

Laporan Tugas Proyek I IF4070

Ontologi dan Sistem Berbasis Pengetahuan untuk Klasifikasi Minuman Kopi



**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO & INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG**

Kelompok U:

13522011 - Dewantoro Triatmodjo

13522055 - Benardo

13522113 - William Glory Henderson

DAFTAR ISI

DAFTAR ISI.....	1
DAFTAR TABEL.....	4
BAB I	
Domain.....	5
1.1. Penjelasan Domain dan Batasan.....	5
1.2. Kecocokan Domain.....	7
BAB II	
Ontologi.....	9
2.1. Sumber Pengetahuan.....	9
2.2. Gambaran Umum Ontologi.....	10
2.2.1 Arsitektur Ontologi.....	10
2.2.2 Hierarki Konsep Utama (Classes).....	11
2.2.3 Object Properties.....	12
2.2.4 Kelas-Kelas Kopi Spesifik.....	12
2.2.5 Asumsi Desain dan Struktur.....	13
BAB III	
Sistem Berbasis Pengetahuan (KBS).....	14
3.1 Fungsionalitas Sistem.....	14
3.1.1 Klasifikasi Minuman Kopi.....	14
3.1.2 Identifikasi Kopi Berdasarkan Nama.....	15
3.1.3 Query Helper Functions.....	16
3.2 Pemanfaatan Ontologi.....	17
3.2.1 Mapping TBox ke Prolog Rules.....	17
3.2.2 Mapping ABox ke Prolog Facts.....	18

BAB IV

Pembahasan.....	19
4.1. Hasil Sistem Berbasis Pengetahuan.....	19
4.1.1 Testing Klasifikasi Dasar.....	19
4.1.2 Testing Identifikasi Kopi.....	20
4.1.3 Testing Query Helper Functions.....	21
4.1.4 Testing Distinguishability.....	22
4.1.5 Testing Boundary Cases.....	23
4.1.6 Verification dengan Ontologi.....	24
4.2 Keterlibatan Logika Deskripsi.....	25
4.2.1 Konstruksi Logika Deskripsi yang Digunakan.....	25
4.3 Keterbatasan Ontologi dan Sistem.....	29
4.3.1 Keterbatasan Domain.....	29
4.3.2 Keterbatasan Ontologi.....	29
4.3.3 Keterbatasan Sistem Prolog.....	30
BAB V	
Simpulan.....	31
BAB VI.....	32
Lampiran: Matriks Kontribusi.....	32

DAFTAR TABEL

Tabel 2.1 Object Property CoffeeDrink.....	12
Tabel 3.1 Penjelasan Parameter classify_coffee/7.....	14

BAB I

Domain

1.1. Penjelasan Domain dan Batasan

Domain yang dipilih untuk tugas ini adalah klasifikasi minuman kopi (coffee drink classification). Domain ini berfokus pada representasi pengetahuan untuk mendeskripsikan dan mengklasifikasikan beberapa jenis minuman kopi yang umum ditemukan. Kopi merupakan minuman yang memiliki variasi yang sangat beragam di seluruh dunia, dengan metode penyajian, komposisi bahan, dan proses pembuatan yang berbeda-beda. Perbedaan-perbedaan ini dapat direpresentasikan secara formal melalui ontologi serta aturan klasifikasi dapat dinyatakan dengan jelas menggunakan logika deskripsi. Untuk menjaga agar ontologi tetap fokus (tidak terlalu luas) dan dapat diimplementasikan sesuai spesifikasi tugas, diterapkan batasan sebagai berikut:

1. Fokus pada Minuman

- Espresso
- Bica
- Americano
- Latte
- Caffè Macchiato
- Cappuccino
- Flat White
- Latte Macchiato
- Kopi Tubruk
- Greek Coffee
- Cafe Mocha

2. Fokus pada Klasifikasi

- Base (Basis Kopi)
 - Espresso Base

- Brewed Coffee Base
- Milk (Jenis Susu)
 - No Milk
 - Steamed Milk
 - Foamed Milk
 - Steamed And Foamed Milk
 - Microfoam Milk
- Additives (Bahan Tambahan)
 - No Additive
 - Hot Water Additive
 - Sugar Additive
 - Chocolate Additive
- Preparation (Metode Pembuatan)
 - Pressure Method
 - Boiled Method
 - Diluted Method
 - Combined Method
- Serving (Gaya Penyajian)
 - Small Cup Serving
 - Tall Glass Serving
 - Large Cup Serving
 - Cup Serving
 - Demitasse Serving
 - Unfiltered Serving
- Origin (Asal/Negara)
 - Italy Origin
 - Portugal Origin
 - Indonesia Origin
 - Greece Origin

- Australia / New Zealand Origin

1.2. Kecocokan Domain

Domain ini dipilih karena sangat cocok dengan tujuan pengembangan sistem berbasis pengetahuan:

1. Aturan Klasifikasi yang Logis

Setiap jenis minuman kopi pada dasarnya memiliki resep atau karakteristik yang relatif stabil dan dapat dinyatakan secara eksplisit. Hal ini membuat domain sangat cocok untuk logika deskripsi (DL) pada level ontolog dan aturan prolog pada level KBS. Misalnya:

- Latte: minuman berbasis espresso dengan steamed milk dalam jumlah dominan, hampir tanpa foam, biasanya disajikan di gelas tinggi
- Cappuccino: minuman berbasis espresso dengan kombinasi steamed milk dan foam
- Kopi Tubruk: kopi seduh bergaya Indonesia, basisnya brewed coffee, direbus bersama gula dan disajikan tidak disaring sehingga ampas mengendap di dasar cangkir

2. Hierarki Konsep yang Natural

Domain kopi menyediakan hierarki konsep yang natural:

- CoffeeDrink sebagai konsep tertinggi
- Subclass berdasarkan basis: misalnya EspressoCoffee (berbasis EspressoBase) dan minuman seperti KopiTubrukCoffee / GreekCoffeeDrink yang berbasis BrewedCoffeeBase
- Karakteristik tambahan direpresentasikan melalui property restrictions seperti:
 - hasBase
 - hasMilk
 - hasAdditive
 - hasPreparation
 - hasServing
 - hasOrigin

Struktur ini sesuai dengan cara ontologi dan logika deskripsi bekerja yaitu setiap minuman adalah kombinasi dari beberapa fitur dan kelas-kelas seperti LatteCoffee atau FlatWhiteCoffee didefinisikan sebagai irisan (intersection) dari CoffeeDrink dengan sejumlah restrictions atas properti-properti tersebut. Hal yang sama kemudian diimplementasikan kembali di Prolog melalui fakta dan aturan-aturan.

3. Referensi yang Mudah dan banyak

Kelebihan lain dari domain ini adalah mudah diverifikasi:

- Definisi minuman klasik seperti Espresso, Latte, Cappuccino, Macchiato, Flat White, Cafe Mocha, Kopi Tubruk, dan Greek Coffee banyak dibahas di artikel kopi dan wikipedia
- Konsumen / peminat kopi pada umumnya punya intuisi tentang perbedaan minuman-minuman ini, sehingga hasil klasifikasi mudah dijelaskan dan diuji secara informal

Dengan kata lain, verifikasi dan validasi terhadap ontologi maupun KBS bisa dilakukan dengan membandingkan dengan rujukan tertulis dan menguji sistem dengan contoh kasus nyata di dunia sehari-hari.

BAB II

Ontologi

2.1. Sumber Pengetahuan

Ontologi dirancang berdasarkan gabungan pengetahuan dari artikel dan wikipedia yang mendeskripsikan karakteristik minuman kopi secara konsisten. Sumber-sumber ini menyediakan informasi standar industri mengenai komposisi, metode pembuatan, dan gaya penyajian minuman. Pengetahuan tentang 11 jenis kopi didapatkan dari link berikut:

1. Coffee Preparation (overview)
https://en.wikipedia.org/wiki/Coffee_preparation
2. Espresso
<https://en.wikipedia.org/wiki/Espresso>
3. Bica (coffee)
[https://en.wikipedia.org/wiki/Bica_\(coffee\)](https://en.wikipedia.org/wiki/Bica_(coffee))
4. Americano (coffee)
[https://en.wikipedia.org/wiki/Americano_\(coffee\)](https://en.wikipedia.org/wiki/Americano_(coffee))
5. Latte
<https://en.wikipedia.org/wiki/Latte>
6. Caffè Macchiato
https://en.wikipedia.org/wiki/Caff%C3%A8_macchiato
7. Cappuccino
<https://en.wikipedia.org/wiki/Cappuccino>
8. Flat White
https://en.wikipedia.org/wiki/Flat_white
9. Latte Macchiato
https://en.wikipedia.org/wiki/Latte_macchiato
10. Kopi Tubruk
https://en.wikipedia.org/wiki/Kopi_tubruk
11. Greek Coffee

https://en.wikipedia.org/wiki/Greek_coffee

12. Cafe Mocha

https://en.wikipedia.org/wiki/Cafe_mocha

Dari setiap artikel di atas, informasi diekstraksi secara sistematis menjadi enam jenis fitur ontologis:

1. Base - bahan dasar kopi (espresso atau brewed)
2. Milk Type - jenis susu/foam yang digunakan
3. Additive - bahan tambahan seperti air panas, gula, coklat
4. Preparation Method - metode pembuatan (pressure, boiled, diluted, combined)
5. Serving Style - gaya penyajian (small cup, tall glass, large glass, unfiltered, demitasse)
6. Origin - asal gaya minuman (Italia, Portugal, Indonesia, Yunani, Aus/NZ)

Tujuan utama penggunaan sumber ini adalah memastikan bahwa definisi minuman yang dimodelkan selaras dengan praktik umum dan konsisten dengan sumber terpercaya.

2.2. Gambaran Umum Ontologi

Ontologi dikembangkan menggunakan Protégé. Ontologi ini mendefinisikan TBox (terminologi) dari domain jenis minuman kopi.

2.2.1 Arsitektur Ontologi

1. TBox (Terminological Box)
Berisi definisi konsep (classes) dan object properties yang menghubungkan konsep tersebut. Pada bagian ini juga didefinisikan axioms dan beberapa equivalent class yang menyatakan aturan formal sebuah minuman.
2. ABox (Assertional Box)
Berisi instances atau individu konkret, digunakan untuk contoh komponen (misal: NoMilk) dan contoh minuman (misal: LatteCoffee, KopiTubrukCoffee) untuk validasi struktur ontologi.

2.2.2. Hierarki Konsep Utama (Classes)

Ontologi memiliki 7 kelas, masing-masing dengan subclass yang merepresentasikan fitur spesifik:

1. CoffeeDrink (Konsep utama untuk semua minuman kopi)
2. Base (Basis Kopi)
 - EspressoBase: kopi yang diekstrak menggunakan tekanan (espresso machine)
 - BrewedCoffeeBase: kopi yang diseduh biasa
3. MilkType (Jenis Susu/foam)
 - NoMilk: tanpa susu
 - SteamedMilk: susu yang dipanaskan dengan uap
 - FoamedMilk: susu dengan busa / foam
 - SteamedAndFoamedMilk: kombinasi susu steamed dan foam
 - MicrofoamMilk: susu dengan busa halus dan lembut
4. Additive (Bahan Tambahan)
 - NoAdditive: tanpa tambahan
 - HotWaterAdditive: air panas
 - SugarAdditive: gula
 - ChocolateAdditive: coklat\
5. PreparationMethod (Metode Pembuatan)
 - PressureMethod: dengan mesin espresso
 - BoiledMethod: direbus
 - DilutedMethod: diencerkan
 - CombinedMethod: kombinasi dari beberapa komponen
6. ServingStyle (Gaya Penyajian)
 - SmallCupServing: cangkir kecil
 - DemitasseServing: cangkir demitasse khas Eropa
 - TallGlassServing: gelas tinggi
 - LargeCupServing: cangkir besar
 - CupServing: cangkir standar

- UnfilteredServing: tanpa filter
7. Origin (Asal Tradisi Minuman)
- ItalyOrigin: Itali
 - PortugalOrigin: Portugal
 - IndonesiaOrigin: Indonesia
 - GreeceOrigin: Yunani
 - AusNzOrigin: Australia / New Zealand

2.2.3 Object Properties

Ontologi menggunakan 6 object properties untuk menghubungkan CoffeeDrink dengan fitur-fiturnya:

Tabel 2.1 Object Property CoffeeDrink

Property	Domain	Range	Deskripsi
hasBase	CoffeeDrink	Base	Menentukan basis kopi
hasMilk	CoffeeDrink	MilkType	Menentukan jenis susu
hasAdditive	CoffeeDrink	Additive	Menentukan bahan tambahan
hasPreparation	CoffeeDrink	PreparationMethod	Menentukan metode pembuatan
hasServing	CoffeeDrink	ServingStyle	Menentukan gaya penyajian
hasOrigin	CoffeeDrink	Origin	Menentukan asal minuman

2.2.4. Kelas-Kelas Kopi Spesifik

Terdapat 11 kelas kopi spesifik yang didefinisikan sebagai subclass dari CoffeeDrink:

1. EspressoCoffee
2. BicaCoffee
3. AmericanoCoffee

4. LatteCoffee
5. CaffèMacchiatoCoffee
6. CappuccinoCoffee
7. FlatWhiteCoffee
8. LatteMacchiatoCoffee
9. KopiTubrukCoffee
10. GreekCoffeeDrink
11. CafeMochaCoffee

2.2.5. Asumsi Desain dan Struktur

1. Disjointness

Kelas-kelas top-level (CoffeeDrink, Base, MilkType, Additive, PreparationMethod, ServingStyle, Origin) dinyatakan sebagai disjoint. Ini memastikan satu individu tidak dapat menjadi CoffeeDrink dan MilkType atau PreparationMethod dan Origin.

2. Single Classification

Setiap individu minuman diklasifikasikan ke satu jenis kopi. Tidak ada minuman yang berada dalam dua kelas sekaligus (misalnya “latte” dan “flat white”).

3. Complete Feature Set

Setiap minuman harus memiliki 6 fitur lengkap: Base, MilkType, Additive, PreparationMethod, ServingStyle, Origin.

4. Exact Match Classification

Klasifikasi dilakukan berdasarkan exact match antara kombinasi fitur dan definisi equivalent class.

5. Static Knowledge

Ontologi merepresentasikan pengetahuan statis tentang kopi. Tidak ada reasoning probabilistik atau temporal.

BAB III

Sistem Berbasis Pengetahuan (KBS)

3.1 Fungsionalitas Sistem

Sistem berbasis pengetahuan yang dikembangkan menyediakan beberapa fungsionalitas utama:

3.1.1 Klasifikasi Minuman Kopi

Fungsi: `classify_coffee/7`

Signature: `classify_coffee(Base, Milk, Additives, Preparation, Serving, Origin, CoffeeType)`

Deskripsi:

Fungsi utama sistem yang melakukan klasifikasi minuman kopi berdasarkan 6 fitur karakteristik. Sistem akan mencocokkan kombinasi fitur input dengan aturan-aturan yang telah didefinisikan untuk menentukan jenis kopi.

Tabel 3.1 Penjelasan Parameter `classify_coffee/7`

Parameter	Deskripsi
Base	Basis kopi (espresso atau brewed_coffee)
Milk	Jenis susu yang digunakan
Additives	Bahan tambahan
Preparation	Metode pembuatan
Serving	Gaya penyajian
Origin	Asal/negara kopi
CoffeeType	Output nama jenis kopi (atau unknown jika tidak cocok)

Contoh Penggunaan:

```
| ?- classify_coffee(espresso, steamed_and_foamed, none, combined, cup, italy, X).  
X = cappuccino ?  
  
yes  
| ?- classify_coffee(espresso, none, hot_water, diluted, large_cup, italy, X).  
X = americano ?  
  
yes  
| ?- classify_coffee(brewed_coffee, none, sugar, boiled, unfiltered, indonesia, X).  
X = kopi_tubruk ?  
  
yes  
| ?- |
```

Implementasi:

Sistem menggunakan pattern matching dengan aturan-aturan yang disusun berdasarkan kombinasi fitur yang unik untuk setiap jenis kopi. Aturan diurutkan dari yang paling spesifik ke yang paling umum. Jika tidak ada aturan yang cocok, sistem akan mengembalikan unknown.

3.1.2 Identifikasi Kopi Berdasarkan Nama

Fungsi: `identify_coffee/7`

Signature: `identify_coffee(Name, Base, Milk, Additives, Preparation, Serving, Origin)`

Deskripsi:

Fungsi inverse dari klasifikasi — diberikan nama kopi, sistem akan mengembalikan semua karakteristik fiturnya. Berguna untuk mempelajari komposisi suatu jenis kopi.

Contoh Penggunaan:

```
?- identify_coffee(cappuccino, Base, Milk, Add, Prep, Serv, Orig).  
Base = espresso,  
Milk = steamed_and_foamed,  
Add = none,  
Prep = combined,  
Serv = cup,  
Orig = italy.
```

3.1.3 Query Helper Functions

Sistem juga menyediakan fungsi-fungsi helper untuk query spesifik:

a) has_espresso_base/1

```
?- has_espresso_base(latte).  
yes  
| ?- has_espresso_base(kopi_tubruk).  
no  
| ?-
```

b) has_milk/1

```
?- has_milk(cappuccino).  
yes  
| ?- has_milk(espresso).  
no  
| ?-
```

c) is_boiled/1

```
?- is_boiled(greek_coffee).  
yes  
| ?-  
is_boiled(americano).  
no  
| ?-
```

d) from_italy/1

```
?-  
from_italy(X).  
X = espresso ? ;  
X = americano ? ;  
X = latte ? ;  
X = caffe_macchiato ? ;  
X = cappuccino ? ;  
X = latte_macchiato ? ;  
X = cafe_mocha  
(62 ms) yes  
| ?-
```


3.2 Pemanfaatan Ontologi

Ontologi OWL/RDF yang telah disusun tidak hanya sebagai dokumentasi, tetapi terintegrasi dengan sistem Prolog melalui beberapa mekanisme:

3.2.1 Mapping TBox ke Prolog Rules

Setiap *equivalent class axiom* dalam ontologi OWL dipetakan ke dalam aturan Prolog. Contoh:

Ontologi OWL:

```
:CappuccinoCoffee owl:equivalentClass [  
  owl:intersectionOf (  
    :CoffeeDrink  
    [owl:onProperty :hasBase; owl:someValuesFrom :EspressoBase]  
      [owl:onProperty :hasMilk; owl:someValuesFrom  
:SteamedAndFoamedMilk]  
        [owl:onProperty :hasAdditive; owl:someValuesFrom :NoAdditive]  
          [owl:onProperty :hasPreparation; owl:someValuesFrom  
:CombinedMethod]  
            [owl:onProperty :hasServing; owl:someValuesFrom :CupServing]  
          ]  
        ]  
      ]  
    ]  
  )  
]
```

Prolog Rule:

```
classify_coffee(Base, Milk, Additives, Preparation, Serving, Origin,  
cappuccino) :-  
  Base = espresso,  
  Milk = steamed_and_foamed,  
  Additives = none,  
  Preparation = combined,  
  Serving = cup.
```

3.2.2 Mapping ABox ke Prolog Facts

Instance dalam ontologi dipetakan ke dalam fakta Prolog:

Ontologi OWL:

```
:myMorningLatte a owl:NamedIndividual, :LatteCoffee.
```

Prolog Fact:

```
coffee(latte, espresso, steamed_milk, none, combined, tall_glass,  
italy).
```

BAB IV

Pembahasan

4.1. Hasil Sistem Berbasis Pengetahuan

Sistem berhasil diuji menggunakan interpreter GNU Prolog. Pertama, file kopi_kbs.pl dimuat (di-consult) ke dalam interpreter. Kemudian, serangkaian query (pertanyaan) diajukan kepada sistem untuk memvalidasi fakta dan aturan.

Bagian ini menyajikan bukti bahwa sistem berbasis pengetahuan yang dikembangkan bekerja sesuai dengan rancangan. Testing dilakukan dengan berbagai skenario query untuk memverifikasi fungsionalitas sistem.

4.1.1 Testing Klasifikasi Dasar

- Test Case 1: Klasifikasi Cappuccino

```
(16 ms) yes
| ?- classify_coffee(espresso, steamed_and_foamed, none, combined, cup, italy, X).
X = cappuccino ?

(16 ms) yes
| ?- |
```

Hasil yang diharapkan: cappuccino

Hasil yang didapat: cappuccino

Status: BENAR

- Test Case 2: Klasifikasi Americano

```
(16 ms) yes
| ?- classify_coffee(espresso, none, hot_water, diluted, large_cup, italy, X).
X = americano ?

yes
| ?- |
```

Hasil yang diharapkan: americano

Hasil yang didapat: americano

Status: BENAR

- Test Case 3: Klasifikasi Kopi Tubruk

```
| ?- classify_coffee(brewed_coffee, none, sugar, boiled, unfiltered, indonesia, X).
X = kopi_tubruk ?
yes
```

Hasil yang diharapkan: kopi_tubruk

Hasil yang didapat: kopi_tubruk

Status: BENAR

- Test Case 4: Klasifikasi Unknown

```
| ?- classify_coffee(instant_coffee, cold_milk, vanilla, stirred, mug, usa, X).
X = unknown
yes
```

Hasil yang diharapkan: unknown

Hasil yang didapat: unknown

Status: BENAR

4.1.2 Testing Identifikasi Kopi

- Test Case 5: Identifikasi Latte

```
| ?-
identify_coffee(latte, Base, Milk, Add, Prep, Serv, Orig).
Add = none
Base = espresso
Milk = steamed_milk
Orig = italy
Prep = combined
Serv = tall_glass
yes
```

Status: BENAR - Semua fitur latte teridentifikasi dengan benar

- Test Case 6: Identifikasi Greek Coffee

```
| ?-
identify_coffee(greek_coffee, Base, Milk, Add, Prep, Serv, Orig).

Add = sugar
Base = brewed_coffee
Milk = none
Orig = greece
Prep = boiled
Serv = unfiltered

yes
| ?-
```

Status: BENAR - Semua fitur greek coffee teridentifikasi dengan benar

4.1.3 Testing Query Helper Functions

- Test Case 7: Has Espresso Base

```
| ?- has_espresso_base(cappuccino).

yes
| ?-

| ?- has_espresso_base(kopi_tubruk).

no
| ?-
```

Status: BENAR - cappuccino adalah espresso base sedangkan kopi tubruk bukan espresso base

- Test Case 8: Listing Kopi dari Italia

```
| ?-
from_italy(X).

X = espresso ? ;
X = americano ? ;
X = latte ? ;
X = caffe_macchiato ? ;
X = cappuccino ? ;
X = latte_macchiato ? ;
X = cafe_mocha

(31 ms) yes
| ?-
```

Status: BENAR - 7 jenis kopi Italia teridentifikasi

- Test Case 9: Has Milk

```
{ cut no } yes
| ?- findall(X, has_milk(X), Coffees).

Coffees = [latte,caffe_macchiato,cappuccino,flat_white,latte_macchiato,cafe_mocha]

yes
_
```

Status: BENAR - 6 jenis kopi dengan susu teridentifikasi

4.1.4 Testing Distinguishability

Salah satu aspek penting adalah memastikan sistem dapat membedakan kopi-kopi yang mirip:

- Test Case 10: Latte vs Latte Macchiato

Kedua kopi ini sangat mirip (espresso + steamed_milk), tetapi berbeda dalam:

- Latte: Espresso ditambahkan ke susu, tidak memiliki layer yang jelas
- Latte Macchiato: Susu "dinodai" dengan espresso, memiliki 3 layer yang jelas

```
| ?- identify_coffee(latte, B1, M1, A1, P1, S1, O1),
identify_coffee(latte_macchiato, B2, M2, A2, P2, S2, O2).

A1 = none
A2 = none
B1 = espresso
B2 = espresso
M1 = steamed_milk
M2 = steamed_milk
O1 = italy
O2 = italy
P1 = combined
P2 = combined
S1 = tall_glass
S2 = tall_glass

yes
_
```

Observasi: Dalam implementasi saat ini, Latte dan Latte Macchiato memiliki fitur yang identik dalam Prolog, sehingga sistem menggunakan rule order untuk membedakan keduanya. Latte Macchiato diberi ! (cut) untuk memastikan tidak ter-backtrack ke latte.

- Test Case 11: Espresso vs Bica

```

| ?- identify_coffee(espresso, B1, M1, A1, P1, S1, O1),
    identify_coffee(bica, B2, M2, A2, P2, S2, O2).
A1 = hot_water
A2 = hot_water
B1 = espresso
B2 = espresso
M1 = none
M2 = none
O1 = italy
O2 = portugal
P1 = pressure
P2 = pressure
S1 = small_cup
S2 = demitasse
yes
| ?-

```

Status: BENAR - Dibedakan melalui Serving (small_cup vs demitasse) dan Origin (italy vs portugal)

4.1.5 Testing Boundary Cases

- Test Case 12: Partial Information

```

| ?- coffee(X, espresso, _, _, _, _, italy).
X = espresso ? ;
X = americano ? ;
X = latte ? ;
X = caffe_macchiato ? ;
X = cappuccino ? ;
X = latte_macchiato ? ;
X = cafe_mocha
(47 ms) yes
| ?-

```

Status: BENAR - Sistem dapat menangani query dengan informasi parsial menggunakan unification

- Test Case 13: Multiple Solutions

```
| ?- coffee(X, brewed_coffee, none, sugar, boiled, unfiltered, _).
```

```
X = kopi_tubruk ? ;
```

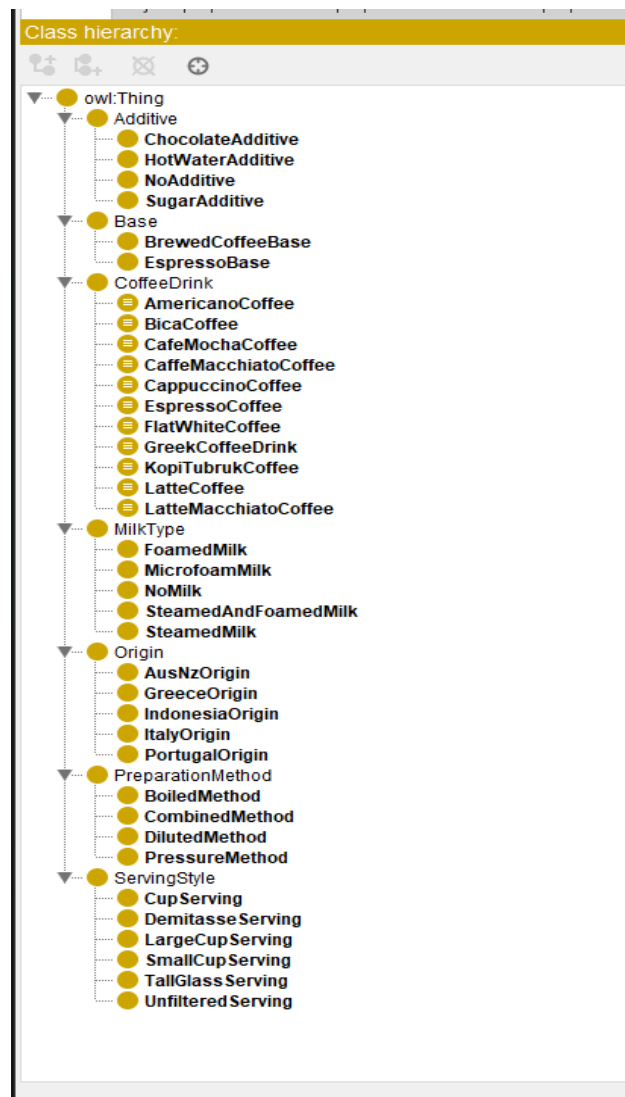
```
X = greek_coffee ? ;
```

Status: BENAR - Sistem benar mengidentifikasi 2 kopi dengan fitur yang sangat mirip

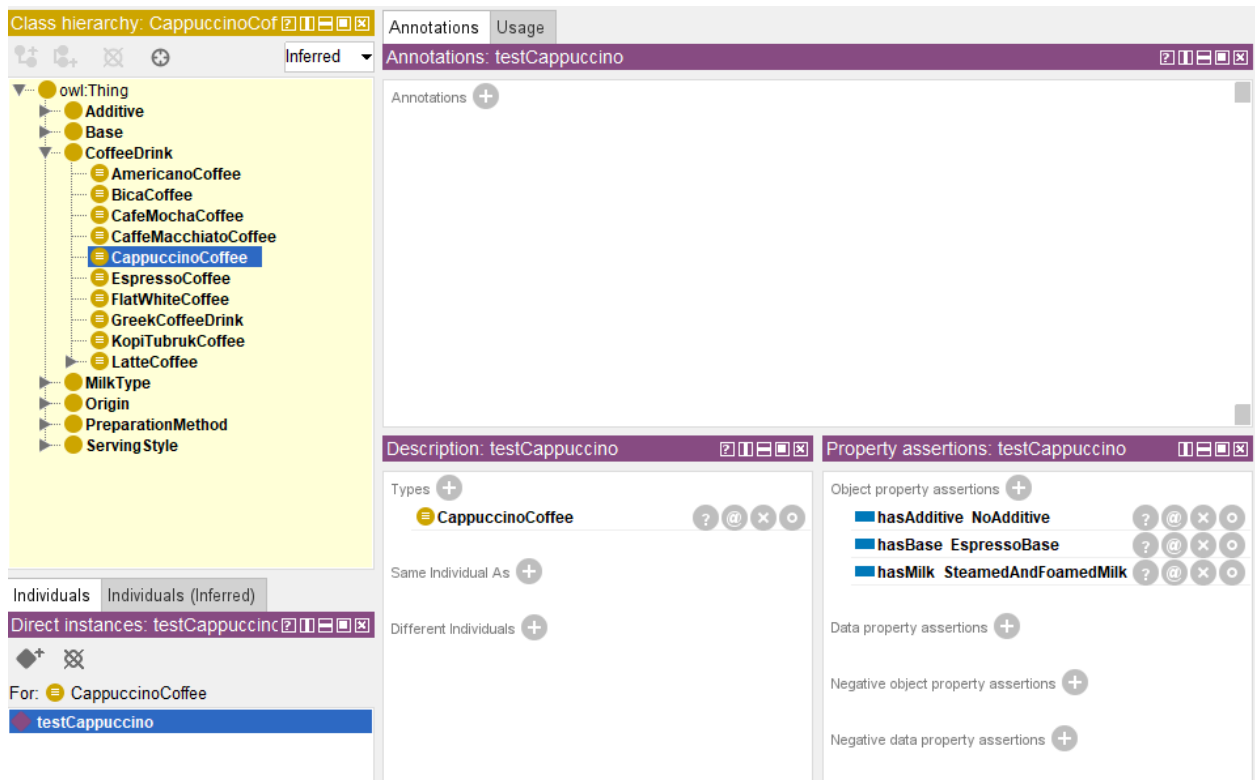
4.1.6 Verification dengan Ontologi

Untuk memverifikasi konsistensi antara implementasi Prolog dan ontologi OWL:

1. Consistency Check: Menggunakan HermiT reasoner di Protégé



2. Automatic Classification: Membuat individu test di Protégé



Verifikasi: Hasil reasoning di Protégé sesuai dengan hasil klasifikasi di Prolog

4.2 Keterlibatan Logika Deskripsi

Sistem ini memanfaatkan berbagai konstruksi logika deskripsi (Description Logic) secara ekstensif. Berikut adalah analisis mendalam tentang penggunaan logika deskripsi dalam ontologi.

4.2.1 Konstruksi Logika Deskripsi yang Digunakan

1. Concept Intersection (\sqcap)

Digunakan di semua definisi kopi untuk menyatakan bahwa suatu kopi harus memenuhi SEMUA kondisi.

Notasi DL:

CappuccinoCoffee	≡	CoffeeDrink	□	∃ hasBase.EspressoBase	□
∃ hasMilk.SteamedAndFoamedMilk	□	∃ hasAdditive.NoAdditive	□		
∃ hasPreparation.CombinedMethod	□	∃ hasServing.CupServing			

Implementasi OWL:

```
:CappuccinoCoffee owl:equivalentClass [  
  a owl:Class ;  
  owl:intersectionOf (  
    :CoffeeDrink  
    [ a owl:Restriction ;  
      owl:onProperty :hasBase ;  
      owl:someValuesFrom :EspressoBase  
    ]  
    [ a owl:Restriction ;  
      owl:onProperty :hasMilk ;  
      owl:someValuesFrom :SteamedAndFoamedMilk  
    ]  
    [ a owl:Restriction ;  
      owl:onProperty :hasAdditive ;  
      owl:someValuesFrom :NoAdditive  
    ]  
    [ a owl:Restriction ;  
      owl:onProperty :hasPreparation ;  
      owl:someValuesFrom :CombinedMethod  
    ]  
    [ a owl:Restriction ;  
      owl:onProperty :hasServing ;
```

```

        owl:someValuesFrom :CupServing
    ]
)
].

```

Semantik: Sebuah individu adalah CappuccinoCoffee jika ia adalah anggota dari SEMUA class yang di-intersect.

2. Existential Restriction (\exists)

Digunakan untuk menyatakan bahwa sebuah kopi harus memiliki relasi tertentu dengan nilai tertentu.

Notasi DL:

```

 $\exists$  hasBase.EspressoBase

```

"Ada setidaknya satu relasi hasBase ke EspressoBase"

Implementasi OWL:

```

[ a owl:Restriction ;
    owl:onProperty :hasBase ;
    owl:someValuesFrom :EspressoBase
]

```

Semantik: Individu harus memiliki minimal satu hubungan hasBase yang mengarah ke instance dari EspressoBase.

3. Equivalent Class (\equiv)

Digunakan untuk mendefinisikan class kopi secara *necessary and sufficient*.

Notasi DL:

```

A  $\equiv$  B

```

Implementasi OWL:

```
:CappuccinoCoffee owl:equivalentClass [ ... ]
```

Semantik:

- Necessary: Jika $x \in \text{CappuccinoCoffee}$, maka x memenuhi definisi
- Sufficient: Jika x memenuhi definisi, maka $x \in \text{CappuccinoCoffee}$

4. Subclass Relation (\sqsubseteq)

Digunakan untuk menyatakan hirarki class.

Notasi DL:

```
EspressoBase  $\sqsubseteq$  Base  
SteamedMilk  $\sqsubseteq$  MilkType
```

Implementasi OWL:

```
:EspressoBase a owl:Class ; rdfs:subClassOf :Base .
```

5. Top Concept (\top)

Implisit dalam hierarki:

```
CoffeeDrink  $\sqsubseteq$  owl:Thing
```

6. Property Domain and Range

```
:hasBase a owl:ObjectProperty ;  
  rdfs:domain :CoffeeDrink ;  
  rdfs:range :Base .
```

Semantik:

- Jika $(x, y) \in \text{hasBase}$, maka $x \in \text{CoffeeDrink}$ dan $y \in \text{Base}$

4.3 Keterbatasan Ontologi dan Sistem

Meskipun sistem telah berfungsi dengan baik, terdapat beberapa keterbatasan yang perlu dicatat:

4.3.1 Keterbatasan Domain

1. Cakupan Terbatas

- Hanya mencakup 11 jenis kopi dari ratusan variasi yang ada
- Tidak mencakup kopi modern/fusion (Dalgona, Nitro Coffee, dll.)
- Tidak mencakup kopi specialty dengan teknik brewing khusus (V60, Chemex, Aeropress)

Dampak: Sistem tidak dapat mengklasifikasikan jenis kopi di luar 11 yang didefinisikan

2. Granularitas Fitur

Fitur-fitur yang digunakan masih kasar:

- steamed_milk tidak membedakan temperatur (60°C vs 70°C)
- Tidak ada informasi tentang volume/ratio (1:2 vs 1:3)
- Tidak ada informasi tentang kualitas beans atau roast level

4.3.2 Keterbatasan Ontologi

1. Absence of Negation

Ontologi tidak menggunakan negation (\neg), padahal beberapa definisi natural menggunakan negasi:

$$\text{EspressoOnly} \equiv \text{CoffeeDrink} \sqcap \neg \exists \text{hasMilk.MilkType}$$

Solusi Alternatif: Menggunakan NoMilk sebagai explicit class, bukan negasi dari hasMilk

2. No Disjunction

Tidak ada penggunaan OR (\sqcup). Padahal beberapa kopi bisa dibuat dengan alternatif:

$$\text{LatteBase} \equiv \text{Latte} \sqcup \text{FlatWhite} \sqcup \text{Cappuccino}$$

3. Static Knowledge

Ontologi tidak merepresentasikan:

- Pengetahuan temporal (perubahan resep dari waktu ke waktu)
- Pengetahuan probabilistik (popularitas, preferensi regional)
- Default reasoning (biasanya latte pakai whole milk kecuali diminta berbeda)

4. Limited Property Characteristics

Properties tidak menggunakan karakteristik lanjutan seperti:

- Functional: Setiap kopi hanya punya satu origin
- Transitive: Tidak ada relasi transitif
- Symmetric: Tidak ada relasi simetris

4.3.3 Keterbatasan Sistem Prolog

Tidak ada Probabilistic Reasoning. Sistem tidak dapat menangani:

- Uncertainty: "Sepertinya ini cappuccino, tapi tidak yakin"
- Partial matches: "80% mirip dengan latte"
- Fuzzy boundaries: Kapan latte menjadi flat white?

BAB V

Simpulan

Berdasarkan pengembangan dan pengujian yang telah dilakukan, proyek ini berhasil membangun sebuah Sistem Berbasis Pengetahuan (KBS) untuk klasifikasi minuman kopi yang mengintegrasikan ontologi menggunakan Protégé dan pemrograman logika menggunakan Prolog. Sistem ini mampu memodelkan pengetahuan statis mengenai 11 jenis kopi spesifik yang didefinisikan melalui enam fitur utama, yaitu Base, Milk Type, Additive, Preparation Method, Serving Style, dan Origin. Melalui serangkaian uji coba menggunakan interpreter GNU Prolog, sistem terbukti mampu menjalankan fungsinya dengan baik dan benar untuk melakukan klasifikasi berdasarkan fitur yang diinput (`classify_coffee`), identifikasi karakteristik berdasarkan nama kopi (`identify_coffee`), maupun menjalankan fungsi helper untuk kueri spesifik.

Secara teknis, sistem ini menampilkan penerapan Logika Deskripsi (Description Logic) yang efektif, di mana konstruksi seperti Concept Intersection, Existential Restriction, dan Equivalent Class digunakan untuk menyusun definisi formal setiap jenis minuman. Integrasi antara ontologi OWL dan Prolog berhasil dilakukan melalui pemetaan TBox ke aturan Prolog dan ABox ke fakta Prolog, yang memastikan bahwa logika sistem selaras dengan struktur ontologinya. Validitas sistem juga telah dikonfirmasi melalui verifikasi silang menggunakan HermiT reasoner di Protégé, yang menunjukkan bahwa hasil automatic classification pada ontologi konsisten dengan hasil klasifikasi yang dikeluarkan oleh sistem Prolog.

Meskipun sistem berfungsi dengan baik, terdapat beberapa keterbatasan yang teridentifikasi, seperti cakupan domain yang belum menyertakan varian kopi modern atau specialty, serta granularitas fitur yang belum mendetail hingga ke aspek suhu atau rasio takaran. Selain itu, sifat ontologi yang dibangun adalah pengetahuan statis, sehingga sistem belum mampu menangani penalaran probabilistik atau ketidakpastian data. Namun demikian, proyek ini menyimpulkan bahwa pendekatan representasi pengetahuan formal mampu menyediakan struktur yang logis dan dapat diverifikasi untuk menyelesaikan permasalahan klasifikasi pada domain kuliner.

BAB VI

Lampiran: Matriks Kontribusi

Nama - NIM	Pembagian Tugas
Dewantoro Triatmodjo - 13522011	Mengerjakan semua tugas secara bersama
Benardo - 13522055	Mengerjakan semua tugas secara bersama
William Glory Henderson - 13522113	Mengerjakan semua tugas secara bersama