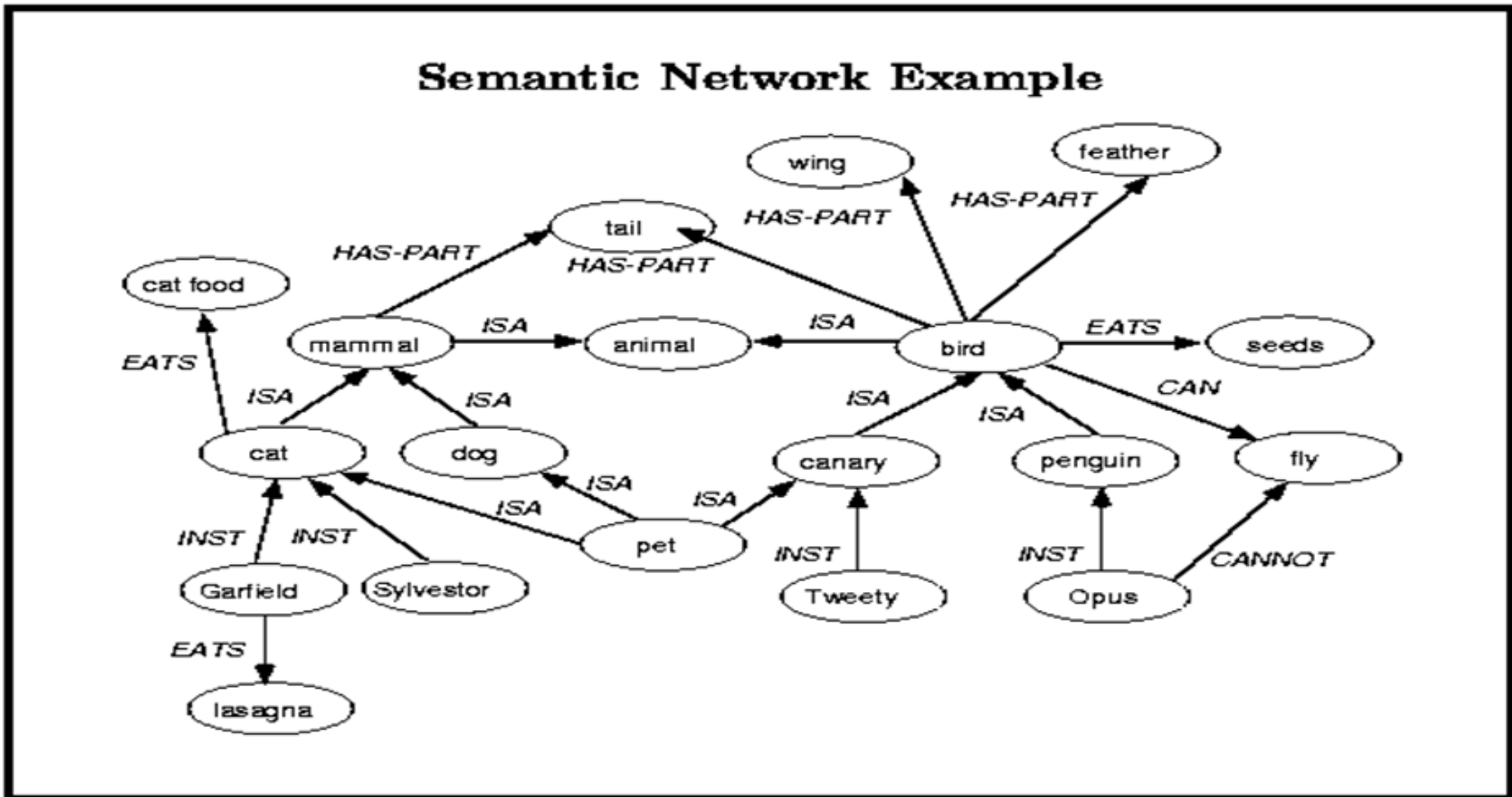


Jenis relasi: Inheritance

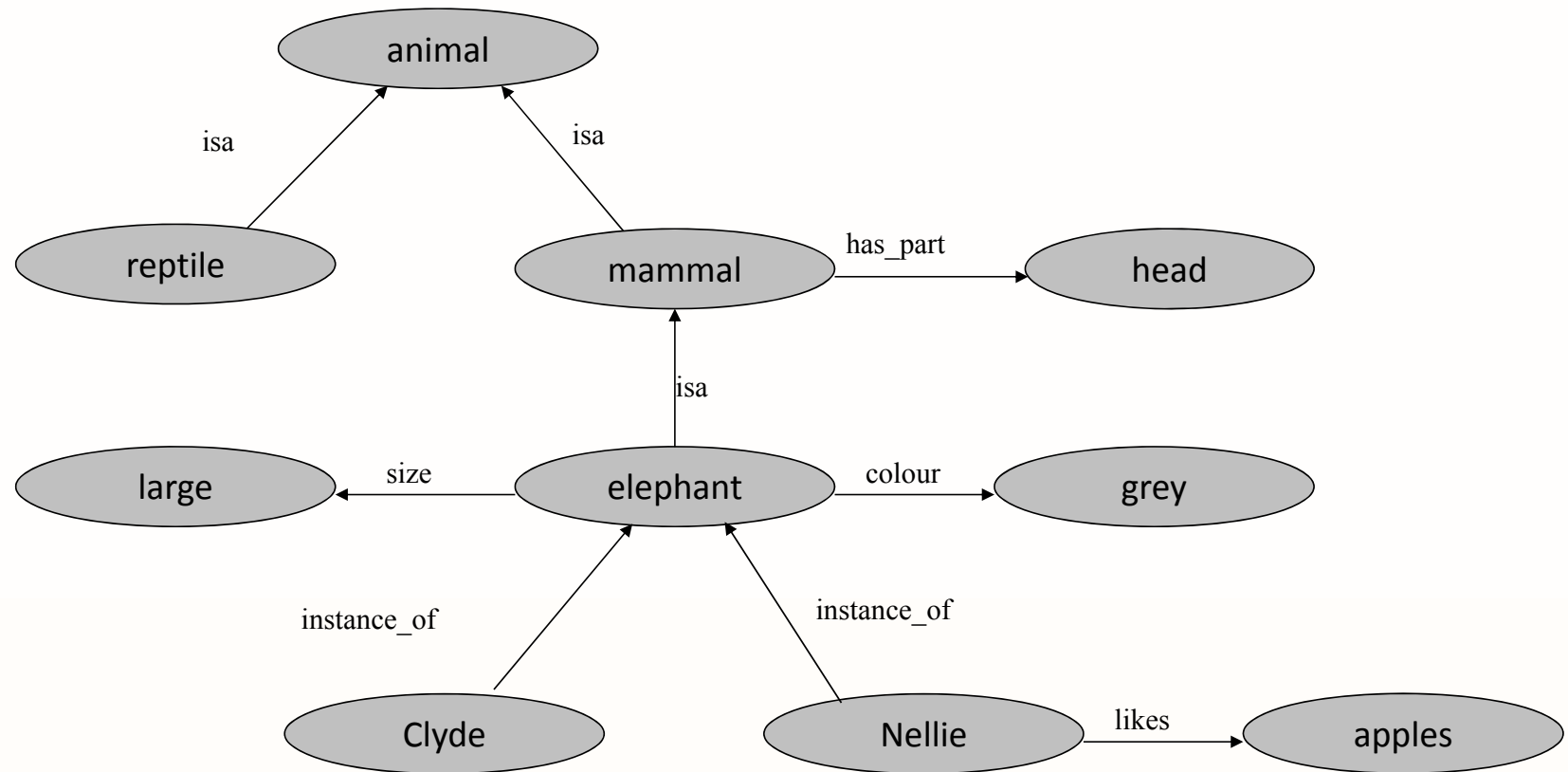
- Konsep Inheritance dinyatakan dengan relasi **is-a**
- Secara umum, jika **class X** mempunyai **properti P** maka semua class yang merupakan **subclass** dari **class X** juga mempunyai **properti P**.
- Tapi perkecualian untuk di dunia nyata
- Dalam prakteknya, properti yang **diwariskan** dari parent class ke subclass, biasanya diperlakukan sebagai **nilai default**. Jika sebuah class memiliki **nilai properti sendiri** yang berbeda dengan properti yang diwariskan maka nilai default akan **dioverride/diganti**.



Contoh 1: Inheritance



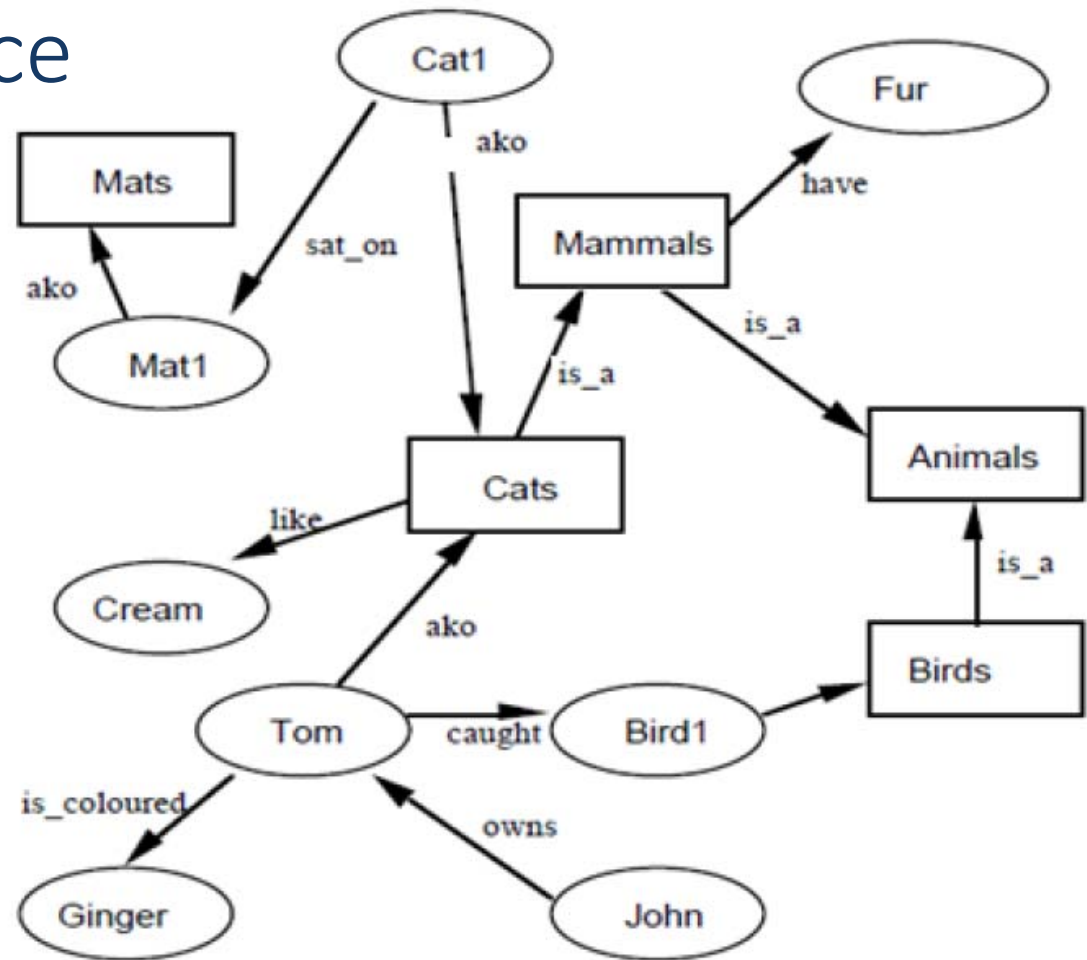
Contoh 2: Inheritance



Contoh 2: Inheritance

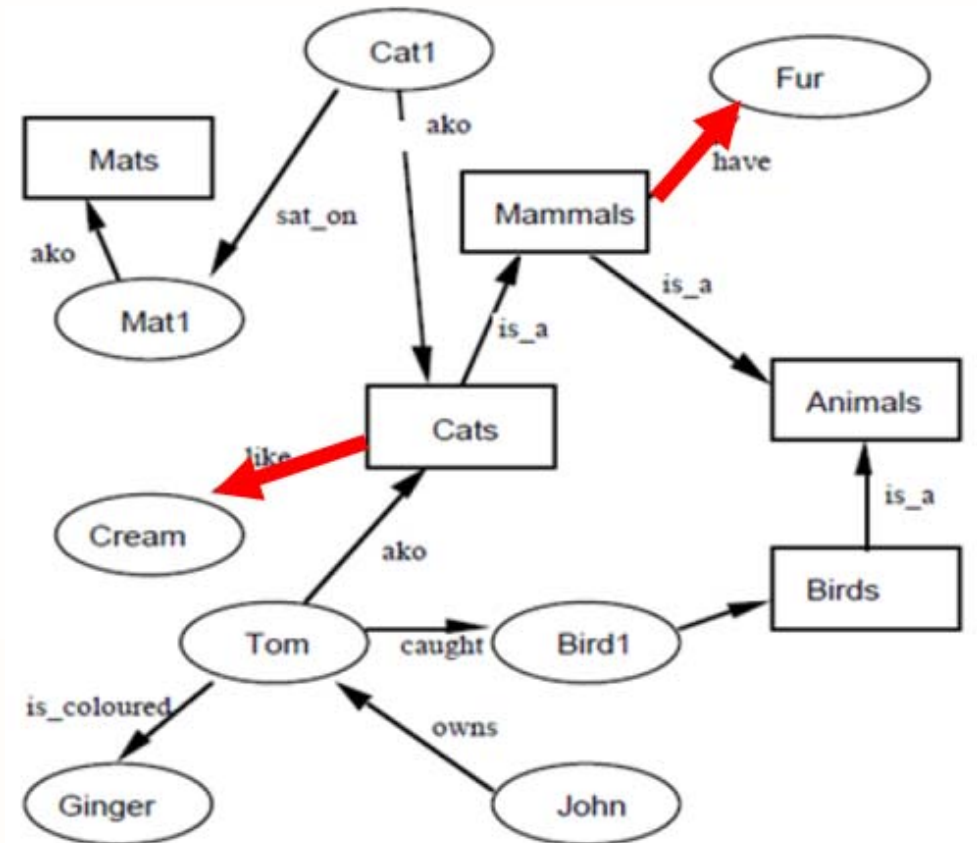
like(tom,cream).
YES

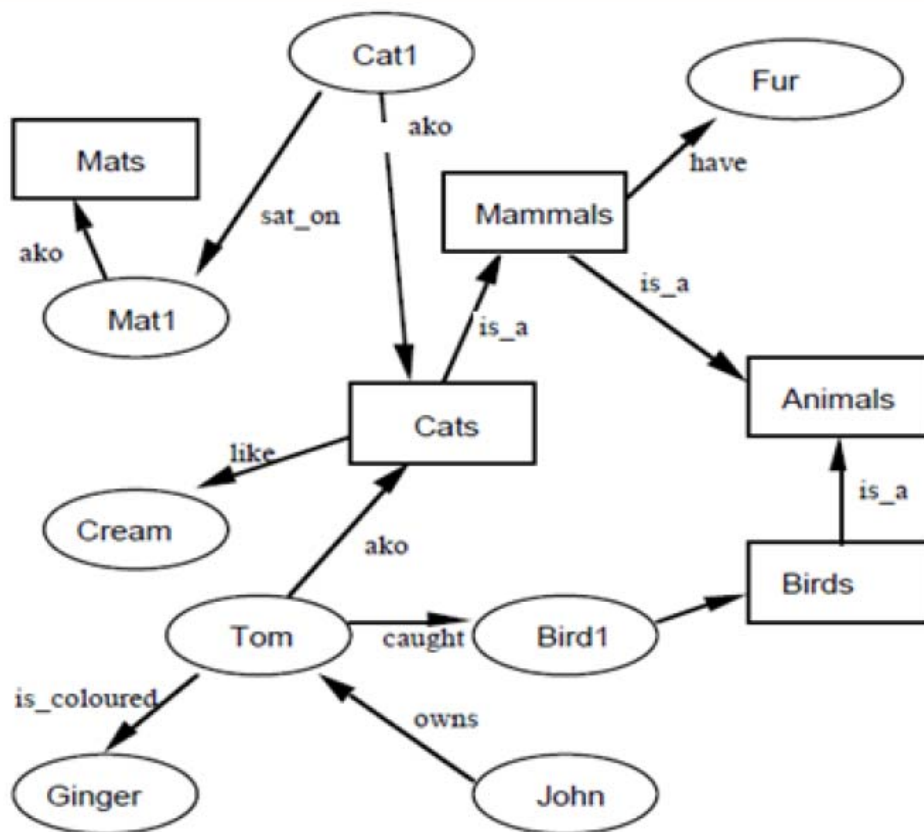
has(tom,X).
X=fur



Reification (Knowledge Representation)

Proses mengubah predikat/relasi menjadi objek dalam sistem representasi pengetahuan

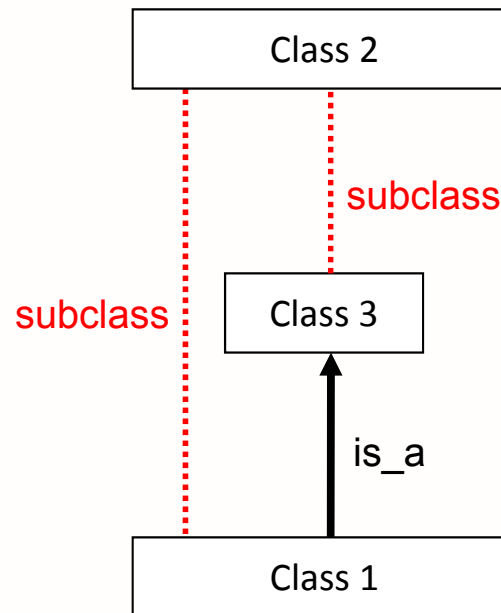
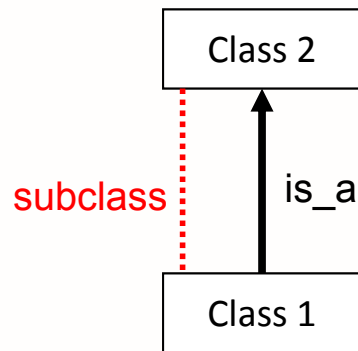




```

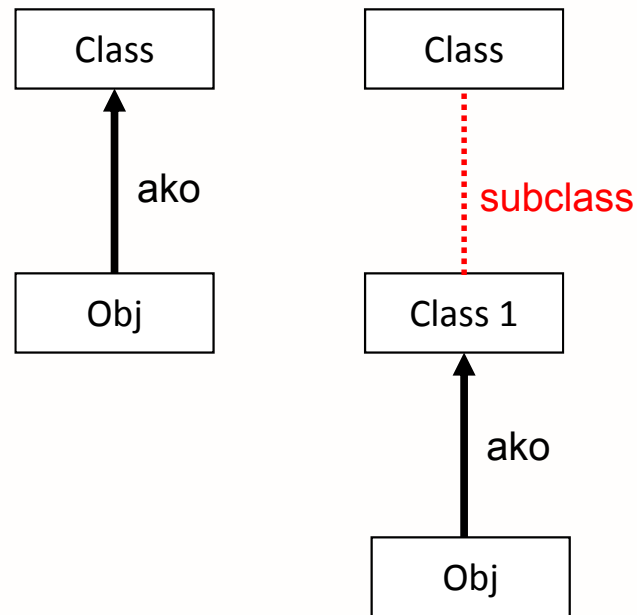
a_kind_of(mat1,mats).
a_kind_of(cat1,cats).
a_kind_of(tom,cats).
a_kind_of(bird1,birds).
caught(tom,bird1).
is_a(cats,mammals).
is_a(mammals,animals).
is_a(birds,animals).
like(cats,cream).
owns(john,tom).
sat_on(cat1,mat1).
is_coloured(tom,ginger).
have(mammals,fur).
    
```

Subclass



```
subclass(Class1,Class2) :- is_a(Class1,Class2).  
subclass(Class1,Class2) :- is_a(Class1,Class3), subclass(Class3,Class2).
```

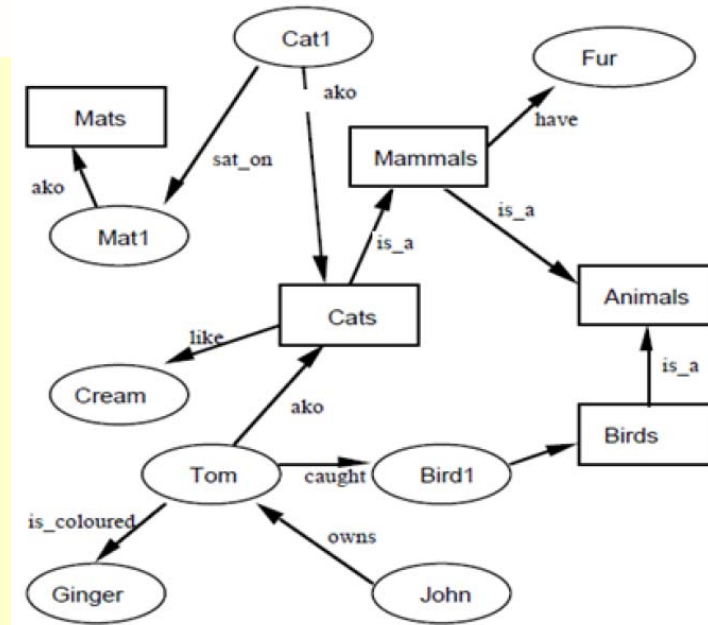
Instance of Class



`aninstance(Obj,Class) :- a_kind_of(Obj,Class).`
`aninstance(Obj,Class) :- a_kind_of(Obj,Class1), subclass(Class1,Class).`

```

a_kind_of(mat1,mats).
a_kind_of(cat1,cats).
a_kind_of(tom,cats).
a_kind_of(bird1,birds).
caught(tom,bird1).
is_a(cats,mammals).
is_a(mammals,animals).
is_a(birds,animals).
like(cats,cream).
owns(john,tom).
sat_on(cat1,mat1).
is_coloured(tom,ginger).
have(mammals,fur).
    
```



```

subclass(Class1,Class2) :- is_a(Class1,Class2).
subclass(Class1,Class2) :- is_a(Class1,Class3), subclass(Class3,Class2).
    
```

```

aninstance(Obj,Class) :- a_kind_of(Obj,Class).
aninstance(Obj,Class) :- a_kind_of(Obj,Class1), subclass(Class1,Class).
    
```

```

attribute(Obj,X) :- aninstance(Obj,Class),have(Class,X).
attribute(Obj,X) :- aninstance(Obj,Class),like(Class,X).
    
```

Multiple Inheritance

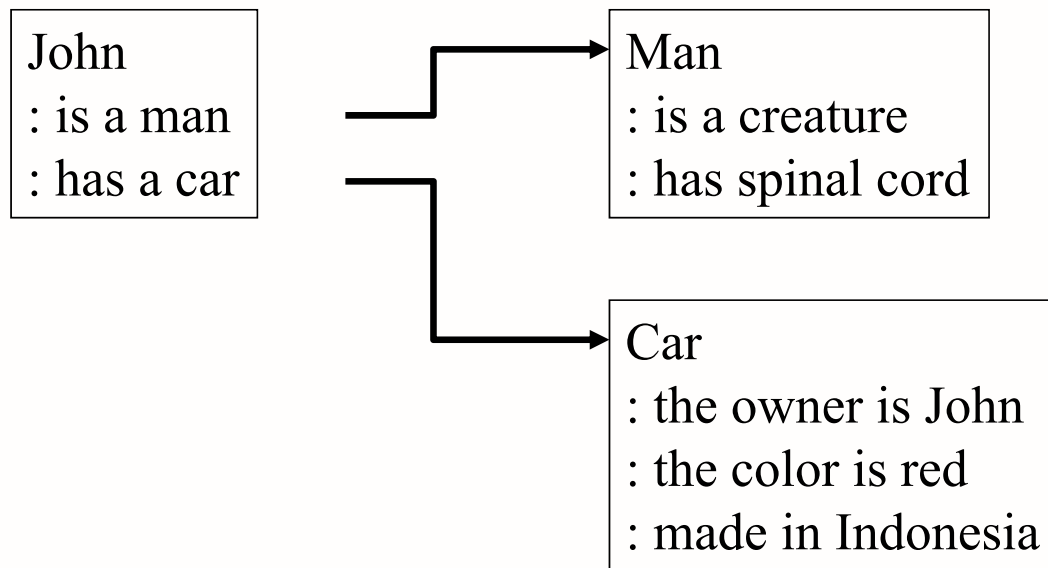
- Memungkinkan sebuah objek untuk mewarisi properti dari banyak class.
- Memungkinkan sebuah objek mewarisi properti yang berbeda (conflicting properties)
- Permasalahan diatas dapat dihindari menggunakan strategi konflik resolution (conflict resolution strategies)

Frames

- Frame digunakan untuk menyimpan knowledge
- Frame menyatakan sebuah entitas/objek yang memiliki sekumpulan **slot** (attribut) dan **nilai** (nilai untuk tiap attribut).
 - **objects** – *Mammal, Elephant*;
 - **slots** – properties such as *colour* and *size*;
 - **slot-values** – values stored in the slots, e.g. *grey* and *large*.
- Setiap atribut mempunyai nilai legal
- Frame secara tidak langsung **terhubung** dengan frame lainnya karena nilai dari atribut adalah frame



Contoh 1



Contoh 2

Mammal

isa: ANIMAL

*haspart: HAIR

*breathes: AIR

NBA_BASKETBALL_PLAYER

isa: ADULTMALE

cardinality: 400

*height: > 6'

*salary: > \$200,000

HUMAN

isa: MAMMAL

cardinality: 6 million

*haspart: LEGS(2)

MICHAELJORDAN

instance: NBABASKETBALLPLAYER

height: 6'9"

ADULTMALE

isa: HUMAN

cardinality: 2 million

*gender: male

JOHNSTOCKTON

instance: NBABASKETBALLPLAYER

height: 6'1"

asterisk (*) = attribut yang dapat diwariskan

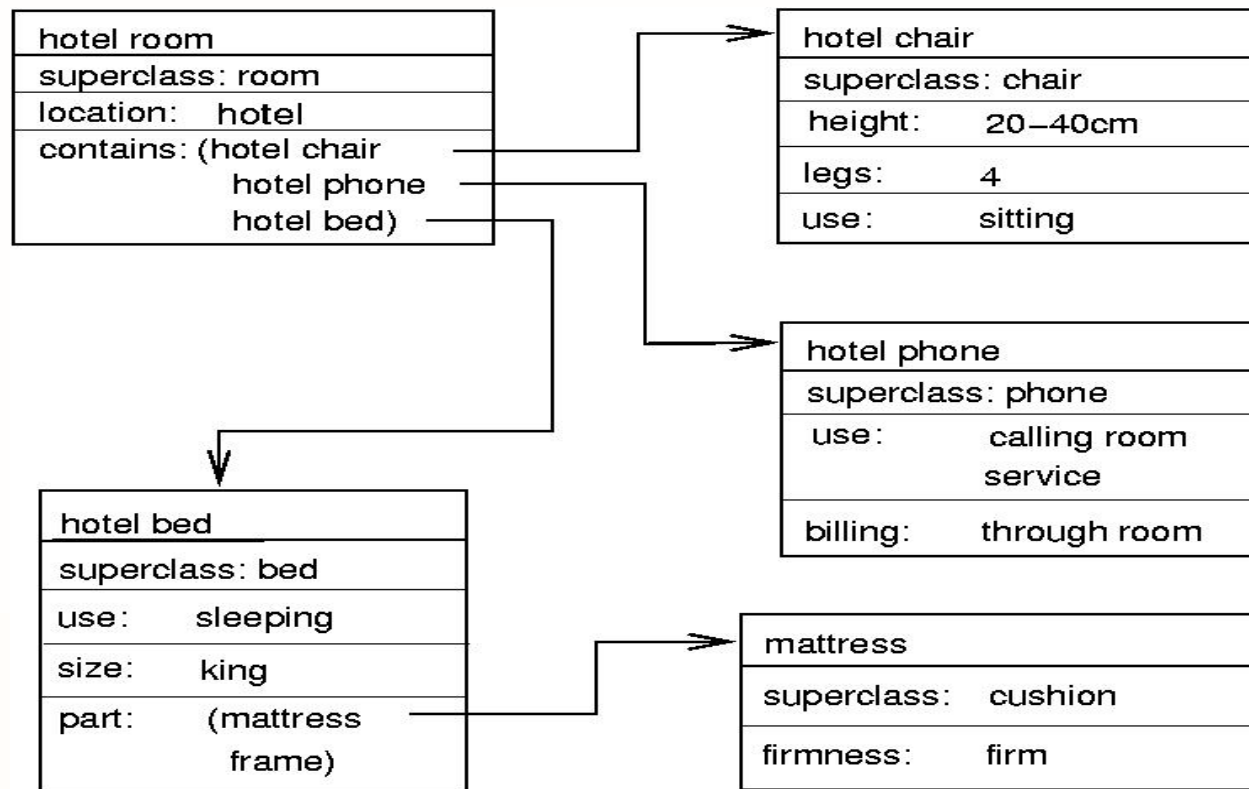


Contoh 3: Default values

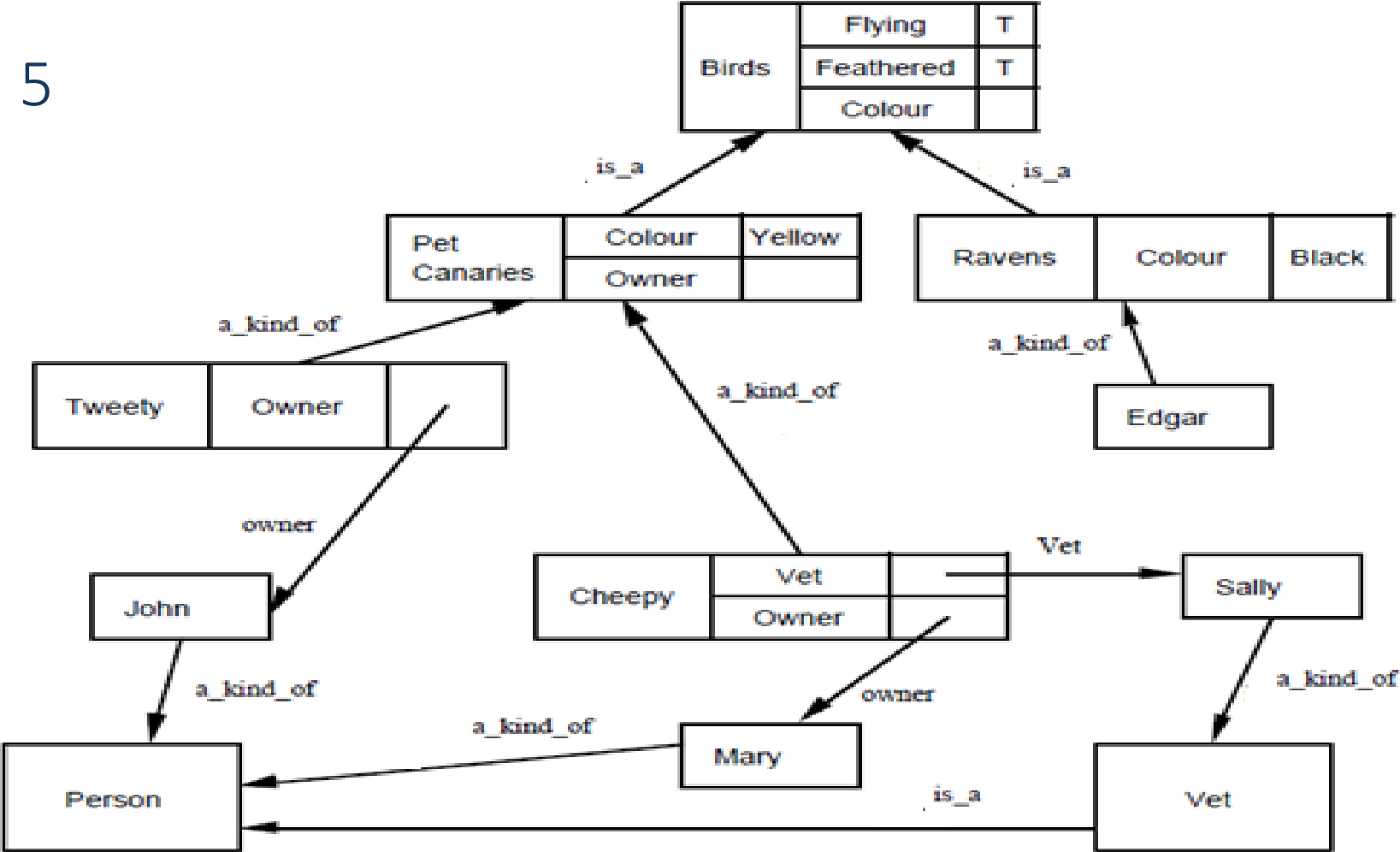
Mammal:		
	subclass:	Animal
	has-part:	head
	warm-blooded:	yes
	*furry:	yes
Elephant:		
	subclass:	Mammal
	*colour:	grey
	*size:	large
	*furry:	no
Nellie:		
	instance:	Elephant
	likes:	apples
	owner:	Fred
	colour:	pink
Clyde:		
	instance:	Elephant
	size:	small

asterisk (*) = atribut yang dapat diwariskan

Contoh 4



Contoh 5



Merubah Semantic net Menjadi Frame

Mammal:

subclass: Animal
has-part: head

Elephant:

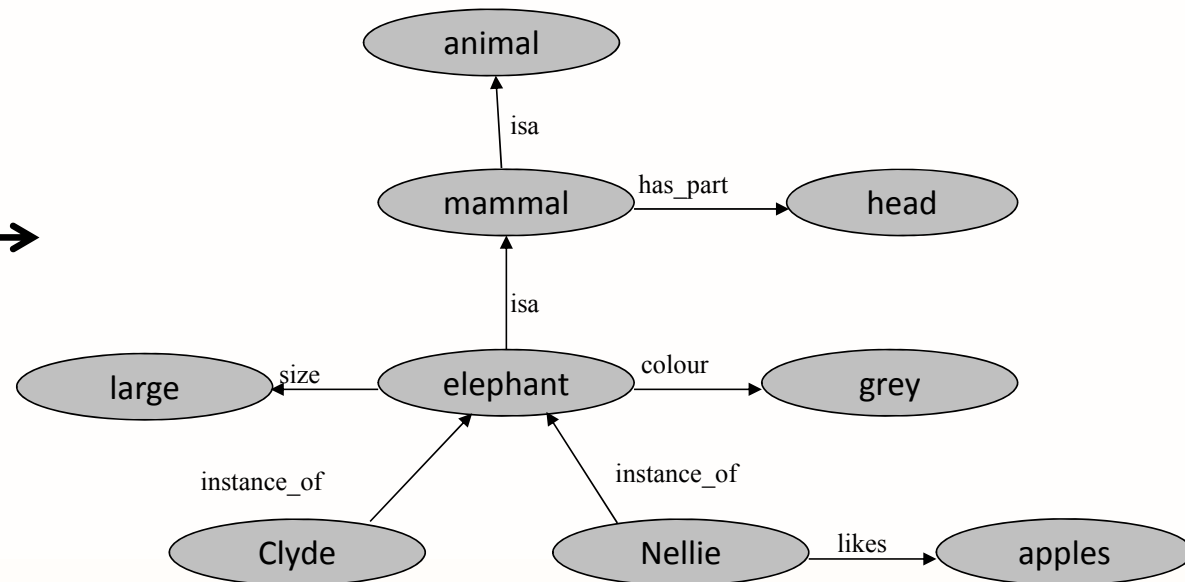
subclass: Mammal
colour: grey
size: large

Nellie:

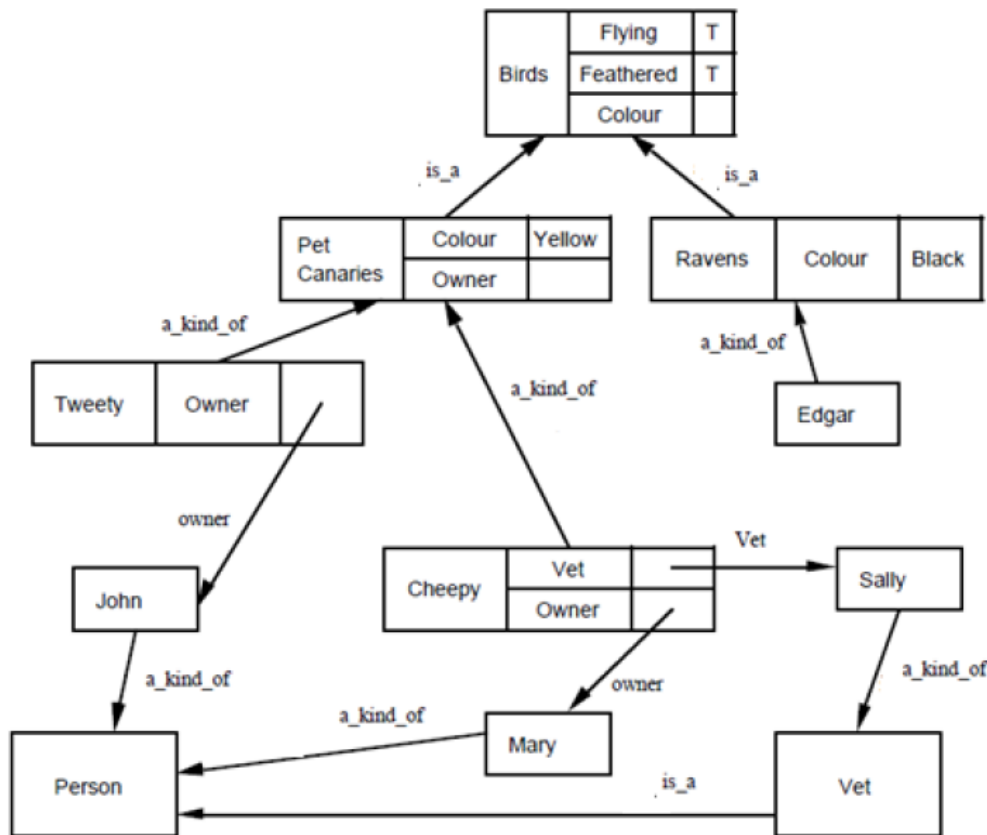
instance: Elephant
likes: apples

Clyde:

instance: Elephant



Frames



```

attribute(birds, flying, true).
attribute(birds, feathered, true).
attribute(pet_canaries, colour, yellow).
attribute(ravens, colour, black).
attribute(tweety, owner, john).
attribute(cheepy, owner, mary).
attribute(cheepy, vet, sally).
is_a(pet_canaries, birds).
is_a(ravens, birds).
is_a(vet, person).
a_kind_of(edgar, ravens).
a_kind_of(tweety, pet_canaries).
a_kind_of(cheepy, pet_canaries).
a_kind_of(sally, vet).
a_kind_of(john, person).
a_kind_of(mary, person).
    
```

Demons

⦿ Salah satu keuntungan utama dari frame adalah kemampuan untuk memasukkan **demon untuk menghitung nilai slot.**

⦿ **Demon** adalah fungsi yang menghitung nilai slot sesuai permintaan.

HUMAN

isa: (MAMMAL)

mortal: (yes :inheritable yes)

cardinality: (6 million :inheritable no)

age: (:inheritable yes :demon compute_age)

MARY

int Compute_Age (frame)

instance: HUMAN

return(today- (query birthday slot));

gender: FEMALE

birthday: 11/04/60



Kelebihan Frame

- Frame dapat mendukung nilai-nilai yang lebih alami daripada semantic net
- Frame dapat dengan mudah diimplementasikan menggunakan pemrograman berorientasi objek.
- Konsep inheritance dapat mudah dilihat dan dikontrol.

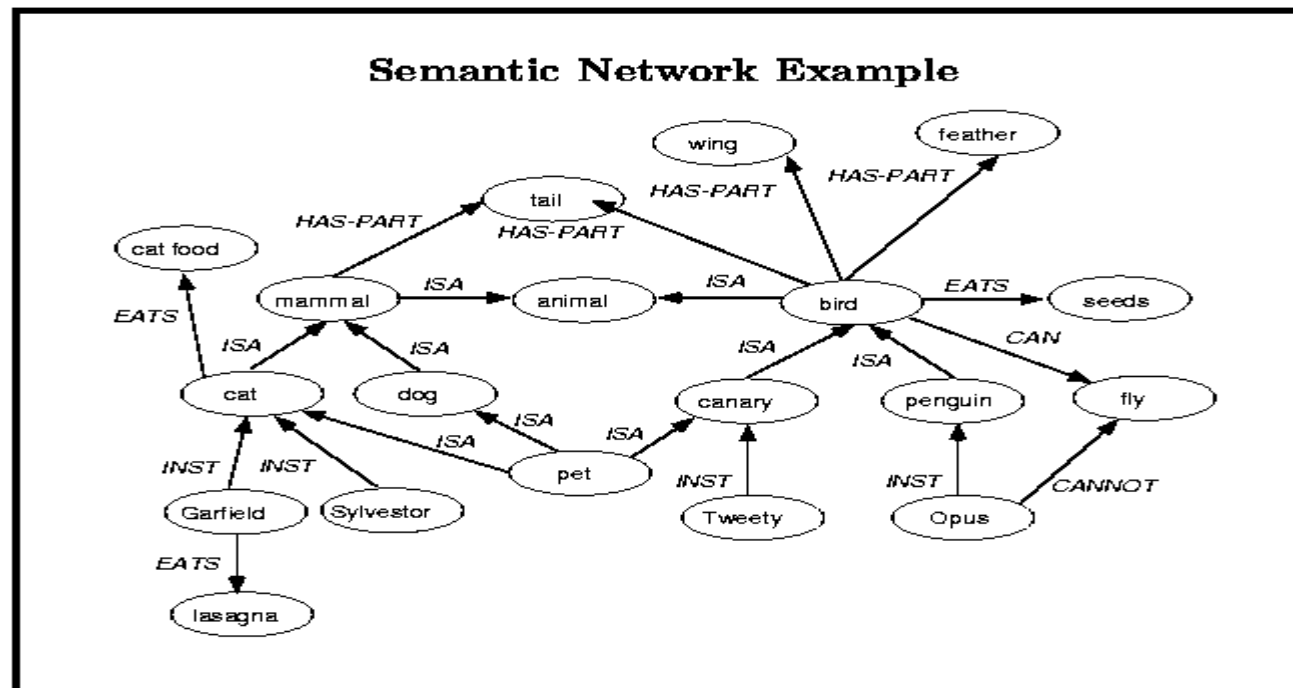
Tugas 1

- Anas adalah seorang lulusan SD
- Anas adalah WNI
- Anas dilahirkan pada tahun 1952
- Semua orang WNI yang lulusan SD tidak dapat menjadi PNS lagi jika umurnya lebih dari 35 tahun
- Anas mencoba daftar menjadi PNS pada tahun 1985
- Semua PNS akan pensiun jika umurnya mencapai 60 tahun
- Sekarang tahun 2005



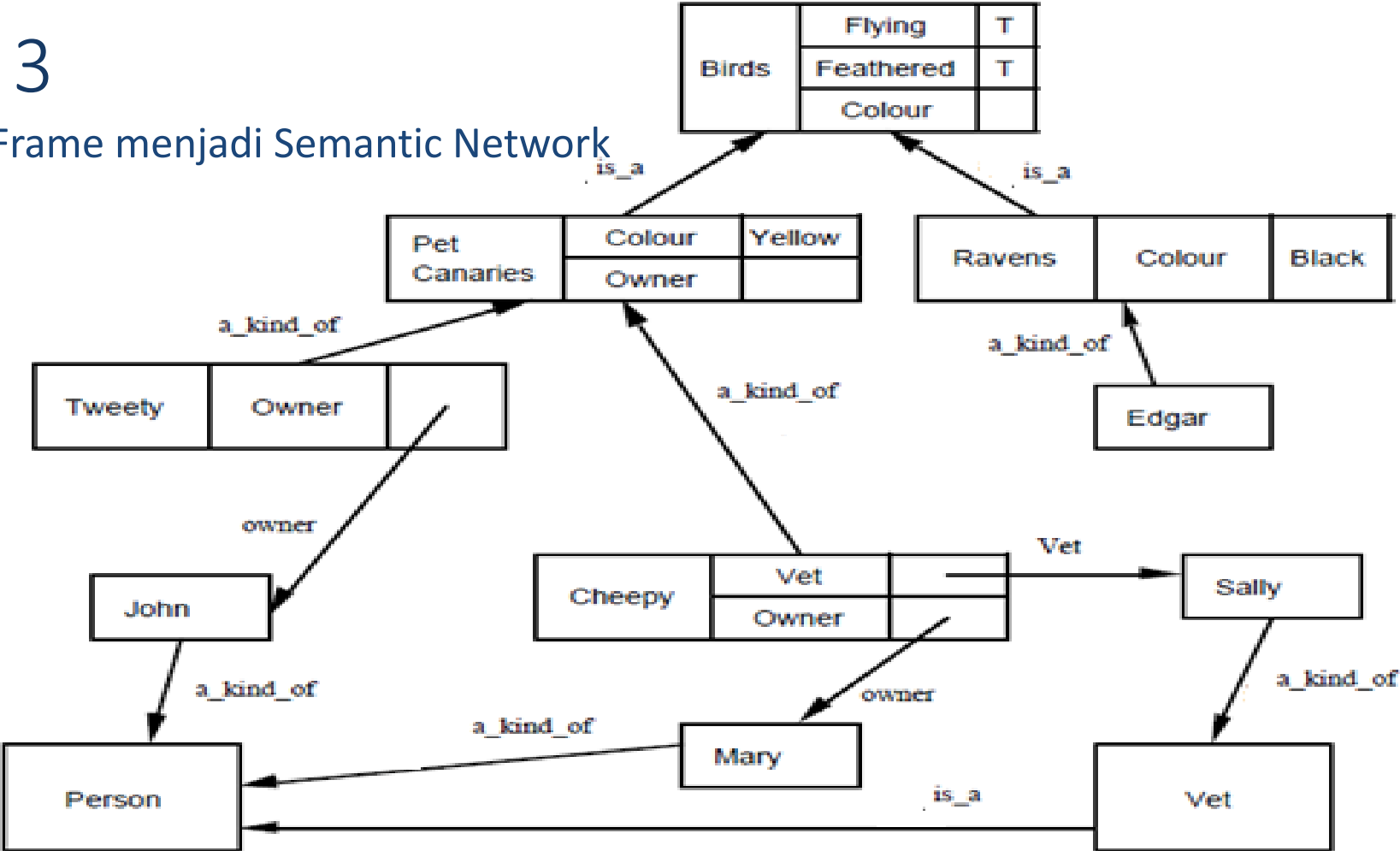
Tugas 2

Ubahlah Semantic Network menjadi Frame



Tugas 3

- Ubahlah Frame menjadi Semantic Network





bridge to the future

<http://www.eepis-its.edu>