



# Peran Ontologi pada Kecerdasan Buatan

ADRIAN NURADIANSYAH

RESEARCH ASSOCIATE AT  
TECHNISCHE UNIVERSITÄT DRESDEN, GERMANY

[adrian.nuradiansyah@tu-dresden.de](mailto:adrian.nuradiansyah@tu-dresden.de)

<https://lat.inf.tu-dresden.de/~adrian>

# Apa itu Ontologi?

- ▶ AI System: bertindak dan berperilaku secara rasional
- ▶ Untuk menjadi rasional, maka **sistem** (apapun!) **butuh pengetahuan**
- ▶ Bagaimana **merepresentasikan pengetahuan?** dan Bagaimana melakukan **penalaran (*reasoning*)** dari **pengetahuan** tersebut?
- ▶ Penalaran: **memperoleh informasi baru** dari pengetahuan yang tertulis secara eksplisit  
→ Cabang ilmu: *Knowledge Representation and Reasoning* (KRR)

## Ontologi:

- ✓ Sebuah *framework* untuk merepresentasikan pengetahuan pada suatu domain
- ✓ Model dari suatu aspek di dunia

# Apa itu Ontologi?

Berdasarkan komponennya, ontologi dapat dilihat sebagai berikut:

- ▶ Memiliki **kosakata (vocabulary)** pada domain tersebut
  - ❖ Kosakata: *classes, relationship (property), object (individual)*
  - ❖ *classes (e.g., Male, Female, ...), property (e.g., hasWife, hasChild, ...), individual (e.g., Doni, Ella, ...)*
- ▶ Memberikan **semantik/arti** pada setiap kosakata tersebut
  - ❖ *OWL (Web Ontology Language)\**: Salah satu ontology language
  - ❖ Umumnya diformalisasikan dengan sebuah logic, e.g., Description Logic
- ▶ Terdiri dari minimal dua bagian
  - ❖ Himpunan **aturan-aturan** yang mendeskripsikan struktur pada ontologi
  - ❖ Himpunan **fakta-fakta** yang mendeskripsikan situasi tertentu pada ontologi

\*<https://www.w3.org/TR/owl-features/>

# Aturan dan Fakta

- ▶ **Aturan:** Mendeskripsikan **struktur** pada ontologi

Class: Wife  
SubClassOf: Female

Class: Male and hasWife\*  
EquivalentTo: Husband

Class: Husband  
EquivalentTo: hasWife only Female

Class: Parent  
SubClassOf: Husband or Wife

ObjectProperty: hasChild  
InverseProperty: isChildOf

- ▶ **Fakta:** Mendeskripsikan **situasi-situasi** apa saja yang terjadi pada ontologi tersebut

Individual: Doni  
Types: Husband  
Facts: hasWife Ella

Individual: Ferdi  
Types: Child  
Facts: isChildOf Doni

Individual: Fina  
Types: Child  
Facts: hasBrother Ferdi

\* *Abused notation used only for the sake of convenience.*

Umumnya, sebuah property *hasWife* ditulis dalam suatu konsep dengan menggunakan *quantifier some hasWife* atau *all hasWife*

# Ontology = Database?

- ▶ Aturan pada ontologi memiliki kemiripan dengan **skema DB**
  - ❖ Skema DB mendeskripsikan struktur dan constraint pada data yang disimpan
  - ❖ Metadata: data yang menyediakan informasi tentang data pada DB
- ▶ Fakta-fakta pada ontologi memiliki kemiripan dengan **data pada DB**
  - ❖ Menginstansiasi skema
  - ❖ Harus memenuhi constraint yang diberlakukan oleh skema DB
- ▶ Namun, ternyata ada banyak **perbedaan...**

# Database vs Ontologi

## Database

- ▶ Unique Name Assumption (UNA)
  - ❖ Setiap individu/object memiliki nama yang berbeda
- ▶ Closed World Assumption
  - ❖ *Missing information treated as false*
  - ❖ *Complete information*
- ▶ Skema berfungsi sebagai **constraint** yang harus dipenuhi oleh data

## Ontologi

- ▶ No UNA
  - ❖ Satu individu/object bisa memiliki lebih dari satu nama
- ▶ Open World Assumption
  - ❖ *Missing information treated as unknown*
  - ❖ *Incomplete information*
- ▶ Aturan pada ontologi berfungsi sebagai constraint dan juga layaknya **implikasi**
  - ❖ *Inference rule*
  - ❖ *Entail implicit information*

# Database vs Ontologi

Diberikan fakta:

Individual: Doni

Facts: hasWife Ella

Facts: hasChild Ferdi

Facts: hasChild Fina

Query: Apakah Elsa adalah istri Doni?

DB: Tidak  
Ontology: Tidak tahu

Tidak ada yang mengatakan bahwa Elsa berbeda dengan Ella

Ada interpretasi yang mengatakan bahwa Doni memiliki istri bernama Ella dan Elsa sama dengan Ella

# Database vs Ontology

Diberikan fakta:

Individual: Doni

Facts: hasWife Ella

Facts: hasChild Ferdi

Facts: hasChild Fina

Query: Apakah Elsa adalah istri Doni?

DB: Tidak

Ontology: Tidak tahu

DifferentIndividuals: Ella Elsa

Tidak ada yang mengatakan Doni hanya boleh punya satu istri

Masih ada interpretasi yang mengatakan Doni memiliki istri bernama Elsa, meski Elsa berbeda dengan Ella



# Database vs Ontology

Diberikan fakta:

Individual: Doni

Types: Husband

Facts: hasWife Ella

Facts: hasChild Ferdi

Facts: hasChild Fina

Class: Husband

SubClassOf: hasWife exactly one

DifferentIndividuals: Ella Elsa

Query: Apakah Elsa adalah istri Doni?

DB: Tidak

Ontology: Tidak!!

# Database vs Ontology

Diberikan fakta:

Individual: Doni

Facts: hasWife Ella

Facts: hasChild Ferdi

Facts: hasChild Fina

Query: Berapa jumlah anak Doni?

DB: 2

Ontology: minimal 1

Tidak ada yang mengatakan bahwa Ferdi berbeda dengan Fina

Masih ada interpretasi dimana Ferdi = Fina dan Doni hanya punya satu anak di interpretasi tsb

# Database vs Ontology

Diberikan fakta:

Individual: Doni

Facts: hasWife Ella

Facts: hasChild Ferdi

Facts: hasChild Fina

Query: Berapa jumlah anak Doni?

DB: 2

Ontology: minimal 2

DifferentIndividuals: Ferdi Fina

Tidak ada yang mengatakan bahwa Doni **hanya** punya dua anak bernama Ferdi dan Fina

Masih ada interpretasi dimana Doni punya anak bernama Ferdi, Fina, dan anak lain yang tidak perlu punya nama

# Database vs Ontology

Diberikan fakta:

Individual: Doni

Facts: hasWife Ella

Facts: hasChild Ferdi

Facts: hasChild Fina

Facts: hasChild ONLY Ferdi OR Fina

Query: Berapa jumlah anak Doni?

DifferentIndividuals: Ferdi Fina

DB: 2

Ontology: 2!

# Database Query Answering

- ▶ Skema *tidak memiliki peran* selama proses *query answering*
  - ❖ Skema hanya berisikan *constraints* yang harus dipenuhi oleh data DB
- ▶ Proses *query answering* *hanya mempertimbangkan data* pada DB
  - ❖ Mirip seperti *model checking*
  - ❖ Implikasi dari paradigma *Closed World Assumption*
- ▶ Dapat diimplementasikan secara efisien
  - ❖ *Worst case complexity* bisa mencapai logspace menurut ukuran input data

# Ontology Query Answering

- ▶ Aturan pada ontologi **memiliki banyak peran**
  - ❖ Memberikan informasi implisit sebagai jawaban dari *query* yang diberikan
- ▶ Query answering pada ontologi berbasiskan *logical entailment*
  - ❖ Ontologi dan query dipandang sebagai himpunan formula logika
  - ❖ *Logical entailment*: hubungan dua formula logika bernilai benar jika formula yang satu merupakan *logical consequence* dari formula yang kedua
- ▶ Bisa memiliki *worst case complexity* yang sangat tinggi
  - ❖ Alasan sederhana: **mempertimbangkan semua kemungkinan interpretasi** yang sesuai dengan ontologi
  - ❖ Untuk OWL, *query answering problem* adalah **NP-hard** menurut ukuran input data  
→ Tidak ada *polynomial-time algorithm* untuk menyelesaikan *query answering* pada OWL.
  - ❖ Untuk beberapa “fragment” dari OWL, implementasi *query answering* masih dapat dioptimasi dan memiliki kompleksitas yang efisien

# Standard Reasoning Tasks pada Ontologi

Jenis penalaran apa yang user umumnya lakukan terhadap ontologi?

► **Pengecekan konsistensi**

Apakah input ontologi tidak mengandung informasi yang kontradiktif? E.g.,

Children SubClassOf NOT Married

Husband SubClassOf Married

IsChildOf SubClassOf Children    IsChildOf(Doni, Candra)

Husband(Doni)

Doni itu anak, tapi kok sudah menikah? 🤔

# Standard Reasoning Tasks pada Ontologi

Jenis penalaran apa yang user umumnya lakukan terhadap ontologi?

▶ **Subsumption**

Apakah class A **lebih spesifik/general/ekuivalen** terhadap class B menurut input ontology?  
E.g., Husband SubClassOf HeadOfFamily HeadOfFamily SubClassOf TaxPayer  
apakah setiap suami adalah pembayar pajak? 🙄

▶ **Pengecekan keanggotaan (*instance checking*)**

Apakah suatu individu merupakan **anggota dari suatu kelas** menurut input ontology?  
E.g., Doni's example → apakah Doni hanya punya dua anak?



# Non-Standard Reasoning Tasks pada Ontologi

Penalaran apa lagi yang bisa kita lakukan lebih terhadap ontologi?

- ▶ Most Specific Class/Concept of an Individual
  - ❖ Membangun ontologi secara bottom-up
  - ❖ Diberikan suatu ontologi dan individu, tentukan class dari individu tersebut yang paling spesifik!
  - ❖ *Apakah ada class semacam itu* untuk input individual menurut ontologi yang diberikan?
- ▶ Computing Repairs
  - ❖ Diberikan *consistent ontology*  $O$  yang ternyata mengandung informasi rahasia, carikan sebuah *ontologi (repair)*  $O'$  yg *logically entailed by*  $O$  dan *tidak mengandung informasi rahasia* sedemikian hingga *tidak ada repair*  $O''$  lain yang lebih informatif daripada  $O'$ .
  - ❖ *Apakah ada repair* semacam itu?
  - ❖ Empty ontology juga sebuah repair, tapi tentu bukan itu yg kita inginkan

# Kapan dan Dimana Ontologi/DB Dapat Digunakan?

## Ontologi

- ▶ Skema/himpunan aturan berukuran besar
  - ❖ Dapat digunakan untuk reasoning
- ▶ Tidak membutuhkan informasi yang komplit
- ▶ Siap dengan performa komputasi yang tidak cepat/tidak efisien

## Database

- ▶ Skema berukuran kecil/ sederhana
- ▶ Membutuhkan informasi yang komplit
  - ❖ Contoh: Database transaksi pemesanan tiket pesawat
- ▶ Performa komputasi harus seefisien mungkin

# Aplikasi-Aplikasi Ontologi

- ▶ Banyak digunakan di bidang Semantic Web, Biomedis, Life Sciences, Ilmu Perpustakaan, dll  
e.g., SNOMED, Gene Ontology, GALEN, Foundational Model of Anatomy, National Cancer Institute (NCI) Thesaurus, dll.
- ▶ **Ontology editor**: aplikasi untuk membuat ontologi  
e.g., Protégé, SWOOP, TopBraidComposer, OWLGrEd, dll.
- ▶ **Ontology reasoners**: aplikasi untuk melakukan penalaran pada ontologi  
e.g., Pellet, ELK, FaCT++, CEL, Racer, RDFox, dll.

# Literature

Sebagian besar isi dari slides ini diadaptasi dengan bahasa Indonesia dari “*Ontologies and Databases*” by Ian Horrocks  
link: <http://www.cs.ox.ac.uk/ian.horrocks/Seminars/download/onto-db.ppt>

Literatur terkait lainnya:

- ▶ Horrocks, I. *What are ontologies good for?*. Evolution of Semantic Systems. 2013
- ▶ Baader F., Horrocks, I., Lutz C., Sattler U. *An Introduction to Description Logic*. Cambridge University Press. 2017.
- ▶ Turhan, A.Y., *On the Computation of Common Subsumers in Description Logics*. Dresden University of Technology. 2007.
- ▶ Baader F., Kriegel F., Nuradiansyah A., Peñaloza R. *Making Repair in Description Logics More Gentle*. KR 2018.