# Model Pengetahuan Berbasis Ontologi pada Domain Big Data di Perguruan Tinggi

Yunizar Fahmi

17917131

*Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer*

*Konsentrasi Sistem Informasi Enterprise*

*Program Studi Informatika Program Magister*

*Fakultas Teknologi Industri*

*Universitas Islam Indonesia*

2021

**Lembar Pengesahan Pembimbing**

**Model Pengetahuan Berbasis Ontologi pada Domain Big Data di Perguruan Tinggi**

Yunizar Fahmi

17917131

Yogyakarta, Desember 2021

Pembimbing

Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.

**Lembar Pengesahan Penguji**

**Model Pengetahuan Berbasis Ontologi pada**
**Domain Big Data di Perguruan Tinggi**

Yunizar Fahmi

17917131

Yogyakarta, Desember 2021

Tim Penguji,

Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.
Ketua

Dr. Raden Teduh Dirgahayu, S.T., M.Sc.
Anggota I

Ahmad Raf'ie Pratama, S.T., MIT., Ph.D
Anggota II

Mengetahui,
Ketua Program Studi Informatika Program Magister
Universitas Islam Indonesia

Izzati Muhimmah, S.T., M.Sc., Ph.D

## Abstrak

### Model Pengetahuan Berbasis Ontologi pada Domain Big Data di Perguruan Tinggi

Fenomena *big data* juga dirasakan dampaknya oleh sektor perguruan tinggi. Bagi institusi perguruan tinggi, data menjadi aset yang sangat berharga yang dapat dimanfaatkan untuk memperoleh informasi yang berguna sebagai dukungan untuk pengambilan keputusan, pemecahan masalah, perencanaan serta perbaikan layanan bagi organisasi. Pada dasarnya, berbagai studi yang menghasilkan pengetahuan untuk memanfaatkan *big data* sudah cukup banyak tersedia. Namun, pengetahuan tersebut belum dikelola secara baik sehingga tidak sedikit perguruan tinggi yang masih belum memahami arah pemanfaatan *big data*. Berdasarkan masalah tersebut, ontologi dapat digunakan untuk memodelkan pengetahuan. Pada penelitian ini, akan dikembangkan sebuah model pengetahuan berbasis ontologi mengenai pemanfaatan *big data* di perguruan tinggi berdasarkan konsep *analytics*, *data source*, dan *platforms*. Proses pemodelan ontologi pada penelitian ini dibagi menjadi 3 (tiga) fase, yaitu: (1) Konseptualisasi; (2) Implementasi; kemudian (3) Evaluasi. Sumber pengetahuan berasal dari panduan umum DAMA-DMBOK (*Data Management Body of Knowledge*) dan ensiklopedia dari Encyclopedia of Big Data Technologies serta studi *review/survey* mengenai *big data* di perguruan tinggi. Model ontologi direpresentasikan dalam format OWL (*Web Ontology Language*). Berdasarkan hasil evaluasi pengukuran ontologi menggunakan OntoQA *schema metrics*, diperoleh nilai RR (*Relationship Richness*) sebesar 0.62, artinya model ontologi memuat keberagaman informasi; nilai IR (*Inheritance Richness*) sebesar 1.78, artinya model ontologi termasuk kategori cukup spesifik, dan nilai AR (*Attribute Richness*) sebesar 0.06 bahwa informasi yang dimuat dapat ditingkatkan lagi. Melalui analisis hasil uji coba *query*, ontologi dinilai mampu menunjukkan pengetahuan yang tersimpan di dalamnya dengan akurat berdasarkan validasi yang menyesuaikan hasil *query* dengan ontologi.



**Kata kunci**

Ontologi, *Big Data*; Perguruan Tinggi.

## Abstract

### Ontology-based Knowledge Model on the Domain of Big Data in Higher Education

The big data phenomenon is also having an impact on the higher education sector. The readiness of education sector is non-negotiable for the face of the big data era so that organizations are able to adapt. Thus, knowledge to utilize big data needs to be owned by higher education institutions in order to innovate. Basically, there are quite a lot of studies that produce models as knowledge to take advantage of big data. However, reviewing the research that has been done previously, it is also necessary to conduct research that discusses the development of models to represent knowledge about big data analytics in higher education institutions. Knowledge that has been represented as a knowledge model can then be interpreted and stored as a knowledge base in a knowledge management system so that knowledge can be managed and utilized better. Based on these problems, ontology can be used to model knowledge. This research will develop an ontology-based knowledge model related to the use of big data in the education sector, especially higher education based on the concepts of analytics, data sources, and platforms. The ontology modeling process in this study is divided into 3 (three) phases: 1) Conceptualization; 2) Implementation; and 3) Evaluation. Based on the evaluation results of ontology measurements using schema metrics, the RR (Relationship Richness) value is 0.62, meaning that the ontology model contains a variety of information; the IR (Inheritance Richness) value is 1.78, meaning that the ontology model belongs to a fairly specific category, and the AR (Attribute Richness) value is 0.06, meaning that the information contained can be improved further. Based on the results of the query test analysis, the ontology is considered capable of retrieving the knowledge stored in it accurately based on validation that adjusts the query results with the ontology.



**Keywords**

Ontology, Big Data; Higher Education.

## Pernyataan Keaslian Tulisan

Dengan ini saya menyatakan bahwa tesis ini merupakan tulisan asli dari penulis, dan tidak berisi material yang telah diterbitkan sebelumnya atau tulisan dari penulis lain terkecuali referensi atas material tersebut telah disebutkan dalam tesis. Apabila ada kontribusi dari penulis lain dalam tesis ini, maka penulis lain tersebut secara eksplisit telah disebutkan dalam tesis ini.

Dengan ini saya juga menyatakan bahwa segala kontribusi dari pihak lain terhadap tesis ini, termasuk bantuan analisis statistik, desain survei, analisis data, prosedur teknis yang bersifat signifikan, dan segala bentuk aktivitas penelitian yang dipergunakan atau dilaporkan dalam tesis ini telah secara eksplisit disebutkan dalam tesis ini.

Segala bentuk hak cipta yang terdapat dalam material dokumen tesis ini berada dalam kepemilikan pemilik hak cipta masing-masing. Apabila dibutuhkan, penulis juga telah mendapatkan izin dari pemilik hak cipta untuk menggunakan ulang materialnya dalam tesis ini.

Yogyakarta, Desember 2021




Yunizar Fahmi, S.Kom.

## Daftar Publikasi

**<u>Publikasi yang menjadi bagian dari tesis</u>**

Publikasi berikut menjadi bagian dari penulisan tesis ini

Fahmi, Y., & Fudholi, D., H. (2021). Model Pengetahuan Berbasis Ontologi pada Domain Big Data di Perguruan Tinggi. *JURNAL MEDIA INFORMATIKA BUDIDARMA*.

Sitasi publikasi 1

| Kontributor | Jenis Kontribusi |
|---|---|
| Yunizar Fahmi | Mendesain eksperimen (70%)<br>Menulis *paper* (100%) |
| Dhomas Hatta Fudholi | Memberi ide dan saran (30%)<br>Telaah artikel |

## Halaman Kontribusi

Penelitian ini tentunya tidak terlepas dari bimbingan dan masukan dari berbagai pihak. mulai dari pra penelitian, seminar proposal, seminar kemajuan hingga seminar pendadaran. Pihak-pihak tersebut di antaranya:

1. Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.
2. Dr. Raden Teduh Dirgahayu, S.T., M.Sc.
3. Ahmad Raf'ie Pratama, S.T., MIT., Ph.D.
4. Dr. Syarif Hidayat, S.T., MIT.
5. Ridho Rahmadi, S.Kom., M.Sc., Ph.D.
6. Hendrik, S.T., M.Eng.
7. Almed Hamzah, S.T., M.Eng.

**Halaman Persembahan**

بسم الله الرحمن الرحيم

Melalui berkah dan ridha Allah SWT, dengan mengucapkan syukur Alhamdulillah, karya ini penulis persembahkan kepada kedua orang tua, Bapak Tommy P. Badron, S.Pi. dan Ibu Faridah Ariyani, S.Pd. yang tanpa henti selalu mendukung dan mendoakan penulis selama menyelesaikan pendidikan jenjang magister ini. Semoga Allah SWT meridhoi kita semua. Aamiin.

## Kata Pengantar

*Assalamu'alaikum Warahmatullahi Wabarakatuh*

Puji syukur penulis panjatkan ke hadirat Allah SWT yang telah memberikan kemudahan melalui rahmat dan karunia-Nya. Hanya karena ridhoNya sehingga penulis dapat menyelesaikan penelitian ini. Shalawat serta salam semoga senantiasa tercurah kepada Rasulullah Muhammad SAW sebagai teladan terbaik bagi umat manusia. Laporan tesis ini disusun untuk memenuhi syarat menyelesaikan pendidikan pada Program Studi Informatika Program Magister, Konsentrasi Sistem Informasi Enterprise, Universitas Islam Indonesia. Dalam penyusunan tesis ini, penulis menyadari tanpa bantuan dan bimbingan dari berbagai pihak, sulit bagi penulis untuk menyelesaikan laporan tesis ini. Oleh karena itu, pada kesempatan ini penulis ingin mengungkapkan rasa terima kasih kepada:

1. Bapak Fathul Wahid, S.T., M.Sc., Ph.D., selaku rektor Universitas Islam Indonesia yang memberikan kesempatan kepada penulis untuk menimba ilmu di Universitas Islam Indonesia.
2. Ibu Izzati Muhimmah, M.Sc., Ph.D., selaku Ketua Jurusan Informatika Program Magister, Fakultas Teknologi Industri, Universitas Islam Indonesia, yang selalu memberikan semangat kepada seluruh mahasiswa agar segera menyelesaikan tesis.
3. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D., selaku Dosen Pembimbing yang telah menyediakan ilmu, waktu, dan bimbingannya kepada penulis selama proses penelitian.
4. Seluruh dosen, staf administrasi serta segenap civitas akademika Program Studi Informatika Program Magister Universitas Islam Indonesia yang telah membantu secara langsung maupun tidak langsung selama penyelesaian studi.
5. Kedua orang tua dan saudara penulis yang tanpa henti selalu mendoakan dan mendukung baik secara moral maupun material dalam menyelesaikan studi di Universitas Islam Indonesia.
6. Seluruh rekan-rekan mahasiswa sebagai kawan seperjuangan penulis dalam menyelesaikan pendidikan di Program Studi Informatika Program Magister khususnya Konsentrasi Sistem Informasi Enterprise.
7. Semua pihak yang telah membantu penulis dalam menyelesaikan studi, yang tidak dapat dituliskan satu persatu.

Penulis sangat menyadari terdapat banyak kekurangan dalam penyusunan laporan tesis ini karena keterbatasan dan pengalaman penulis. Oleh karena itu, penulis berharap kritik, saran, dan masukan yang bersifat membangun guna menyempurnakan penelitian ini. Akhir kata, penulis ucapkan terima kasih, semoga laporan penelitian ini dapat memberikan inspirasi maupun manfaat bagi semua pihak.

*Wassalamu'alaikum Warahmatullahi Wabarakatuh*

Yogyakarta, Desember 2021



Yunizar Fahmi

## Daftar Isi

Halaman

Lembar Pengesahan Pembimbing ..... i
Lembar Pengesahan Penguji.....iii
Abstrak.....iii
Abstract.....iv
Pernyataan Keaslian Tulisan ..... v
Daftar Publikasi .....vi
Halaman Kontribusi.....vii
Halaman Persembahan .....viii
Daftar Isi .....xi
Daftar Tabel.....xiv
Daftar Gambar .....xv
Glosarium .....xvi
BAB 1 Pendahuluan ..... 1
1.1 Latar Belakang..... 1
1.2 Rumusan Masalah ..... 3
1.3 Batasan Masalah..... 3
1.4 Tujuan Penelitian..... 3
1.5 Manfaat Penelitian..... 4
1.6 Sistematika Penulisan..... 4
BAB 2 Landasan Teori ..... 5
2.1 Pemodelan Pengetahuan..... 5
2.2 Rekayasa Ontologi..... 5
2.2.1 Pengertian Ontologi ..... 5

2.2.2 Elemen Ontologi ..... 6
2.2.3 Formalisasi Ontologi ..... 7
2.2.3 SPARQL ..... 9
2.2.4 Ontology Development 101 ..... 9
2.3 Big Data di Bidang Pendidikan ..... 11
2.3.1 Karakteristik Big Data ..... 11
2.3.2 *Big Data Analytics* ..... 12
2.3.2 *Big Data* Pendidikan ..... 14
2.4 Penelitian Terdahulu ..... 15
BAB 3 Metodologi Penelitian ..... 18
3.1 Metodologi Konseptualisasi ..... 18
3.1.1 Pemetaan Literatur ..... 18
3.1.2 Penentuan Taksonomi ..... 19
3.1.3 Penentuan Properti ..... 19
3.2 Metodologi Implementasi ..... 19
3.2.1 Pembentukan Kelas & Properti ..... 19
3.2.2 Pembentukan Objek ..... 20
3.3 Metodologi Evaluasi ..... 20
3.3.1 Analisis Ontologi ..... 20
3.3.2 Analisis *Query* ..... 22
BAB 4 Hasil dan Pembahasan ..... 23
4.1 Konseptualisasi ..... 23
4.1.1 Pemetaan Literatur ..... 23
4.1.2 Taksonomi ..... 29
4.1.2 Properti ..... 31

4.2 Implementasi .....31
4.2.1 Pembentukan Kelas & Properti .....32
4.2.2 Pembentukan Objek Kelas .....34
4.3 Evaluasi .....39
4.3.1 Analisis Ontologi .....39
4.2.2 Analisis *Query* .....40
BAB 5 Kesimpulan dan Saran .....44
5.1 Kesimpulan .....44
5.2 Saran .....44
Daftar Pustaka .....46
LAMPIRAN .....50

## Daftar Tabel

Halaman

Tabel 2.1. Penelitian Big Data di Perguruan Tinggi ..... 6

Tabel 3.1 Panduan Umum *Big Data* ..... 18

Tabel 3.1 Tabel 3.2 Studi *Big Data & Analytics* ..... 18

Tabel 3.1 Panduan Umum *Big Data* ..... 19

Tabel 3.3 Skala Klasifikasi Nilai RR ..... 21

Tabel 3.4 Skala Klasifikasi Nilai IR ..... 21

Tabel 3.4 Skala Klasifikasi Nilai IR ..... 21

Tabel 3.4 Skala Klasifikasi Nilai IR ..... 21

Tabel 4.1 Kamus Konsep *"Analytics"* ..... 24

Tabel 4.2 Kamus Konsep *"Data Source"* ..... 25

Tabel 4.3 Kamus Konsep *"Platforms"* ..... 25

Tabel 4.4 Daftar Kelas dan Hierarki Ontologi ..... 26

Tabel 4.5 Daftar Properti ..... 27

Tabel 4.6 *Ontology Metrics* ..... 34

Tabel 4.7 Hasil Pengukuran Ontologi ..... 35

## Daftar Gambar

Halaman

Gambar 2.1 Representasi Kelas-Properti-Objek .....10

Gambar 2.2 Semantic Web Stack .....12

Gambar 2.3 Bagan 5V's Big Data.....14

Gambar 2.4 Representasi Kelas-Properti-Objek .....10

Gambar 2.5 Empat Jenis Analytics.....16

Gambar 3.1 Alur Pengembangan Ontologi .....18

Gambar 4.1 Taksonomi Ontologi .....27

Gambar 4.2 Deklarasi Skema OWL/XML .....28

Gambar 4.3 Kelas dalam Kode OWL/XML.....29

Gambar 4.4 Properti dalam Kode OWL/XML .....30

Gambar 4.5 Visualisasi Kelas dan Properti .....30

Gambar 4.6 Objek dalam Kode OWL/XML .....31

Gambar 4.7 Kode OWL/XML dari Relasi Setiap Objek.....32

Gambar 4.8 Visualisasi Relasi Antar Objek .....33

Gambar 4.9 Hasil Uji Konsistensi pada Ontologi .....35

Gambar 4.10 Kondisi Uji Konsistensi Gagal .....36

Gambar 4.11 Uji Coba *Query* StudentsPerformance.....38

Gambar 4.12 Relasi dari StudentsPerformance .....38

## Glosarium

| | |
|---|---|
| AR | : Attribute Richness |
| DAMA | : Data Management Association |
| DMBOK | : Data Management Body of Knowledge |
| IR | : Inheritance Richness |
| OWL | : Web Ontology Language |
| OntoQA | : Ontology Quality Analysis |
| RDF | : Resource Description Framework |
| RR | : Relationship Richness |
| SPARQL | : SPARQL Protocol and RDF Query Language |
| XML | : Extensible Markup Language |

# BAB 1
# Pendahuluan

## 1.1 Latar Belakang

Saat ini adalah era di mana data secara terus menerus dihasilkan. Hal tersebut menciptakan sebuah fenomena yang disebut dengan istilah *big data* . Salah satu sektor yang terdampak fenomena tersebut adalah perguruan tinggi (Chaurasia & Rosin, 2017). Data menjadi aset yang sangat berharga bagi organisasi, termasuk bagi institusi perguruan tinggi seiring dengan berkembangnya sumber daya data (*data resource*) pada era digital saat ini (Lee, 2017; Fischer et al., 2019). Seperti halnya aset organisasi lainnya (misal SDM, modal, inventaris), data yang terus berkembang ini perlu dimanfaatkan agar memperoleh informasi yang berguna sebagai dukungan untuk pengambilan keputusan, pemecahan masalah, perencanaan serta perbaikan layanan bagi organisasi (Kurilovas, 2018).

Kesiapan institusi perguruan tinggi sangat diperlukan dalam menghadapi era *big data* agar organisasi dapat beradaptasi. Pada tahun 2012, Pusat Analisis dan Penelitian EDUCAUSE melakukan survei di mana 86% responden percaya bahwa *big data analytics* akan menjadi semakin penting bagi keberhasilan sektor pendidikan tinggi di masa depan (Saggi & Jain, 2018). Dengan demikian, pengetahuan untuk memanfaatkan *big data* perlu dimiliki oleh perguruan tinggi agar dapat berinovasi. Salah satu manfaat yang dapat diperoleh oleh perguruan tinggi yang memanfaatkan *big data analytics* yaitu mengukur kinerja peserta didik melalui *big data* dari lingkungan belajar virtual (*Virtual Learning Environtment / VLE*). Informasi seperti jam akses, jumlah sumber daya yang diakses, dan waktu penyerahan tugas akan memberikan representasi kinerja peserta didik tersebut (Punlumjeak et al., 2017; Waheed et al., 2020).

Pengetahuan menjadi bekal bagi organisasi untuk memudahkan pengambilan keputusan dengan lebih akurat. Di samping itu, pengetahuan juga memicu inovasi dan perubahan yang diperlukan oleh organisasi. Tanpa pengetahuan yang memadai, proses pengambilan keputusan, penciptaan inovasi dan perubahan organisasi akan lebih sulit dicapai (De Long, 1997; Hung et al., 2010; De Silva, 2018). Bagi institusi perguruan tinggi, pengetahuan mengenai potensi aset data tersebut sangat diperlukan sehingga data dapat dimanfaatkan dengan benar sebagai dasar evaluasi kinerja serta pengambilan keputusan yang didasarkan oleh data (*data-driven decision making*) guna meningkatkan

efisiensi operasional dan efektivitas pelayanan kepada sivitas akademika, penghematan biaya, serta peningkatan kualitas pelayanan pada *stakeholder*-nya (Tulasi, 2013; Mandinach, 2015).

Beberapa studi terdahulu yang membangun model untuk memahami potensi pemanfaatan dari adanya fenomena *big data* yang dapat menjadi pengetahuan bagi perguruan tinggi. Sebagai contoh, studi yang dilakukan oleh Attaran et al. (2009) mengembangkan model konseptual untuk menganalisis keberhasilan implementasi *big data analytics* pada perguruan tinggi. Studi lain yang dilakukan oleh Ashaari et al. (2020) Mengembangkan model kerangka kerja untuk mengetahui kapabilitas teknologi yang mendukung penerapan *big data analytics* di perguruan tinggi. Serta penelitian lainnya yang mengembangkan model untuk menangkap potensi *big data* di bidang pendidikan.

Namun, meninjau dari berbagai studi yang telah ada, perlu dilakukan juga penelitian yang membahas pengembangan model untuk merepresentasikan pengetahuan mengenai *big data analytics* di perguruan tinggi. Pemodelan pengetahuan adalah serangkaian proses untuk merepresentasikan pengetahuan kemudian melakukan komputasi agar dapat dipahami oleh mesin, dalam hal ini komputer. Menurut Yun et al. (2021) model pengetahuan adalah kumpulan pengetahuan dari berbagai jenis fakta, konsep, prinsip yang disusun secara terstruktur dan saling terhubung. Pengetahuan yang telah direpresentasi menjadi model pengetahuan selanjutnya dapat diinterpretasikan dan disimpan sebagai basis pengetahuan (*knowledge base*) dalam suatu manajemen pengetahuan sehingga pengetahuan dapat dikelola dan dimanfaatkan dengan lebih baik (Abubakar et al., 2019).

Untuk menyelesaikan masalah tersebut, ontologi dapat digunakan untuk memodelkan pengetahuan. Ontologi jamak digunakan sebagai cara untuk merepresentasikan pengetahuan dari sekumpulan konsep dan pengetahuan umum dalam sebuah domain pengetahuan dan hubungannya secara semantik sehingga ontologi dapat digunakan untuk penyajian informasi serta melakukan pengorganisasian dan pemetaan suatu konsep secara sistematis dan terstruktur (Fudholi et al., 2016; Fung & Bodenreider, 2019). Pemodelan pengetahuan menggunakan ontologi dapat diterapkan guna memodelkan proses yang terjadi antar setiap konsep (*concept*) yang terlibat dalam suatu pengetahuan serta membuat hubungan (*relationship*) antar konsep. Representasi pengetahuan pada ontologi menggunakan deskripsi berbasis bahasa OWL berbasis RDF (Da Silva, 2015).

Pada penelitian ini, dikembangkan sebuah model ontologi untuk merepresentasikan pengetahuan mengenai *big data analytics* di perguruan tinggi pada sisi pedagogis berdasarkan tiga aspek yaitu: 1) *Analytics*, 2) *Data sources*, dan 3) *Platforms*. Ontologi

dikembangkan dari awal (*from scratch*) dengan mengadaptasi metode *Noy & McGuiness Method* (Noy & McGuiness, 2001). Proses pemodelan ontologi pada penelitian ini dibagi menjadi 3 (tiga) fase, yaitu: (1) Konseptualisasi; (2) Implementasi; kemudian (3) Evaluasi. Panduan umum yang digunakan sebagai sumber pengetahuan berasal dari panduan DAMA-DMBOK (*Data Management Body of Knowledge*) dan ensiklopedia dari Encyclopedia of Big Data Technologies yang diterbitkan oleh Springer. Beberapa studi yang memuat *literature review/survey* mengenai *big data* terutama dalam konteks perguruan tinggi juga dijadikan sebagai sumber pengetahuan. Pada penelitian ini model ontologi dibangun menggunakan perangkat lunak Protégé dan dikodekan dalam format OWL (*Web Ontology Language*).

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan, maka rumusan masalah yang dibahas dalam penelitian ini adalah "Bagaimana penerapan teknik ontologi untuk membangun model yang merepresentasikan pengetahuan dari domain *big data analytics* di perguruan tinggi pada sisi pedagogis?".

## 1.3 Batasan Masalah

Berdasarkan rumusan masalah yang telah disebutkan, agar permasalahan dalam penelitian ini tidak meluas, maka perlu dibatasi pada:

1. Sumber pengetahuan berasal dari buku panduan umum serta studi *review/survey* pilihan mengenai *big data analytics* di perguruan tinggi yang membahas aspek pedagogis berdasarkan kebaruan dan jurnal tempat publikasi.
2. Model pengetahuan pada penelitian ini dibangun berdasarkan tiga aspek yaitu 1) *Analytics*, 2) *Data sources*, dan 3) *Platforms*.
3. Hasil penelitian ini hanya berupa model ontologi yang dikodekan dalam bahasa OWL/RDF dalam format XML.

## 1.4 Tujuan Penelitian

Berdasarkan rumusan masalah, maka tujuan penelitian ini adalah:

1. Mampu membangun model yang merepresentasikan pengetahuan pada domain *big data analytics* di perguruan tinggi dari sisi pedagogis.

2. Mampu menerapkan teknik ontologi untuk membangun model yang merepresentasikan pengetahuan tersebut.

## 1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan kontribusi pengetahuan, antara lain:

1. Model pengetahuan yang direpresentasikan diharapkan dapat menambah referensi bagi perguruan tinggi dalam memahami analisis *big data* pada sisi pedagogis dari aspek metode, sumber data, dan teknologi yang digunakan.
2. Model pengetahuan yang direpresentasikan menggunakan teknik ontologi diharapkan dapat digunakan untuk interoperabilitas pengetahuan pada penelitian selanjutnya.

## 1.6 Sistematika Penulisan

Agar mempermudah proses pembahasan dalam penelitian yang dibuat, maka dibuat sistematika penulisan pada penelitian ini:

**BAB 1 Pendahuluan**

Pendahuluan, merupakan pengantar terhadap permasalahan yang akan dibahas pada penelitian. Pada pendahuluan ini menguraikan tentang gambaran suatu penelitian yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian metodologi penelitian, dan sistematika penulisan.

**BAB 2 Landasan Teori**

Bab 2 menjelaskan teori-teori yang digunakan untuk mendukung dalam mengembangkan pemecahan masalah pada penelitian yang mengembangkan Model Pengetahuan Berbasis Ontologi pada Domain Big Data di Perguruan Tinggi. Teori-teori yang berkaitan di antaranya penelitian terdahulu mengenai pengembangan kerangka kerja/model untuk memahami pemanfaatan *big data* di institusi pendidikan, teori pemodelan pengetahuan, rekayasa ontologi, dan kajian mengenai *big data* di bidang pendidikan khususnya perguruan tinggi.

**BAB 3 Metodologi Penelitian**

Bab 3 membahas tentang langkah-langkah penelitian dan gambaran umum langkah penyelesaian yang dibuat secara sistematis dan dapat dijadikan pedoman yang jelas dalam menyelesaikan permasalahan.

**BAB 4 Hasil dan Pembahasan**

Bab 4 berisi uraian hasil dan pembahasan penelitian, yaitu 1) Konseptualisasi model pengetahuan, 2) Implementasi model pengetahuan ke dalam bentuk ontologi menggunakan bahasa OWL (*Web Ontology Language*), kemudian 3) Evaluasi berdasarkan analisis pengukuran kualitas ontologi dan analisis dari uji coba *query* menggunakan SPARQL (*SPARQL Protocol and RDF Query Language*).

**BAB 5 Kesimpulan dan Saran**

Bab 5 berisi tentang kesimpulan dari seluruh bab yang telah dibahas menjawab rumusan masalah dan yang perlu diperhatikan yaitu berdasar keterbatasan yang ditemukan dari penelitian dan asumsi-asumsi yang dibuat selama melakukan penelitian dan juga rekomendasi yang diuraikan untuk pengembangan penelitian selanjutnya.

# BAB 2
# Landasan Teori

## 2.1 Pemodelan Pengetahuan

Penelitian mengenai pemodelan pengetahuan telah berkembang sejak dekade 1990-an. Studi yang dilakukan oleh (Devedzic, 1999) mulai memperkenalkan proyek, teknologi, sistem, dan penerapan dari pemodelan pengetahuan. Pada penelitian selanjutnya (Devedzic, 2001) memperkenalkan konsep dasar, metodologi, dan teknik dalam memodelkan suatu pengetahuan. Model tersebut digunakan untuk merepresentasikan pengetahuan yang akan dimuat dalam sistem berbasis pengetahuan. Pemodelan pengetahuan adalah serangkaian proses untuk mengonseptualisasikan model pengetahuan kemudian melakukan komputasi agar dapat dipahami oleh mesin, dalam hal ini komputer. Menurut (Yun et al., 2021), model pengetahuan adalah kumpulan pengetahuan dari berbagai jenis fakta, konsep, prinsip yang disusun secara terstruktur dan saling terhubung. Pada ilmu komputasi, model pengetahuan adalah representasi suatu pengetahuan yang dapat dipahami oleh komputer yang memungkinkan pengetahuan diinterpretasikan oleh perangkat lunak dan disimpan dalam basis data atau *file* pertukaran data sehingga memungkinkan komputer dapat memahami pengetahuan tersebut.

Pada dasarnya, pengetahuan adalah gabungan dari data dan informasi yang dihubungkan antara satu dengan lainnya. Data merupakan kumpulan dari fakta-fakta, pengukuran, dan statistik. Informasi adalah data yang telah diorganisir dan terproses dengan akurat. Data dan informasi kemudian diproses menjadi pengetahuan yang bernilai relevan secara kontekstual dan dapat dikerjakan (Zagzebski, 2017). Agar pengetahuan dapat diterjemahkan oleh komputer perlu dilakukan akuisisi pengetahuan dengan cara akumulasi, transfer dan transformasi pengetahuan tersebut ke dalam bahasa yang dimengerti oleh komputer. Dalam mengembangkan model pengetahuan, pendekatan yang dapat ditempuh yaitu menggunakan teknik berbasis ontologi / *ontology engineering* (He et al., 2015).

## 2.2 Rekayasa Ontologi

### 2.2.1 Pengertian Ontologi

Istilah ontologi (o*ntology*) pada dasarnya merupakan istilah yang berasal dari bahasa Yunani, yaitu *ontos* dan *logos*. Definisi ontologi dalam ilmu komputer yang sering dirujuk berasal dari (Gruber, 1993) menyatakan bahwa "*An ontology is an explicit specification of*

*a conceptualization*" yang artinya adalah sebuah ontologi adalah sebuah spesifikasi yang eksplisit dari suatu konseptualisasi. Dalam ilmu komputasi, pemodelan ontologi adalah cara untuk merepresentasikan suatu domain pengetahuan secara eksplisit mengenai suatu konsep dengan cara memberikan makna, properti, serta relasi pada konsep tersebut sehingga terhimpun dalam suatu domain pengetahuan dan membentuk sebuah model pengetahuan (El-Diraby, 2013).

Agar dapat dikomputasikan ontologi menggunakan teknologi dari web semantik untuk merepresentasikan pengetahuannya. Organisasi World Wide Web Consortium (W3C) mengeluarkan rekomendasi bahasa yang digunakan untuk komputasi ontologi. Bahasa tersebut adalah OWL (*Web Ontology Language*) yang berbasis RDF (*Resource Description Framework*) yang ditulis dalam format XML (*Extensible Markup Languange*) sebagai dasar sintaks dalam melakukan pengkodean. OWL digunakan untuk memberi pernyataan yang lebih ekspresif dari ontologi, ekspresi tersebut ditulis dalam bentuk subjek-predikat-objek menggunakan RDF (Schneider et al., 2009).

Berdasarkan kajian dari (Davies et al., 2006), disebutkan bahwa ontologi mempunyai beberapa manfaat, di antaranya yaitu:

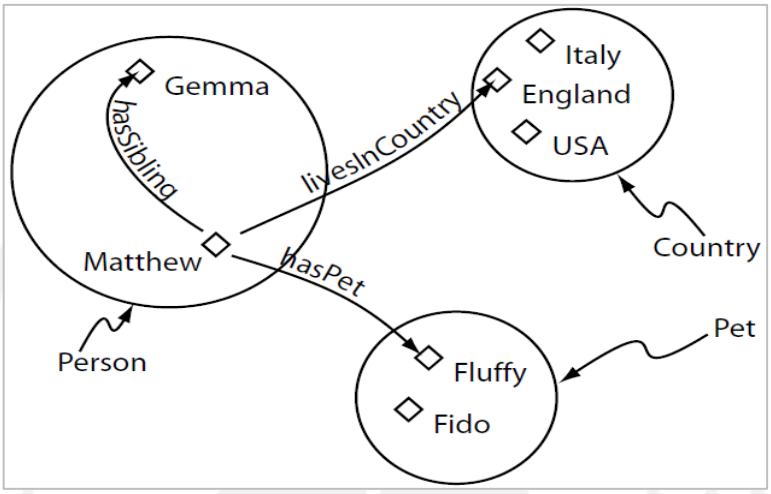
1. Domain pengetahuan dapat dijelaskan secara lebih eksplisit dengan cara menstrukturkan hierarki dari suatu konsep dari sebuah domain dan hubungan di antara mereka.
2. Mampu membagikan pemahaman dari pengetahuan yang terstruktur. Sebagai contoh sistem yang memuat data mengenai mahasiswa. Jika sistem tersebut dipakai bersama dan dipublikasikan dengan basis ontologi yang sama maka sistem dapat menciptakan dan mengumpulkan informasi dari situs yang berbeda.
3. Memungkinkan domain pengetahuan digunakan ulang, melalui cara mengembangkan ontologi yang telah ada sebelumnya dan mengintegrasikan dengan beberapa ontologi lainnya yang relevan dengan ontologi yang ingin dikembangkan.

### 2.2.2 Elemen Ontologi

Elemen pada ontologi terdiri kelas (*classes*), properti (*properties*) dan objek (*objects*). Kelas (*classes*) menjelaskan konsep dari pengetahuan dalam suatu domain. Sebuah kelas, memiliki sub-kelas (*subclasses*) yang ditujukan untuk menyatakan konsep yang lebih spesifik dari kelas utama (*superclass*). Properti (*properties*) merupakan relasi dua arah yang menghubungkan kelas yang satu dengan kelas lainnya sehingga membentuk tautan pengetahuan/konsep.

Ada dua jenis properti yang dapat digunakan dalam ontologi, yaitu properti objek (*object property*) yang digunakan untuk menghubungkan objek dengan objek lainnya, serta

properti data (*datatype property*) yang digunakan untuk menghubungkan objek dengan *datatype value* seperti *text*, *string* atau *number*. Objek (*objects*) atau disebut juga individu/anggota merupakan objek aktual dari suatu kelas. Objek memuat fakta maupun informasi dari suatu konsep/pengetahuan (Gangemi et al., 2006).




Gambar 2.1 Representasi Kelas-Properti-Objek

Contoh representasi suatu kelas-properti-objek dapat dilihat pada Gambar 2.1. Pada gambar tersebut, terlihat bahwa kelas `Person` mempunyai objek yaitu `Matthew` dan `Gemma`. Kedua objek tersebut saling terhubung melalui properti `hasSibling`. Objek `Matthew` terhubung pada objek `England` dari kelas `Country`. Sehingga pengetahuan yang dapat disimpulkan dari hubungan tersebut adalah bahwa Gemma mempunyai saudara bernama Matthew yang tinggal di Inggris. Selanjutnya, representasi pengetahuan tersebut perlu diformalkan ke dalam bahasa yang dimengerti oleh komputer, yaitu menggunakan bahasa OWL (*Web Ontology Language*) yang berbasis RDF (*Resource Description Framework*).

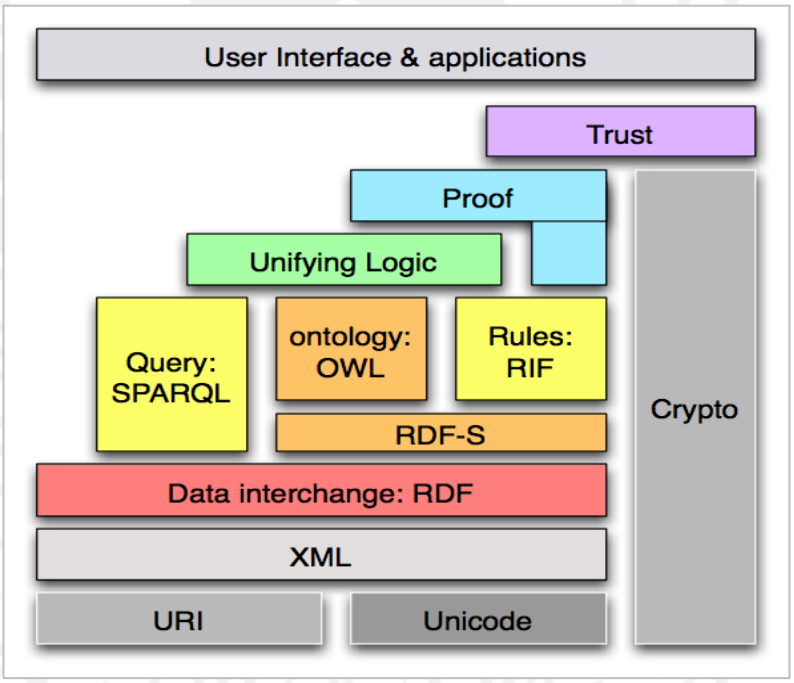
### 2.2.3 Formalisasi Ontologi

OWL (*Web Ontology Language*) adalah bahasa yang direkomendasikan oleh World Wide Web Consortium. OWL adalah bahasa ontologi paling ekspresif yang dapat digunakan untuk merepresentasikan arti dari berbagai konsep dalam suatu domain pengetahuan secara eksplisit serta relasi antara konsep tersebut yang tidak dapat dipisahkan dengan dokumen dan aplikasi web. OWL bersama dengan RDF merupakan bagian dari teknologi web semantik yang dikembangkan oleh W3C (Antoniou & Van Harmelen, 2004). Web semantik merupakan pengembangan dari teknologi web yang telah ada saat ini. Tujuan dari web semantik yaitu memperkaya informasi yang diberikan sehingga menjadi lebih baik dalam pendefinisiannya, agar memungkinkan komputer dapat memahami informasi yang telah diberikan sehingga komputer dan manusia dapat bekerja sama. Web semantik

memungkinkan suatu web menjadi lebih cerdas dikarenakan memiliki basis pengetahuan (*knowledge base*) didalamnya dalam bentuk ontologi (Berners-Lee *et al*., 2001).

Dalam teknologi web semantik, ontologi berperan sebagai teknologi utama sehingga disebut sebagai ontologi web semantik (*semantic web ontology*). Melalui teknologi tersebut, sumber daya yang ada pada web (*web resource*) tidak hanya dapat dipahami oleh manusia (*human-readable*) namun juga dapat dipahami oleh mesin/komputer (*machine-readable*) sehingga memungkinkan untuk melakukan pengelolaan kumpulan sumber daya web secara sistematis dan terstruktur. Dengan demikian, ontologi dari web semantik dapat memodelkan pengetahuan yang akan terus berkembang namun belum terstruktur (Badron et al., 2017).

Teknologi web semantik terbagi dalam beberapa *layer* (lapisan) yang terdapat pada arsitekturnya. Model arsitektur web semantik disebut sebagai *Semantic Web Stack*. Model arsitektur pada teknologi web semantik dapat dilihat pada Gambar 2.2.




Gambar 2.2 Semantic Web Stack (W3C, 2003)

Terlihat bahwa OWL yang merupakan inti dari teknologi web semantik. Representasi makna dari kosakata yang termuat dalam ontologi dan relasi antar kata sehingga makna suatu pengetahuan menjadi eksplisit. OWL dibangun berdasarkan RDF & RDFS dengan cara menambahkan kosakata yang lebih detail untuk mendefinisikan kelas, properti dan objek dalam ontologi.

OWL menyediakan tiga sub-bahasa yang penggunaannya disesuaikan dengan kebutuhan, yaitu OWL Lite, OWL DL, dan OWL Full. Pengembangan ontologi yang

menggunakan OWL dapat menentukan sub-bahasa mana yang tepat sesuai kebutuhan. Pilihan antara OWL Lite dan OWL DL bergantung pada cukup tidaknya konstruksi dengan OWL Lite, yaitu apakah cukup dengan ontologi sederhana menggunakan OWL Lite, atau ontologi yang lebih ekspresif menggunakan OWL DL. Sedangkan pilihan antara OWL DL dan OWL Full bergantung pada apakah lebih penting melakukan *automated reasoning* atau memberikan ekspresi yang lebih tinggi pada model seperti memberikan *metaclasses* (McGuinness, 2004).

### 2.2.4 SPARQL (*SPARQL Protocol and RDF Query Language*)

SPARQL adalah standar yang dikeluarkan oleh W3C guna melakukan *query* untuk memperoleh data dari sumber daya web (*web resource*) yang terdapat pada dokumen RDF dan OWL. SPARQL *query* terdiri atas *triple pattern* yang sama seperti RDF *triple* yaitu subjek-predikat-objek di mana masing-masing dari subjek-predikat-objek dapat menjadi variabel pada SPARQL. *Query* SPARQL didasarkan pada pencocokan pola *triple* pada RDF (Kollia & Glimm, 2013).

Klausa yang digunakan dalam *query* SPARQL, di antaranya:

1. PREFIX

Klausa `PREFIX` merupakan cara deklarasi yang digunakan untuk mewakili URI (*Uniform Resource Identifier*) sebagai penunjuk yang membawa informasi dalam suatu *resource* dalam ontologi. PREFIX digunakan untuk menyingkat sebuah URI.

2. SELECT

Klausa `SELECT` digunakan untuk mendefinisikan variabel yang akan dikembalikan sebagai hasil dari eksekusi *query*. Setiap variabel diawali dengan notasi "`?`".

3. WHERE

Klausa `WHERE` digunakan untuk mendefinisikan pola *triple* yang harus dimiliki oleh setiap hasil *query* yang valid. Seluruh pola yang merepresentasikan suatu kalimat RDF harus sesuai dengan RDF *triples*, yaitu terdiri dari subjek-predikat-objek. RDF *triple* tersebut dapat direpresentasikan oleh URI atau sebuah variable dan nilai literal.

### 2.2.5 Ontology Development 101

Metode *Ontology Development 101* merupakan metode pembangunan model ontologi yang dikembangkan oleh Natalya F. Noy dan Deborah L. McGuiness, peneliti dari Universitas Stanford, Amerika Serikat (Noy & McGuiness, 2001). Terdapat beberapa tahapan berdasarkan metode tersebut, namun tahapan tersebut tidak mengikat dan dapat dikerjakan secara fleksibel.

1. Menentukan domain dan cakupan ontologi

Dimulai dengan mendefinisikan domain dan cakupan dari ontologi tersebut. Hal ini dapat dilakukan agar dapat memahami domain dan cakupan apa yang melingkupi ontologi yang akan dibangun, untuk apa ontologi digunakan, dan siapa yang akan menggunakan dan mengurus ontologi.

2. Mempertimbangkan ontologi yang sudah ada

Pada tahap ini, dilakukan pengecekan apakah ada penelitian atau ontologi lain yang telah mengembangkan dengan domain yang sama. Jika telah ada, maka ontologi tersebut dapat menjadi rujukan, langsung digunakan ataupun diperbaiki. Tetapi jika belum ada maka akan diciptakan ontologi dengan domain yang baru.

3. Enumerasi istilah penting dalam ontology

Enumerasi istilah-istilah (terms) penting adalah proses menuliskan semua istilah yang diperlukan dalam pembentukan ontologi. Jenis kata dalam istilah yang diperlukan adalah kata benda untuk membentuk nama kelas dan kata kerja untuk membentuk nama properti.

4. Mendefinisikan kelas dan hierarki dari kelas

Mendefinisikan kelas dan menyusunnya dalam sebuah hierarki *superclass-subclass*. Pendekatan *top-down* yang dimulai dengan mendefinisikan konsep umum dalam domain dilanjutkan dengan mendefinisikan konsep yang lebih spesifik. Pendekatan *bottom-up* yang merupakan kebalikan dari pendekatan *top-down*. Serta pendekatan *combination* yang merupakan kombinasi antara pendekatan *top-down* dengan pendekatan *bottom-up*.

5. Mendefinisikan properti dari kelas

Tahapan setelah mendefinisikan class yaitu mendefinisikan properti dari kelas. Pada object property, hal yang didefinisikan adalah hubungan antara kelas yang satu dengan yang lain (*binary relation*) beserta *inverse* dari properti tersebut. *Inverse property* digunakan untuk mendefinisikan properti yang berlawanan. Sedangkan pada *datatype property*, hal yang didefinisikan adalah hubungan antara class dengan tipe data beserta beserta kardinalitas dari *datatype property* tersebut.

6. Mendefinisikan batasan dari properti

Mendefinisikan batasan (*facets*) tertentu dari properti setiap kelas yang memiliki tipe nilai khusus. Terdapat dua kategori batasan, yaitu *cardinality*, untuk mendefinisikan banyaknya nilai yang dimiliki setiap properti; dan *type* untuk menegaskan tipe data pada properti yang didefinisikan.

7. Membuat objek

Pada tahap ini membuat pada masing-masing kelas yang terkait dengan domain pengetahuan. Dalam mendefinisikan objek perlu diperhatikan hal-hal yaitu pemilihan kelas, pembuatan objek pada kelas tersebut serta mengisi nilai propertinya.

## 2.3 Big Data di Bidang Pendidikan

### 2.3.1 Karakteristik Big Data

*Big data* atau mahadata adalah istilah yang menunjukkan fenomena dari besarnya volume data, cepat dan kompleks sehingga memerlukan teknik tertentu untuk menyimpan, mentransmisi dan menganalisisnya, yang dikenal dengan *Big Data Analytics* (Tsai et al., 2015). Menurut penelitian Anuradha (2015), secara umum big data mempunyai tiga karakteristik utama yang kemudian dilengkapi oleh dua karakteristik lainnya. Selanjutnya karakteristik tersebut disingkat dengan 5V, yang terdiri dari *Volume, Variety, Velocity*, *Veracity, and Value*. Model 5V's Big Data dapat dilihat pada Gambar 2.3.




Gambar 2.3 Bagan 5V's Big Data (Anuradha, 2015)

Karakteristik 5V's Big Data (*Volume, Variety, Velocity*, *Veracity, and Value*.) dapat dijabarkan sebagai berikut:

1. Volume

Volume menunjukkan sejumlah besar data yang seringkali sulit untuk disimpan, diproses, ditransfer, dianalisis, dan disajikan sehingga memerlukan proses penyimpanan dan pengolahan secara khusus. Aspek ini adalah yang sering dipahami oleh kebanyakan orang sebagai *big data*. Namun, dalam bidang teknologi informasi, volume belum cukup untuk mengatakan suatu kumpulan data disebut sebagai *big data*. Terdapat aspek-aspek lainnya sehingga suatu kumpulan data disebut sebagai *big data*.

2. Variety

Aspek ini menunjukkan bahwa suatu kumpulan data mempunyai format yang beragam baik sebagai data yang terstruktur maupun data yang tidak terstruktur. Data yang terstruktur merupakan data yang tersimpan dalam format tetap yang dapat diproses, disimpan, dan diambil kapan saja. Data terstruktur yang lazim diketahui seperti data dalam excel atau spreadsheet maupun basis data relasional. Pada kasus big data, mayoritas data merupakan data tidak terstruktur sehingga perlu teknik khusus dalam mengolahnya karena data tersebut berbentuk format yang tidak tetap, seperti gambar maupun suara.

3. Velocity

Aspek *velocity* menunjukkan bahwa suatu kumpulan data yang diciptakan dan disimpan memiliki kecepatan aliran data (*data streaming*) yang berlangsung cukup tinggi. Sebagai contoh perusahaan *e-commerce* seperti Amazon menangkap setiap klik/tap saat pembeli menjelajah di situs web maupun aplikasinya yang terjadi dengan cepat. Sehingga, dalam memproses data yang sedemikian cepat perlu teknologi khusus untuk menanganinya.

4. Veracity

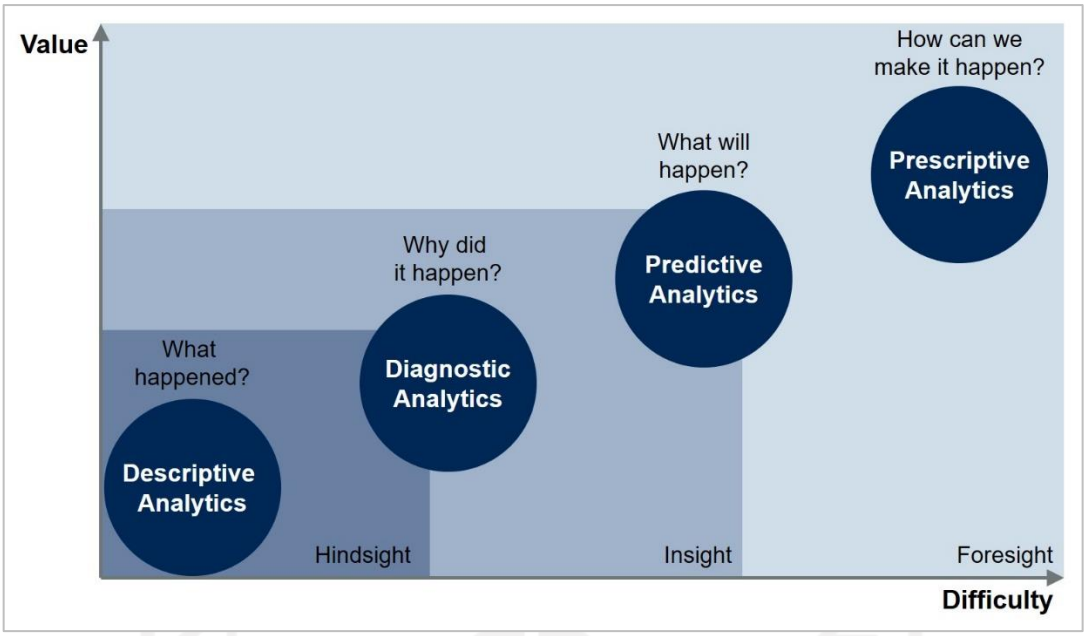
Aspek *veracity* merupakan pelengkap dari tiga karakteristik sebelumnya. Aspek ini mengacu pada kepercayaan terhadap data tersebut karena adanya kemungkinan terjadinya bias atau abnormalitas dalam data yang dihasilkan oleh organisasi. Sebagai konsekuensi dari besar dan beragamnya data, maka perlu kepastian bahwa kumpulan data tersebut dapat dipercaya.

5. Value

Di samping aspek *veracitiy*, aspek *value* juga dinilai penting bagi *big data* karena dari sini kumpulan data tersebut dapat dikatakan berguna sehingga mampu memberikan wawasan yang bermanfaat pada organisasi. Pada konteks *big data analytics*, *value* dari *big data* terletak pada hasil analisis dan prediksi atau tindakan yang dapat diambil dari hasil analisis dan prediksi tersebut

### *2.3.2 Big Data Analytics*

Penelitian maupun pemanfaatan *big data analytics* bagi suatu organisasi terus berkembang hingga saat ini. *Analytics* pada big data umumnya dapat dibagi menjadi empat jenis, yaitu *descriptive analytics, diagnostic analytics, predictive analytics,* dan *prescriptive analytics* (Shi-Nash, 2017; Deshpande, 2019). Masing-masing jenis *analytics* mempunyai tujuan dan tingkat kesulitannya sendiri seperti yang ditunjukkan pada Gambar 2.4.




Gambar 2.4 Empat Jenis Analytics (Gartner, 2019)

Agar lebih jelas memahami maksud dari setiap jenis *analytics*, berikut ini penjabaran dari tujuan dari setiap jenis *analytics*:

1. *Descriptive analytics*

*Descriptive analytics* merupakan pemeriksaan data atau konten yang dapat dilakukan secara manual, untuk menjawab pertanyaan "apa yang terjadi?" (*what happened*?). Pertanyaan tersebut dijawab dengan cara melakukan interpretasi data historis secara holistik sehingga organisasi dapat mengidentifikasi fenomena yang terjadi.

2. *Diagnostic analytics*

*Diagnostic analytics* merupakan bentuk *analytics* lanjutan yang bertujuan untuk menjawab pertanyaan "mengapa itu terjadi?" (*why did it happen?*). Hal ini dilakukan dengan cara menggali lebih dalam data historis untuk menemukan wawasan yang berharga. Sehingga, dapat diambil langkah lebih jauh untuk mengungkap alasan di balik terjadinya suatu fenomena di organisasi.

3. *Predictive analytics*

*Predictive analytics* merupakan bentuk *analytics* lebih lanjut yang bertujuan untuk menjawab pertanyaan "apakah itu akan terjadi?" (*what will happen?*) dengan cara menggunakan teknik seperti *data mining* atau *machine learning* untuk melakukan klasifikasi dan pengelompokkan data. Dengan demikian diketahui pola dari data historis tersebut sehingga dapat digunakan untuk memprediksi hal yang akan untuk kemudian diantisipasi.

4. *Prescriptive analytics*

*Prescriptive analytics* merupakan bentuk *analytics* yang paling sulit sekaligus memberikan nilai (*value*) yang lebih baik dari suatu kumpulan data. *Prescriptive analytics* menjawab pertanyaan tentang "bagaimana kita bisa mewujudkannya?" (*how can we make it happen?*). Pertanyaan tersebut dapat dijawab dengan cara memanfaatkan teknik *machine learning* sehingga dapat merekomendasikan tindakan apa yang harus diambil untuk menghilangkan masalah di masa depan atau mengambil keuntungan penuh dari tren yang akan terjadi.

### 2.3.3 *Big Data* Pendidikan

*Big data* pendidikan (*educational big data*) menjadi aset yang penting sekaligus strategis dalam memajukan institusi pendidikan. Secara umum *big data* menggambarkan sejumlah data yang tersimpan dalam basis data dan repositori institusi, berdasarkan karakteristik *big data* dari himpunan data tersebut. Pada konteks perguruan tinggi, *big data analytics* dapat digunakan untuk melakukan analisis pembelajaran. Analisis pembelajaran berkaitan dengan pengukuran, pengumpulan dan analisis dan pelaporan data tentang peserta didik untuk memahami dan mengoptimalkan pembelajaran dan lingkungan di mana pembelajaran itu terjadi. Hal tersebut dilakukan untuk mendukung peningkatan keberhasilan mahasiswa (Daniel, 2017; Bai et al., 2021).

Istilah *big data* pendidikan berasal dari perkembangan data pendidikan yang berkembang pesat, termasuk data yang dihasilkan oleh pelaku pendidikan, perilaku belajar, maupun keadaan psikologisnya. Salah satu fitur terpenting dari *big data* pendidikan adalah nilai (*values*) yang diperoleh dari kumpulan data tersebut. *Big data* pendidikan mempunyai banyak cara pemanfaatan yang dapat digunakan untuk administrasi pendidikan maupun inovasi pengajaran, seperti memprediksi kinerja akademik mahasiswa berdasarkan data historis dari perkuliahan yang pernah diikuti. Selain dapat digunakan untuk memprediksi, *big data* pendidikan juga dapat digunakan untuk memberikan rekomendasi maupun evaluasi perkuliahan mahasiswa (Cen et al., 2015; Quadir et al., 2020).

Data pendidikan dihasilkan dalam jumlah besar dari banyak institusi pendidikan di dunia yang dalam hal ini mewakili volume *big data*. Selain itu kumpulan data tersebut terus bertumbuh sehingga menciptakan kecepatan arus data yang terus menerus. *Big data* pendidikan berisi banyak entitas (mahasiswa, pendidik, dan pengelola institusi pendidikan) dan hubungan (hubungan pendidik-mahasiswa, hubungan teman satu angkatan, dan hubungan perteman lain), yang membuat sistem *big data* pendidikan lebih menantang seperti duplikasi data dan memproses data dalam jumlah besar (Wang, 2016).

## 2.4 Penelitian Terdahulu

Penelitian mengenai *Big Data* memberikan kontribusi pengetahuan signifikan dalam dunia pendidikan. Pendekatan yang dapat dilakukan untuk memahami manfaat dari penerapan *Big Data* yaitu memodelkan konsep tersebut. Terdapat beberapa penelitian terdahulu yang membahas tentang pemodelan *Big Data* di perguruan tinggi. Penelitian tersebut dapat dilihat pada Tabel 2.1.

Tabel 2.1. Penelitian Big Data di Perguruan Tinggi

| **Penulis** | **Judul** | **Aspek** | **Penelitian** |
|---|---|---|---|
| Rozas et al. (2019) | Readiness of Indonesian Higher Education Programs in the Big Data Era | Adopsi *big data* | Mengembangkan model kerangka kerja untuk menganalisis kesiapan perguruan tinggi dalam menghadapi era *big data*. |
| Attaran et al. (2019) | Opportunities and Challenges for Big Data Analytics in US Higher Education: A Conceptual Model for Implementation | Analisis peluang dan tantangan *big data analytics* | Mengembangkan model konseptual untuk keberhasilan implementasi *big data analytics* pada perguruan tinggi. |
| Seufert et al. (2019) | A Pedagogical Perspective on Big Data and Learning Analytics: a Conceptual Model for Digital Learning Support | Perspektif pedagogis *big data analytics* | Mengembangkan model konseptual yang mencakup dimensi kritis dari *learning analytics* untuk mendukung praktik pembelajaran berbasis digital. |
| Ashaari et al. (2020) | Big Data Analytics Technology Capability in Malaysian Higher Education Institutions: A Conceptual Framework | Kapabilitas teknologi *big data* | Mengembangkan model kerangka kerja untuk memahami kapabilitas teknologi yang mendukung penerapan *big data analytics* di perguruan tinggi. |
| Alsheikh (2020) | Developing an Integrated Framework to Utilize Big Data for Higher Education Institutions in Saudi Arabia | Pemanfaatan *big data* | Mengembangkan model kerangka kerja terintegrasi untuk memanfaatkan *big data* di perguruan tinggi. |
| Fahmi (2021) **Penelitian yang dilakukan** | Ontology-based Model on Big Data Analytics in Higher Education | Pengetahuan *big data analytics* | Mengembangkan model ontologi untuk merepresentasikan pengetahuan *big data analytics* di perguruan tinggi berdasarkan konsep |

| Penulis | Judul | Aspek | Penelitian |
|---|---|---|---|
| | | | *analytics*, *data source*, dan *platforms*. |

Memodelkan konsep/pengetahuan dapat digunakan sebagai cara untuk memahami lebih komprehensif mengenai konsep/pengetahuan tersebut. Penelitian yang mengembangkan model untuk memahami cara kerja maupun penerapan *big data* di perguruan tinggi telah banyak dilakukan. Penelitian yang dilakukan oleh Rozas et al. (2019) mengembangkan model kerangka kerja untuk menganalisis kesiapan perguruan tinggi dalam menghadapi era *big data*. Penelitian dilakukan agar perguruan tinggi mampu menjawab tantangan yang dihadapi dalam mempersiapkan ahli yang mampu mengolah data di era *big data*, termasuk di bidang informatika, statistika, dan aktuaria. Data diambil dari basis data kementerian terkait yang biasa disebut dengan Forlap Dikti.

Penelitian sejenis dilakukan oleh Attaran et al. (2019), yang mengembangkan model konseptual untuk keberhasilan implementasi *big data analytics* pada perguruan tinggi di AS. Model konseptual dikembangkan guna memahami manfaat potensial dari penerapan *big data analytics* di perguruan tinggi dan mengeksplorasi tantangan yang dapat diantisipasi. Penelitian ini mengulas aspek kritikal dengan menggambarkan beberapa cara yang dapat diambil dalam penerapannya, serta mengulas keberhasilan implementasi solusi *big data analytics* di beberapa universitas.

Jumlah peserta didik yang menggunakan *platform* pembelajaran digital yang terus meningkat memunculkan fenomena aliran data dalam jumlah besar. Teknologi *big data* dapat digunakan untuk memanfaatkan data tersebut dalam rangka menganalisis pembelajaran (*learning analytics*). Namun dalam hal ini perspektif pedagogis belum dicakup. Penelitian yang dilakukan oleh Seufert et al. (2019) mengembangkan model konseptual yang mencakup perspektif pedagogis dari *learning analytics*. Peneliti memetakan tujuan serta penerapan *learning analytics* dari perspektif pedagogis. Dengan demikian, kerangka kerja dapat digunakan untuk mudah memahami dan memprediksi pembelajaran dan kinerja mahasiswa dari perspektif pedagogis.

Dalam menerapkan *big data analytics* di perguruan tinggi, perlu diketahui kapasitas teknologi dari institusi tersebut. Penelitian yang dilakukan oleh Ashaari et al. (2020) mengembangkan model kerangka kerja untuk memahami kapabilitas teknologi yang mendukung penerapan *big data analytics* di perguruan tinggi. Berdasarkan kerangka kerja untuk memahami kapabilitas teknologi yang dikembangkan dalam penelitian ini, terdapat tiga aspek yang dianggap signifikan, yaitu: konektivitas, kompabilitas, dan modularitas

yang secara langsung dapat mempengaruhi kapabilitas penerapan *big data analytics* di perguruan tinggi. Dengan demikian tiga aspek tersebut harus diperkuat agar penerapan sukses dilakukan.

Penelitan sejenis dilakukan oleh Alsheikh (2020) yang mengembangkan model kerangka kerja terintegrasi untuk memanfaatkan big data di perguruan tinggi. Penelitian ini mengumpulkan data melalui survei yang dilakukan sendiri sebagai kepala sekolah metode kuantitatif dan melalui wawancara mendalam semi terstruktur sebagai tindak lanjut metode kualitatif. Hasil dari penelitian ini adalah kerangka kerja terintegrasi yang menggambarkan faktor-faktor yang mempengaruhi adopsi dan penerapan *big data analytics* serta di institusi perguruan tinggi.

Berdasarkan berbagai penelitian yang telah dijabarkan, belum ada penelitian yang mengembangkan model untuk merepresentasikan pengetahuan. Penelitian ini mengusulkan pengembangan model menggunakan teknik ontologi untuk merepresentasikan pengetahuan terkait *big data analytics*. Penelitian ini diharapkan dapat memodelkan pengetahuan mengenai *big data analytics* di perguruan tinggi. Konseptualisasi pengetahuan dibagi berdasarkan tiga aspek utama, yaitu: 1) *analytics*, 2) *data source*, dan 3) *platforms*. Model pengetahuan dari ontologi divisualisasikan agar pengetahuan yang saling terkait tersebut dapat lebih dipahami.

# BAB 3
# Metodologi Penelitian

Pada penelitian ini, ontologi dikembangkan dari awal (*from scratch*) dengan mengadaptasi metode *Noy & McGuiness Method* yang dikembangkan oleh Noy & McGuiness (2001). Proses pengembangan model ontologi pada penelitian ini dibagi menjadi 3 (tiga) fase, yaitu: (1) Konseptualisasi; (2) Implementasi; kemudian (3) Evaluasi. Alur pengembangan model ontologi pada penelitian ini dapat dilihat pada Gambar 3.1.




Gambar 3.1 Alur Pengembangan Ontologi

## 3.1 Metodologi Konseptualisasi

Konseptualisasi yaitu menyusun dan memetakan pengetahuan yang akan digunakan sebagai dasar untuk kemudian diimplementasikan ke dalam model ontologi.

### 3.1.1 Pemetaan Literatur

Kegiatan menganalisis dan mengompilasi literatur guna memetakan literatur tersebut untuk menjadi sumber pengetahuan. Beberapa literatur digunakan sebagai panduan umum dalam pemodelan ontologi. Panduan tersebut dapat dilihat pada Tabel 3.1.

Tabel 3.1 Panduan Umum *Big Data*

| **Judul** | **Editor** | **Penerbit** | **Tahun** |
|---|---|---|---|
| Data Management Body of Knowledge – DMBOK 2nd Edition | Early & Coleman | DAMA International | 2017 |
| Encyclopedia of Big Data Technologies | Sakr & Zomaya | Springer | 2019 |

Beberapa studi berupa *review/survey* juga dipilih sebagai sumber pengetahuan berdasarkan topik *big data & analytics* terutama dalam konteks perguruan tinggi. Studi *review/survey* diambil karena mencakup banyak studi yang mengkaji topik tersebut. Makalah yang dikumpulkan dinilai dari cakupan, kebaruan, dan jumlah sitasi. Daftar makalah tersebut dapat dilihat pada Tabel 3.2.

Tabel 3.2 Studi *Big Data & Analytics*

| **Judul** | **Penulis** | **Jurnal** | **Tahun** |
|---|---|---|---|
| Big Educational Data & Analytics – Survey, Architecture and Challenges | Ang et al. | IEEE Access | 2020 |
| Big Data in Education: A State of the Art, Limitations, and Future Research Directions | Baig et al. | International Journal of Educational Technology | 2020 |
| Big Data Technologies- A Survey | Oussous et al. | Journal Computer and Information Sciences | 2019 |
| Educational Data Mining and Learning Analytics for 21st Century Higher Education: A review and synthesis | Alodwah et al. | Telematics and Informatics | 2019 |
| Big data in education: A bibliometric review | Marín et al. | Social Sciences | 2019 |

### 3.1.2 Penentuan Taksonomi

Setelah memetakan literatur, dilakukan pembentukan kelas kemudian tingkatan kelas dan sub-kelas kemudian disusun dalam suatu taksonomi. Taksonomi dibentuk dari konsep umum dilanjutkan pada konsep yang lebih spesifik.

### 3.1.3 Penentuan Properti

Setelah mengonsep taksonomi, dilakukan penentuan properti untuk menghubungkan kelas dengan objek yang ada dalam ontologi. Kegiatan ini berdasarkan dari berbagai sumber pengetahuan yang telah diperoleh sebelumnya.

## 3.2 Metodologi Implementasi

Pada tahapan ini, dilakukan pembentukan kelas, properti dan objek berdasarkan hasil konseptualisasi yang telah dibuat. Berdasarkan rancangan model yang telah dikonseptualisasi, dilakukan implementasi ke dalam bahasa OWL-DL (*Web Ontology Language - Description Logic*). Pada penelitian ini, pengembangan ontologi menggunakan perangkat lunak Protégé 5.5.0.

### 3.2.1 Pembentukan Kelas & Properti

Pada tahapan ini dilakukan pembentukan kelas berdasarkan hierarki dari kerangka konseptual yang sudah dibuat. Setelah membentuk kelas, dilakukan pembentukan properti

dari kelas tersebut. Properti dibentuk berdasarkan relasi yang sudah ditentukan dari kerangka konseptual. Terdapat dua jenis properti, yaitu properti objek yang nanti akan digunakan sebagai relasi kelas dengan objek, dan properti data yang memuat data dari objek.

### 3.2.2 Pembentukan Objek

Pada tahap ini dibentuk objek dari kelas yang terkait dengan domain pengetahuan. Objek adalah pengetahuan aktual yang dimuat oleh kelas dan saling terhubung melalui properti. Seluruh objek akan dibentuk berdasarkan kamus konsep yang telah dibuat dalam tahapan konseptualisasi sebelumnya.

## 3.3 Metodologi Evaluasi

### 3.3.1 Analisis Ontologi

Setelah melakukan pengembangan ontologi, dilakukan evaluasi untuk menganalisis ontologi menggunakan metode OntoQA (*Ontology Quality Analysis*). Model ontologi dianalisis berdasarkan *schema metrics* untuk mengukur *Relationship Richness* (RR), *Inheritance Richness* (IR), *dan Attribute Richness* (AR). Penjelasan dari analisis *schema metrics* menggunakan metode OntoQA dari Tartir & Arpinar (2007).

**1) *Relationship Richness* (RR)**

Relationship Richness menyatakan keberagaman relasi yang dimiliki. RR didefinisikan dengan persamaan sebagai berikut.

$$RR = \frac{|P|}{|H|+|P|} \quad (1)$$

Di mana :
RR : *Relationship Richness*
| P | : Jumlah relasi *non-inheritance*
| H | : Jumlah relasi *inheritance*

Jumlah relasi *non-inheritance* (P) dibagi dengan jumlah relasi yang ada (jumlah relasi *inheritance* (H) ditambahkan dengan jumlah relasi *non-inheritance* (P). Relasi *inheritance* didefinisikan juga sebagai relasi *class-subclass*, sedangkan relasi *non-inheritance* adalah relasi selain *class-subclass*. Jika nilai RR yang didapatkan mendekati angka 0, maka sebagian besar dari relasi adalah *inheritance*. Sebaliknya, jika nilai RR yang didapatkan semakin mendekati angka 1, maka semakin kaya relasi yang dimiliki oleh ontologi.

Acuan penilaian dapat dilihat pada Tabel 3.3.

Tabel 3.3 Skala Klasifikasi Nilai RR

| **Rentang Nilai** | **Keterangan** |
|---|---|

| | |
|---|---|
| 0,00 – 0,20 | Minimum |
| 0,21 – 0,40 | Kurang |
| 0,41 – 0,60 | Cukup |
| 0,61 – 0,80 | Kaya |
| 0,81 – 1,00 | Maksimum |

**2) *Inheritance Richness (IR)***

*Inheritance Richness* merupakan cara untuk mengukur distribusi informasi. Pada analisis pengukuran ini, dapat dibedakan ontologi yang mempunyai karakter *deep* atau *shallow ontology*. Ontologi yang memiliki banyak level *inheritance* merupakan ontologi yang bersifat *deep*. Artinya, ontologi tersebut memiliki representasi pengetahuan yang lebih mendalam atau spesifik terhadap sebuah *domain*. Hal ini dapat dilihat dengan nilai IR yang kecil. Sebaliknya, nilai IR yang besar menunjukkan bahwa representasi pengetahuan yang dimiliki kurang mendalam namun mencakup area yang lebih luas dari sebuah *domain* atau bersifat *shallow*. IR didefinisikan dengan persamaan sebagai berikut.

$$IR = \frac{|H(c)|}{|C|} \quad (2)$$

Di mana :
IR : *Inheritance Richness*
| H (c) | : Jumlah relasi *inheritance* yang mempunyai *subclass*
| C | : Jumlah *superclass*

Acuan penilaian dapat dilihat pada Tabel 3.4.

Tabel 3.4 Skala Klasifikasi Nilai IR

| **Rentang Nilai** | **Keterangan** |
|---|---|
| 0,00 – 1,34 | Spesifik |
| 1,35 – 2,68 | Cukup Spesifik |
| 2,69 – 4,02 | Cukup Umum |
| 4,03 – 5,36 | Umum |

**3) *Attribute Richness (AR)***

*Attribute Richness* merupakan cara untuk mengukur jumlah atribut yang ada. Semakin banyak *attribute* atau *slot* maka semakin kaya pengetahuan yang disediakan oleh ontologi. AR didefinisikan sebagai jumlah rata-rata *attribute* per *class.* AR didapatkan dengan membagi jumlah *attribute* untuk semua *class* (att) dengan jumlah *class* (C). Perhitungan AR dapat dilihat pada persamaan berikut.

$$AR = \frac{|att|}{|C|} \quad (3)$$

Di mana :
AR : *Attribute Richness*
| att | : Jumlah *attribute* untuk semua *class*
| C | : Jumlah *class*

### 3.3.2 Analisis *Query*

Pada model ontologi yang telah dibangun, dilakukan pengecekan konsistensi untuk menguji model tersebut. Pengecekan konsistensi menggunakan *reasoner* yang terdapat pada perangkat lunak Protégé. Jika model telah dinyatakan konsisten, maka dilakukan pengujian *query* pada model ontologi tersebut untuk menarik pengetahuan yang tersimpan dalam ontologi. Pada penelitian ini dilakukan *query* menggunakan SPARQL. Skenario pengujian *query* pada penelitian ini yaitu memberikan beberapa pertanyaan untuk dijawab. Secara umum, pertanyaan tersebut berupa sasaran pemanfaatan *big data* yang ingin dituju, sumber data yang digunakan, teknologi basis data yang untuk menyimpan data berdasarkan sumber data, alat yang dapat digunakan untuk menyimpan *big data*, kemudian teknik untuk memproses data tersebut.

Pertanyaan tersebut untuk menguji sejauh mana ontologi dapat menyediakan pengetahuan yang tersimpan di dalamnya. Pertanyaan-pertanyaan tersebut mengandung sebagian besar dari pengetahuan yang sudah dimodelkan oleh ontologi. Langkah-langkah *query* menggunakan SPARQL dijelaskan dalam penelitian ini. Jawaban dari pertanyaan tersebut berupa hasil *query* SPARQL yang dieksekusi melalui perangkat lunak Protégé. Hasil *query* tersebut kemudian divalidasi berdasarkan dengan kecocokan pengetahuan yang telah tersimpan dalam ontologi.

# BAB 4
# Hasil dan Pembahasan

## 4.1 Konseptualisasi

### 4.1.1 Pemetaan Literatur

Menurut DMBOK (*Data Management Body of Knowledge*), *big data* pada dasarnya tidak mempunyai makna sampai dimanfaatkan dengan baik, yaitu dengan cara melakukan analisis big data tersebut, atau yang lebih dikenal dengan sebutan *big data analytics*. Pada proses *big data analytics*, diperlukan sasaran/tujuan dari pemanfaatan data tersebut, data apa saja yang dilakukan untuk melakukan analisis, dan perangkat apa saja yang digunakan baik sebagai prakondisi sebelum *big data* dianalisis hingga ketika *big data* dianalisis. Berdasarkan DMBOK, pemodelan ontologi pada penelitian ini dibagi menjadi 3 (tiga) konsep, yaitu: (1) *Analytics*, sebagai representasi tentang konsep menganalisis data; (2) *Data Source*, sebagai representasi tentang konsep sumber data untuk analisis; serta (3) *Platforms*, sebagai representasi tentang konsep perangkat yang digunakan untuk mempersiapkan dan menganalisis *big data.*

Teknologi yang umum digunakan dalam pemrosesan *big data* adalah Apache Hadoop. Hadoop menyediakan *platform* untuk memproses dan menganalisis data dengan lebih unggul dibanding teknologi tradisional lainnya. Hadoop memungkinkan pemrosesan data dalam jumlah besar (*big data*) secara terdistribusi dalam berbagai *server cluster*. Apache Hadoop merupakan kerangka kerja (*framework*) yang berisi kumpulan perangkat lunak *open source* yang dapat digunakan untuk memecahkan masalah komputasi data khususnya *big data* melalui dua komponen utamanya yaitu: (1) MapReduce; model pemrograman yang memungkinkan paralelisasi dan distribusi dan (2) HDFS; *filesystem* yang terdistribusi dengan ribuan *node* yang memungkinkan transfer data secara cepat di antara *node*.

#### 4.1.1.1 Konsep "Analytics"

Dilakukan identifikasi istilah yang berkaitan dengan konsep *analytics* untuk dijadikan sebagai kelas dalam ontologi. Pada konsep analytics, terdapat empat kelas yang dapat digunakan sebagai bagian dari ontologi, yaitu Behavior, Learning, Predictive, dan Methods. Kelas Behavior digunakan untuk memuat pengetahuan mengenai analisis perilaku belajar mahasiswa. Kelas Learning untuk memuat pengetahuan mengenai bagaimana memahami gaya belajar mahasiswa, bagaimana berkolaborasi, dan penilaian

belajarnya. Kelas Predictive dijadikan sebagai kelas untuk memuat pengetahuan mengenai apa saja yang bisa dilakukan untuk memprediksi performa belajar mahasiswa, pilihan mata kuliah yang tepat dan evaluasi materi serta apakah mempunyai potensi *dropout* atau tetap bertahan.

Menurut Aldowah et al. (2019) dan Ang et al. (2020), terdapat beberapa sasaran untuk menganalisis mahasiswa seperti perilaku (*behavior*), pembelajaran (*learning*), maupun prediksi mengenai mahasiswa tersebut (*predictive*). Secara umum, berikut ini sasaran yang dapat dituju untuk menganalisis mahasiswa.

1. Perilaku (*behavior*)
    a. Pemodelan tindakan; menganalisis dengan cara memodelkan tindakan yang biasa dilakukan oleh mahasiswa untuk memahami perilaku mahasiswa tersebut.
    b. Pemodelan pengetahuan mahasiswa; menganalisis pengetahuan mahasiswa terkait materi pembelajaran dengan cara memodelkannya.
    c. Pemodelan pola perilaku; memodelkan perilaku mahasiswa dengan menganalisis pola perilaku mahasiswa tersebut sehingga kinerja mahasiswa dapat dipetakan.
2. Pembelajaran (*learning*)
    a. Kemampuan kolaborasi; yaitu menganalisis kemampuan kolaborasi mahasiswa dalam pembelajaran sehingga dapat diketahui dalam bagian mana saja mahasiswa dapat bekerja sama satu dengan lainnya.
    b. Penilaian pembelajaran; yaitu mengevaluasi hasil pembelajaran untuk menyesuaikan dengan sasaran akademis yang ingin diraih.
    c. Personalisasi; menganalisis cara belajar mahasiswa untuk dilakukan penyesuaian dengan sistem pembelajaran yang diadakan institusi.
    d. Analisis sosial; menganalisis jejaring sosial untuk memahami hubungan antara mahasiswa dengan mahasiswa lainnya, tenaga pengajar, komunitas dan sumber daya lainnya.
3. Prediksi (*predictive*)
    a. Pemilihan mata kuliah; melakukan analisis untuk memprediksi mata kuliah yang cocok yang dapat diikuti oleh mahasiswa.
    b. Analisis ketahanan mahasiswa; untuk memprediksi sejauh mana mahasiswa dapat bertahan dalam suatu perkuliahan, berdasarkan faktor-faktor yang mempengaruhi akademik mereka.

c. Evaluasi materi kuliah; menganalisis materi perkuliahan yang ada untuk memprediksi bagaimana materi kuliah dapat dirancang agar sesuai dengan mahasiswa.
d. Kinerja mahasiswa; menilai dan menganalisis berbagai capaian akademik mahasiswa sehingga dapat memprediksi kinerja mahasiswa tersebut dalam mengikuti kegiatan akademik berikutnya.

Menurut Aldowah et al. (2019), Marin et al. (2019), dan Baig et al. (2020), terdapat beberapa metode yang dapat digunakan dalam melakukan analisis data mahasiswa. Berikut ini metode-metode yang dapat digunakan.

1. *Association rule*; teknik *data mining* untuk menemukan hubungan antara variabel dan kelompok atribut dari pola data tertentu. Metode ini dapat digunakan untuk menawarkan mata kuliah sesuai karakteristik dan kompetensi mahasiswa.
2. *Classificiation*; merupakan teknik yang digunakan untuk mengklasifikasikan data ke dalam kelas yang telah ditentukan. Konsep klasifikasi dapat digunakan untuk memprediksi kinerja mahasiswa, pengetahuan, memprediksi/mencegah siswa putus kuliah, mendeteksi perilaku mahasiswa dalam perkuliahan.
3. *Clustering*; mengidentifikasi/mengelompokkan data berdasarkan kesamaannya. *Clustering* dapat digunakan untuk mengelompokkan data interaksi antar mahasiswa, karakteristik atau cara belajarnya, maupun merekomendasikan mata kuliah.
4. *Correlation mining*; mengukur hubungan linier antara dua variabel. Dalam konteks pendidikan metode ini dapat digunakan untuk memprediksi kinerja mahasiswa, mengidentifikasi aturan penilaian pembelajaran, dan memodelkan pengetahuan mahasiswa.
5. *Outlier detection*; metode yang dapat digunakan untuk menganalisis masalah dari pembelajaran mahasiswa, penyimpangan perilaku maupun identifikasi proses pembelajaran yang tidak teratur.
6. *Regression*; metode yang digunakan untuk menentukan hubungan antara variabel dependen atau variabel yang lebih independen, serta menentukan bagaimana hubungan tersebut dapat berkontribusi pada hasil belajar mahasiswa.
7. *Sequential mining*; metode *data mining* yang digunakan untuk menemukan hubungan urutan tertentu dalam kemunculan suatu peristiwa/fenomena dari data. Metode ini dapat digunakan untuk mempersonalisasi rekomendasi perkuliahan, serta memodelkan pengetahuan dan tindakan mahasiswa.

8. *Statistics*; menggunakan perangkat lunak statistika untuk pengumpulan, analisis, interpretasi, dan penyajian data. Metode statistika dapat digunakan untuk mengevaluasi perilaku belajar serta pengembangan strategi pembelajaran.
9. *Text mining*; metode yang digunakan untuk menggali pola menarik dari suatu data teks. Dalam konteks pendidikan, teknik ini dapat digunakan untuk memodelkan pembelajaran kolaboratif, hingga mengeksplorasi perbedaan hasil belajar mahasiswa berdasarkan latar belakang masing-masing mahasiswa.
10. *Visual mining*; menggabungkan teknik *data mining* dengan alat visualisasi untuk memvisualisasikan pola data. *Visual data mining* dapat digunakan untuk mengevaluasi perilaku mahasiswa, partisipasinya, dan kinerja mereka selama perkuliahan.

Kamus konsep *analytics* dapat dilihat pada Tabel 4.1.

Tabel 4.1 Kamus Konsep *"Analytics"*

| **Nama Kelas** | **Daftar Objek** |
|---|---|
| Behavior | ActionModeling \| KnowledgeModeling \| PatternModeling |
| Learning | CollaborativeLearning \| LearningAssesment \| PersonalizedLearning \| SocialLearning |
| Predictive | CourseSelection \| DropoutRetention \| MaterialEvaluation \| StudentsPerformance |
| Methods | AssociationRule Classification \| Clustering \| CorrelationMining OutlierDetection \| Regression \| SequentialMining \| Statistics \| TextMining \| VisualMining |

#### 4.1.1.2 Konsep "Data Source"

Pada konsep *data source*, terdapat tiga kelas yang dapat digunakan dalam ontologi. Kelas tersebut yaitu Structured, SemiStructured dan Unstructured. Kelas Structured digunakan untuk memuat pengetahuan mengenai data terstruktur apa saja yang dapat digunakan untuk menganalisis data. Kelas SemiStructured memuat pengetahuan mengenai data jenis semi terstruktur apa saja yang dapat dipakai untuk analisis. Kelas Unstructured yang memuat pengetahuan mengenai data tak terstruktur yang dapat dipakai untuk analisis big data.

Menurut Bai et al. (2021), terdapat beberapa sumber yang dapat dijadikan sebagai sumber data untuk menganalisis *big data* di perguruan tinggi. Pada penelitian, sumber data dikelompokkan menjadi tiga jenis, yaitu data terstruktur (*structured*), data tidak terstruktur (*unstructured*) dan data semi terstuktur. Berikut sumber data yang dapat digunakan untuk *analytics*:

1. Data admisi, yaitu sumber data yang berisi pendaftaran peserta didik, riwayat hidup, riwayat pendidikan, dan data apa saja yang terkait dengan pendaftaran.

2. Data penilaian, yaitu sumber data mengenai proses dan hasil belajar mahasiswa yang menilai seberapa baik kinerja mahasiswa.
3. Data demografis, berisi mengenai keadaan sosio-ekonomi peserta didik, seperti usia, jenis kelamin, suku dan data lain yang relevan dengan bidang atau tujuan tertentu.
4. Data pengumpulan tugas, yaitu data mengenai ketepatan waktu peserta didik dalam menyerahkan tugas-tugas yang sudah diberikan.
5. Data sosial media, merupakan aktivitas peserta didik di media sosial. Umumnya diperoleh dari *platform* terbuka seperti Twitter, Instagram, Facebook dan lainnya.
6. Data aktivitas daring, aktivitas daring peserta didik dalam sistem LMS. Aktivitas tersebut dapat berupa interaksi dengan peserta didik lainnya, dengan materi yang diberikan dan data yang serupa.

Kamus konsep *data source* dapat dilihat pada Tabel 4.2.

Tabel 4.2 Kamus Konsep *"Data Source"*

| **Nama Kelas** | **Daftar Objek** |
|---|---|
| Structured | AdmissionData \| AssesmentData \| DemographicInformation \| PriorEducation \| StudentSelfAssesment \| SurveyData |
| SemiStructured | LearningMgmt \| TimeOfSubmission |
| Unstructured | SocialMediaData \| OnlineActivities |

#### 4.1.1.3 Konsep "Platforms"

Dari konsep *platforms*, dapat dijabarkan menjadi beberapa kelas yang memuat pengetahuan. Terdapat sepuluh kelas yang memuat pengetahuan mengenai perangkat apa saja yang dapat digunakan untuk analisis *big data*. Kelas-kelas tersebut yaitu Coordination, Deployment, Scheduler, Ingestion, MapReduce, Querying, Resource, SQL-based, NoSQL-based, dan FileSystem.

Menurut Oussous et al. (2019), terdapat beberapa perangkat yang dapat digunakan untuk mengelola dan menganalisis *big data.* Pada dasarnya, perangkat tersebut dapat dibagi menjadi tiga bagian, yaitu perangkat untuk mengelola (*management*), perangkat untuk mengolah (*processing*), dan perangkat untuk penyimpanan (*storage*). Berikut ini perangkat yang digunakan untuk mengelola dan menganalisis *big data.*

1. Perangkat untuk mengelola (*management*)
   a. *Coordination*; dalam mengkoordinasikan pengelolaan *big data*, dapat menggunakan *tools* Apache ZooKeeper yang menyediakan layanan informasi konfigurasi, penamaan, sinkronasisasi, sehingga mudah dilakukan pengelolaan dengan lebih andal.

b. *Deployment*; Apache Ambari adalah aplikasi yang menyediakan pengelolaan dan pemantauan kluster Hadoop melalui *web dashboard* yang disediakan, serta mengintegrasikan Hadoop dengan infrastruktur yang ada.
c. *Scheduler*; Apache Oozie digunakan sebagai sistem penjadwalan alur kerja untuk menjalankan dan mengelola pekerjaan di klaster Hadoop.

2. Perangkat untuk mengolah (*processing*)
   a. *Ingestion*; Apache Flume adalah aplikasi yang dapat digunakan untuk mengumpulkan, mengagregasi, dan mentransformasi data dalam jumlah besar ke dalam HDFS/HBase.
   b. *MapReduce*; aplikasi Cascading menyediakan komponen untuk pengembangan, pengujian, dan integrasi *big data* di Hadoop berdasarkan teknik MapReduce yang digunakan untuk mendefinisikan dan menjalankan alur kerja pengolahan data.
   c. *Querying*; Apache Hive menyediakan HiveQL untuk mengakses dan memanipulasi *big data* dari HDFS/HBase dengan merepresentasikan data dalam basis data terstruktur yang lebih familiar bagi penggua.
   d. *Resource*; Apache Hadoop YARN digunakan untuk mengalokasikan sumber daya dalam Hadoop yang akan digunakan untuk pengolahan *big data* ke berbagai aplikasi yang berjalan di klaster Hadoop dan menjadwalkan tugas untuk dieksekusi pada *node* klaster yang berbeda.
3. Perangkat untuk penyimpanan (*storage*)
   a. *Databases*; merupakan perangkat untuk menyimpan data dalam bentuk basis data relasional. Jenis basis data dibagi menjadi berbasis SQL dan tidak berbasis SQL. Basis data berbasis SQL seperti MySQL dan MariaDB digunakan untuk menangani data terstruktur, sedangkan basis data No-SQL seperti HBase dan MongoDB digunakan untuk menangani data semi-terstruktur maupun tidak terstruktur. HBase dapat digunakan untuk menangani data dalam jumlah besar yang disimpan dalam klaster Hadoop.
   b. *File System*; HDFS adalah sistem penyimpanan data dalam bentuk *file* untuk menangani data jumlah besar dengan memecah *file* yang disimpan dalam klaster Hadoop menjadi bagian-bagian lebih kecil dan kemudian didistribusikan ke klaster-klaster sehingga memungkinkan pemrosesan secara pararel.

Kamus konsep *platforms* dapat dilihat pada Tabel 4.3.

Tabel 4.3 Kamus Konsep *"Platforms"*

| **Nama Kelas** | **Daftar Objek** |
|---|---|
| Coordination | Zookeper |
| Deployment | Ambari |
| Scheduler | Oozie |
| Ingestion | Flume |
| MapReduce | Cascading |
| Querying | Hive |
| Resource | YARN |
| SQL-based | MySQL \| MariaDB |
| NoSQL-based | HBase \| MongoDB |
| FileSystem | HDFS |

### 4.1.2 Taksonomi

Pada tahapan ini dibentuk taksonomi sebagai cara untuk mengorganisasi seluruh kelas yang ada dalam ontologi. Pada Analytics, kelas Objectives dibuat sebagai kelas induk dari kelas Behavior, Learning, dan Predictive. Kelas-kelas tersebut dikumpulkan karena mempunyai kesamaan sebagai sasaran/*objectives* dari kegiatan *analytics*. Kelas-kelas lainnya yang akan dikumpulkan, yaitu kelas Coordination, Deployment, dan Scheduler. Kelas-kelas tersebut dikumpulkan dalam kelas Management karena mempunyai kesamaan sebagai *platform* yang mempunyai aplikasi untuk digunakan dalam memanajemen *big data* sebelum diolah, berdasarkan fungsi aplikasi untuk mengkoordinasikan pengelolaan, penyediaan *dashboard*, dan penjadwalan alur kerja. Kelas lainnya yaitu Ingestion, MapReduce, Querying, dan Resource dikumpulkan ke dalam kelas Processing karena sama-sama bagian dari *platform* untuk memproses *big data*.

Kelas lain yaitu SQL-based, NoSQL-based, dan FileSystem dikumpulkan dalam kelas Storage karena mempunyai kesamaan sebagai platform untuk penyimpanan *big data*. Namun, dalam hal ini khusus untuk kelas SQL-based dan NoSQL-based dikumpulkan sebagai kelas dari Databases karena sama-sama sebagai basis data yang dapat digunakan untuk menyimpan big data. Sedangkan FileSystem sebagai platform teresendiri yang khusus untuk data besar yang telah diolah melalui *database* ke dalam sistem Hadoop. Seluruh kelas dan hierarkinya dapat dilihat pada Tabel 4.4.

| **1** | **Analytics** | **2** | **DataSource** | **3** | **Platforms** |
|---|---|---|---|---|---|
| 1.1 | Objectives | 2.1 | Structured | 3.1 | Management |
| 1.1.1 | Behavior | 2.2 | SemiStructured | 3.1.1 | Coordination |
| 1.1.2 | Learning | 2.3 | Unstructured | 3.1.2 | Deployment |
| 1.1.3 | Predictive | | | 3.1.3 | Scheduler |

| 1 | **Analytics** | 2 | **DataSource** | 3 | **Platforms** |
|---|---|---|---|---|---|
| 1.1 | Objectives | 2.1 | Structured | 3.2 | Processing |
| 1.1.1 | Behavior | 2.2 | SemiStructured | 3.2.1 | Ingestion |
| 1.1.2 | Learning | 2.3 | Unstructured | 3.2.2 | MapReduce |
| 1.1.3 | Predictive | | | 3.2.3 | Querying |
| 1.2 | Methods | | | 3.2.4 | Resource |
| | | | | 3.3 | Storage |
| | | | | 3.3.1 | Databases |
| | | | | 3.3.1.1 | SQL-based |
| | | | | 3.3.1.2 | NoSQL-based |
| | | | | 3.3.2 | FileSystem |

Agar taksonomi lebih mudah dipahami, dilakukan pemetaan relasi *is-a*. Sebagai contoh, SQL-based *is-a* Databases *is-a* Storage *is-a* Platforms. Ini berarti bahwa kelas SQL-based merupakan sub-kelas dari Databases yang merupakan sub-kelas dari Storage di mana kelas tersebut merupakan sub-kelas dari kelas Platforms. Visualisasi pemetaan taksonomi dapat dilihat pada Gambar 4.1.




Gambar 4.1 Taksonomi Ontologi

### 4.1.3 Properti

Setelah melakukan pembentukan kelas dan hierarkinya, dilanjutkan dengan membentuk properti, termasuk karakteristik dari properti tersebut. Berdasarkan kelas dan hierarki yang telah dibentuk terdapat enam pasang properti *functional* serta *inverse* yang digunakan untuk menghubungkan satu kelas dengan kelas lainnya yang ada dalam ontologi.

Menurut Marín et al. (2019) dan Ang et al. (2020), sasaran *analytics* di perguruan tinggi mempunyai beberapa metode yang dapat diterapkan dan sumber data yang dapat digunakan. Pada penelitian ontologi ini, pemberian nama properti menggunakan penamaan yang secara umum diberikan dalam berbagai penelitian terkait pengembangan model ontologi. Untuk menghubungkan kelas Objectives sebagai *domain* dengan kelas Methods sebagai *range*, maka digunakan sebuah properti yang mempunyai karakter fungsional yang memuat nilai objek. Properti tersebut diberi nama hasMethods.

Menurut Baig et al. (2020), sumber data (*data sources*) yang digunakan untuk *big data analytics* dapat disimpan dalam suatu basis data (*databases*) berdasarkan jenis sumber data tersebut. Pada penelitian ontologi ini, hal tersebut dikonseptualisasikan dalam bentuk kelas DataSources sebagai *domain* yang dihubungkan dengan kelas Databases sebagai *range* melalui properti hasDatabaseFor maupun *inverse* dari properti tersebut. Untuk membentuk hubungan sasaran *analytics* berdasarkan dari sumber data (*data sources*) yang digunakan, maka kelas Objectives sebagai *domain* dihubungkan dengan kelas DataSources sebagai *range* melalui properti hasDataSource, berlaku sebaliknya untuk *inverse* dari properti tersebut.

Menurut Oussous et al. (2019), aplikasi dari *platform big data* memiliki hubungan dengan aplikasi lain dari *platform* tersebut sehingga dapat digunakan untuk menangani *big data*. Aplikasi untuk melakukan teknik *MapReduce* memproses data yang disimpan dalam bentuk *file system*. Pada ontologi ini, hal tersebut dikonseptualisasi dengan menghubungkan kelas MapReduce dengan kelas FileSystem melalui properti hasProcessFor. Selain itu, data tersebut ditarik dari basis data melalui aplikasi yang menggunakan bahasa *query* khusus. Hal tersebut dikonseptualisasi dengan menghubungkan kelas Querying kepada kelas Storage yang memuat basis data tersebut, melalui properti hasQueryingFor atau sebaliknya jika melalui properti *inverse*. Kemudian data dari basis data tersebut diolah oleh berbagai aplikasi yang digunakan untuk mengolah data tersebut, yang dikonseptualisasi dengan properti hasIngestFor terhadap kelas Processing kepada kelas Databases.

Daftar properti yang digunakan dapat dilihat pada Tabel 4.5.

Tabel 4.5 Daftar Properti

| **Property** | **Domain** | **Range** | **Character** | **Value** |
|---|---|---|---|---|
| hasMethods | Objectives | Methods | Functional | Object |
| hasDatabaseFor | DataSources | Databases | Functional | Object |
| hasDataSource | Objectives | DataSources | Functional | Object |
| hasProcessFor | MapReduce | FileSystem | Functional | Object |
| hasQueryingFor | Querying | Storage | Functional | Object |
| hasIngestFor | Processing | Databases | Functional | Object |
| isDatabaseOf | Databases | DataSources | Inverse | Object |
| isDataSourceFor | DataSources | Objectives | Inverse | Object |
| isMethodOf | Methods | Objectives | Inverse | Object |
| isProcessBy | FileSystem | MapReduce | Inverse | Object |
| isQueryingBy | Storage | Querying | Inverse | Object |
| isIngestedBy | Databases | Processing | Inverse | Object |

## 4.2 Implementasi

Implementasi dari hasil konseptualisasi dilakukan menggunakan bahasa OWL (Web Ontology Language) dalam skema XML, yang dapat disebut juga sebagai OWL/XML. Penulisan kode OWL/XML didahului dengan deklarasi `<?xml version="1.0"?>` sebagai versi XML yang digunakan. Kemudian dilanjutkan dengan pembukaan *tag* ontologi, yaitu `<Ontology>` dan diakhiri dengan penutupan tag yaitu `</Ontology>`. Penulisan seluruh kode OWL dilakukan di dalam `<Ontology>..</Ontology>` seperti ditunjukkan pada Gambar 4.2.

```
1  <?xml version="1.0"?>
2  <Ontology xmlns="http://www.w3.org/2002/07/owl#"
3      xml:base="http://www.semanticweb.org/17917131/ontologies/2021/01/BigData-Edu"
4      xmlns:xml="http://www.w3.org/XML/1998/namespace"
5      xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
6      ontologyIRI="http://www.semanticweb.org/17917131/ontologies/2021/01/BigData-Edu">
7      <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
8      <Prefix name="BigData-Edu" IRI="http://www.semanticweb.org/17917131/ontologies/2021/01/BigData-Edu#"/>
9  .
10 .
11 .
12 </Ontology>
```

Gambar 4.2 Deklarasi Skema OWL/XML

Dari gambar tersebut, dapat dilihat pada baris pertama, dilakukan deklarasi versi XML. Baris kedua menunjukkan pembukaan ontologi, dilanjutkan dengan deklarasi skema XML yang akan dilakukan *coding* OWL. Baris ketiga sampai kelima dilakukan deklarasi *resource location* dari ontologi, dalam hal ini diberikan nama BigData-Edu. Baris keenam dilakukan deklarasi IRI (*Internationalized Resource Identifiers*) yang memuat *resource*

*location* dari ontologi. Selanjutnya pada baris ketujuh dan kedelapan dilakukan deklarasi Prefix untuk mempersingkat *resource location* tersebut. Setelah selesai melakukan deklarasi OWL/XML, dilanjutkan dengan menuliskan kode ontologi.

### 4.2.1 Pembentukan Kelas & Properti

Berdasarkan hasil konseptualisasi, kelas pada Analytics terdiri dari enam kelas, kelas pada DataSource terdiri empat kelas, kemudian kelas pada Platforms terdiri dari lima belas kelas. Deklarasi kelas dimulai dengan menulis `<Declaration>` dan ditutup dengan `</Declaration>`. Untuk penulisan nama kelas, dimulai dengan Class kemudian prefix dari IRI, lalu nama kelas, kemudian ditutup dengan tanda *slash*. Contoh penulisan: `<Declaration> <Class BigData-Edu="#Analytics"/> </Declaration>`. Setelah melakukan deklarasi kelas, dilakukan penulisan kode untuk sub-kelas dari masing-masing kelas. Penulisan kode sub-kelas didahului dengan `<SubClassOf>` kemudian menulis nama sub-kelas setelah itu nama kelas induk, lalu ditutup dengan `</SubClassOf>`. Potongan kode kelas dan sub-kelas dapat dilihat pada Gambar 4.3.

```
<!--Class Declaration-->
<Declaration> <Class BigData-Edu="#Analytics"/> </Declaration> <Declaration> <Class BigData-Edu="#Objective"/>
</Declaration> <Declaration> <Class BigData-Edu="#Behavior"/> </Declaration> <Declaration> <Class BigData-
Edu="#Learning"/> </Declaration> <Declaration> <Class BigData-Edu="#Predictive"/> </Declaration>  <Declaration>
<Class BigData-Edu="#Methods"/> </Declaration> <Declaration> <Class BigData-Edu="#DataSource"/> </Declaration>
<Declaration> <Class BigData-Edu="#SemiStructured"/> </Declaration> <Declaration> <Class BigData-
Edu="#Structured"/> </Declaration> <Declaration> <Class BigData-Edu="#Unstructured"/> </Declaration> <Declaration>
<Class BigData-Edu="#Platforms"/> </Declaration> <Declaration> <Class BigData-Edu="#Management"/> </Declaration>
<Declaration> <Class BigData-Edu="#Coordination"/> </Declaration> <Declaration> <Class BigData-Edu="#Deployment"/>
</Declaration> <Declaration> <Class BigData-Edu="#Scheduler"/> </Declaration> <Declaration> <Class BigData-
Edu="#Processing"/> </Declaration> <Declaration> <Class BigData-Edu="#Ingestion"/> </Declaration> <Declaration>
<Class BigData-Edu="#MapReduce"/> </Declaration> <Declaration> <Class BigData-Edu="#Querying"/> </Declaration>
<Declaration> <Class BigData-Edu="#Resource"/> </Declaration> <Declaration> <Class BigData-Edu="#Storage"/>
</Declaration> <Declaration> <Class BigData-Edu="#FileSystem"/> </Declaration> <Declaration> <Class BigData-
Edu="#Databases"/> </Declaration> <Declaration> <Class BigData-Edu="#SQL-based"/> </Declaration> <Declaration>
<Class BigData-Edu="#NoSQL-based"/> </Declaration>
<!--Subclass Declaration-->
<SubClassOf> <Class BigData-Edu="#SemiStructured"/> <Class BigData-Edu="#DataSource"/> </SubClassOf> <SubClassOf>
<Class BigData-Edu="#Structured"/> <Class BigData-Edu="#DataSource"/> </SubClassOf> <SubClassOf> <Class BigData-
Edu="#Unstructured"/> <Class BigData-Edu="#DataSource"/> </SubClassOf> <SubClassOf> <Class BigData-
Edu="#Objective"/> <Class BigData-Edu="#Analytics"/> </SubClassOf> <SubClassOf> <Class BigData-Edu="#Methods"/>
<Class BigData-Edu="#Analytics"/> </SubClassOf>          <SubClassOf> <Class BigData-Edu="#Behavior"/> <Class
BigData-Edu="#Objective"/>      </SubClassOf> <SubClassOf> <Class BigData-Edu="#Learning"/> <Class BigData-
Edu="#Objective"/>      </SubClassOf> <SubClassOf> <Class BigData-Edu="#Predictive"/> <Class BigData-
Edu="#Objective"/> </SubClassOf> <SubClassOf> <Class BigData-Edu="#Management"/> <Class BigData-Edu="#Platforms"/>
</SubClassOf> <SubClassOf> <Class BigData-Edu="#Processing"/> <Class BigData-Edu="#Platforms"/> </SubClassOf>
<SubClassOf> <Class BigData-Edu="#Storage"/> <Class BigData-Edu="#Platforms"/> </SubClassOf> <SubClassOf> <Class
BigData-Edu="#Coordination"/> <Class BigData-Edu="#Management"/> </SubClassOf> <SubClassOf> <Class BigData-
Edu="#Deployment"/> <Class BigData-Edu="#Management"/> </SubClassOf> <SubClassOf> <Class BigData-Edu="#Scheduler"/>
<Class BigData-Edu="#Management"/> </SubClassOf> <SubClassOf> <Class BigData-Edu="#Ingestion"/> <Class BigData-
Edu="#Processing"/> </SubClassOf> <SubClassOf> <Class BigData-Edu="#MapReduce"/> <Class BigData-Edu="#Processing"/>
</SubClassOf> <SubClassOf> <Class BigData-Edu="#Querying"/> <Class BigData-Edu="#Processing"/> </SubClassOf>
<SubClassOf> <Class BigData-Edu="#Resource"/> <Class BigData-Edu="#Processing"/> </SubClassOf> <SubClassOf> <Class
BigData-Edu="#FileSystem"/> <Class BigData-Edu="#Storage"/> </SubClassOf> <SubClassOf> <Class BigData-
Edu="#Databases"/> <Class BigData-Edu="#Storage"/> </SubClassOf> <SubClassOf> <Class BigData-Edu="#SQL-based"/>
<Class BigData-Edu="#Databases"/> </SubClassOf> <SubClassOf> <Class BigData-Edu="#NoSQL-based"/> <Class BigData-
Edu="#Databases"/> </SubClassOf>
```

Gambar 4.3 Kelas dalam Kode OWL/XML

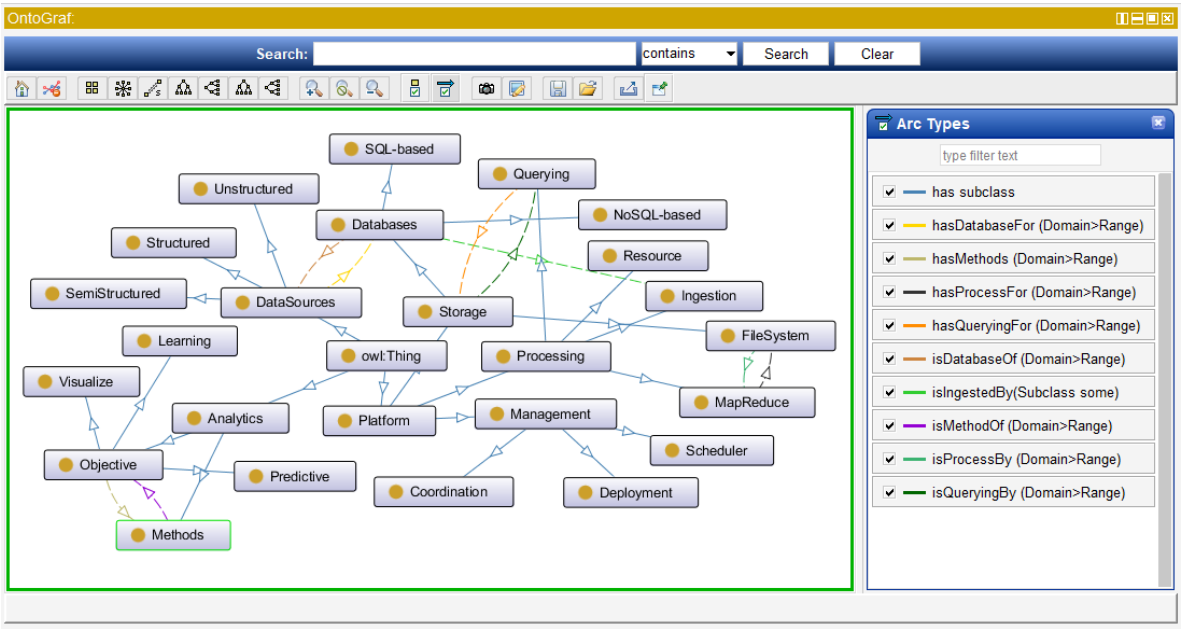
Setelah menulis kode kelas, dapat dilanjutkan dengan menulis kode untuk properti. Berdasarkan hasil konseptualisasi, ontologi mempunyai sepuluh properti. Penulisan kode properti dimulai dengan deklarasi `<Declaration>` dan ditutup dengan `</Declaration>`. Penulisan nama properti dimulai dengan menulis

`ObjectProperty` kemudian prefix dari IRI, lalu nama kelas, kemudian ditutup dengan tanda *slash*. Contoh penulisan: `<ObjectProperty BigData-Edu="#hasMethod"/>`. Setelah mendeklarasi semua properti, dilanjutkan dengan menuliskan karakteristik properti. Penulisan karakteristik dimulai dengan menulis jenis karakteristik kemudian nama properti. Contoh penulisan: `<FunctionalObjectProperty> <ObjectProperty BigData-Edu="#hasIngestFor"/> </FunctionalObjectProperty>`. Potongan kode penulisan properti dapat dilihat pada Gambar 4.4.

```
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#ActionModeling"/> <NamedIndividual
IRI="#AssociationRule"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#ActionModeling"/> <NamedIndividual IRI="#CausalMining"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#ActionModeling"/> <NamedIndividual
IRI="#Classification"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#ActionModeling"/> <NamedIndividual IRI="#Clustering"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#ActionModeling"/> <NamedIndividual
IRI="#CorrelationMining"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#ActionModeling"/> <NamedIndividual IRI="#OutlierDetection"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#ActionModeling"/> <NamedIndividual
IRI="#Regression"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#ActionModeling"/> <NamedIndividual IRI="#SequentialMining"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion>
<ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#ActionModeling"/> <NamedIndividual IRI="#Statistics"/>
</ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#ActionModeling"/> <NamedIndividual IRI="#VisualMining"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion>
<ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#CollaborativeLearning"/> <NamedIndividual IRI="#AssociationRule"/>
</ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#CollaborativeLearning"/> <NamedIndividual IRI="#Classification"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion>
<ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#CollaborativeLearning"/> <NamedIndividual IRI="#Clustering"/>
</ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#CollaborativeLearning"/> <NamedIndividual IRI="#CorrelationMining"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion>
<ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#CollaborativeLearning"/> <NamedIndividual IRI="#Regression"/>
</ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#CollaborativeLearning"/> <NamedIndividual IRI="#SequentialMining"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion>
<ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#CollaborativeLearning"/> <NamedIndividual IRI="#Statistics"/>
</ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#CollaborativeLearning"/> <NamedIndividual IRI="#TextMining"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion>
<ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#CollaborativeLearning"/> <NamedIndividual IRI="#VisualMining"/>
</ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#DropoutRetention"/> <NamedIndividual IRI="#Classification"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion>
<ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#DropoutRetention"/> <NamedIndividual IRI="#Clustering"/>
</ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#DropoutRetention"/> <NamedIndividual IRI="#Regression"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion>
<ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#DropoutRetention"/> <NamedIndividual IRI="#Statistics"/>
</ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#KnowledgeModeling"/> <NamedIndividual IRI="#AssociationRule"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion>
<ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#KnowledgeModeling"/> <NamedIndividual IRI="#CausalMining"/>
</ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#KnowledgeModeling"/> <NamedIndividual IRI="#Classification"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion>
<ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#KnowledgeModeling"/> <NamedIndividual IRI="#Clustering"/>
</ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#KnowledgeModeling"/> <NamedIndividual IRI="#CorrelationMining"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion>
<ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#KnowledgeModeling"/> <NamedIndividual IRI="#OutlierDetection"/>
</ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#KnowledgeModeling"/> <NamedIndividual IRI="#Regression"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion>
<ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#KnowledgeModeling"/> <NamedIndividual IRI="#SequentialMining"/>
</ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#KnowledgeModeling"/> <NamedIndividual IRI="#Statistics"/> </ObjectPropertyAssertion>
```

Gambar 4.4 Properti dalam Kode OWL/XML

Agar mudah dipahami, maka dilakukan visualisasi untuk menunjukkan hubungan antara kelas yang satu dengan kelas yang lainnya. Hasil visualisasi kelas dan properti dalam ontologi dapat dilihat pada Gambar 4.5.




Gambar 4.5 Visualisasi Kelas dan Properti

### 4.2.2 Pembentukan Objek Kelas

Berdasarkan hasil konseptualisasi, kelas yang mempunyai objek terdiri dari 17 kelas. Pembentukan objek ontologi dimulai dengan deklarasi `<Declaration>` dan ditutup dengan `</Declaration>`. Penulisan nama objek dimulai dengan menulis `NamedIndividual` kemudian prefix IRI, lalu nama objek, kemudian ditutup dengan *slash*. Contoh penulisan: `<NamedIndividual BigData-Edu="#ActionModeling"/>`.

```
<ClassAssertion> <Class IRI="#Behavior"/> <NamedIndividual IRI="#ActionModeling"/> </ClassAssertion> <ClassAssertion> <Class
IRI="#Structured"/> <NamedIndividual IRI="#AdmissionData"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Deployment"/>
<NamedIndividual IRI="#Ambari"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Structured"/> <NamedIndividual IRI="#AssesmentData"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Methods"/> <NamedIndividual IRI="#AssociationRule"/> </ClassAssertion> <ClassAssertion>
<Class IRI="#MapReduce"/> <NamedIndividual IRI="#Cascading"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Methods"/>
<NamedIndividual IRI="#CausalMining"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Ingestion"/> <NamedIndividual IRI="#Chukwa"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Methods"/> <NamedIndividual IRI="#Classification"/> </ClassAssertion> <ClassAssertion>
<Class IRI="#Methods"/> <NamedIndividual IRI="#Clustering"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Learning"/>
<NamedIndividual IRI="#CollaborativeLearning"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Methods"/> <NamedIndividual
IRI="#CorrelationMining"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Predictive"/> <NamedIndividual IRI="#CourseSelection"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Structured"/> <NamedIndividual IRI="#DemographicInformation"/> </ClassAssertion>
<ClassAssertion> <Class IRI="#Methods"/> <NamedIndividual IRI="#DensityEstimation"/> </ClassAssertion> <ClassAssertion> <Class
IRI="#Predictive"/> <NamedIndividual IRI="#DropoutRetention"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Structured"/>
<NamedIndividual IRI="#EnrollmentData"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Ingestion"/> <NamedIndividual IRI="#Flume"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#NoSQL-based"/> <NamedIndividual IRI="#HBase"/> </ClassAssertion> <ClassAssertion>
<Class IRI="#FileSystem"/> <NamedIndividual IRI="#HDFS"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Querying"/> <NamedIndividual
IRI="#Hive"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Behavior"/> <NamedIndividual IRI="#KnowledgeModeling"/> </ClassAssertion>
<ClassAssertion> <Class IRI="#Learning"/> <NamedIndividual IRI="#LearningAssesment"/> </ClassAssertion> <ClassAssertion> <Class
IRI="#Unstructured"/> <NamedIndividual IRI="#LearningMgmt"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Predictive"/>
<NamedIndividual IRI="#MaterialEvaluation"/> </ClassAssertion> <ClassAssertion> <Class IRI="#NoSQL-based"/> <NamedIndividual
IRI="#MongoDB"/> </ClassAssertion> <ClassAssertion> <Class IRI="#SQL-based"/> <NamedIndividual IRI="#MySQL"/> </ClassAssertion>
<ClassAssertion> <Class IRI="#NoSQL-based"/> <NamedIndividual IRI="#Neo4j"/> </ClassAssertion> <ClassAssertion> <Class
IRI="#Unstructured"/> <NamedIndividual IRI="#OnlineActivities"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Scheduler"/>
<NamedIndividual IRI="#Oozie"/> </ClassAssertion> <ClassAssertion> <Class IRI="#SQL-based"/> <NamedIndividual IRI="#Oracle"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Methods"/> <NamedIndividual IRI="#OutlierDetection"/> </ClassAssertion>
<ClassAssertion> <Class IRI="#Behavior"/> <NamedIndividual IRI="#PatternModeling"/> </ClassAssertion> <ClassAssertion> <Class
IRI="#Learning"/> <NamedIndividual IRI="#PersonalizedLearning"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Querying"/>
<NamedIndividual IRI="#Pig"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Structured"/> <NamedIndividual IRI="#PriorEducation"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#NoSQL-based"/> <NamedIndividual IRI="#Redis"/> </ClassAssertion> <ClassAssertion>
<Class IRI="#Methods"/> <NamedIndividual IRI="#Regression"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Methods"/>
<NamedIndividual IRI="#SequentialMining"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Learning"/> <NamedIndividual
IRI="#SocialLearning"/> </ClassAssertion> <ClassAssertion> <Class IRI="#SemiStructured"/> <NamedIndividual IRI="#SocialMediaData"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Ingestion"/> <NamedIndividual IRI="#Sqoop"/> </ClassAssertion> <ClassAssertion> <Class
IRI="#Methods"/> <NamedIndividual IRI="#Statistics"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Structured"/> <NamedIndividual
IRI="#StudentSelfAssesment"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Predictive"/> <NamedIndividual
IRI="#StudentsPerformance"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Structured"/> <NamedIndividual IRI="#SurveyData"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Methods"/> <NamedIndividual IRI="#TextMining"/> </ClassAssertion> <ClassAssertion>
<Class IRI="#Structured"/> <NamedIndividual IRI="#TimeOfSubmission"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Methods"/>
<NamedIndividual IRI="#VisualMining"/> </ClassAssertion> <ClassAssertion> <Class IRI="#Resource"/> <NamedIndividual IRI="#YARN"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Coordination"/> <NamedIndividual IRI="#Zookeper"/> </ClassAssertion>
```

Gambar 4.6 Objek dalam Kode OWL/XML

Setelah mendeklarasikan semua objek, dilakukan penulisan kode kelas yang memuat objek, kemudian propertinya. Potongan kode objek dapat dilihat pada Gambar 4.6. Setelah menulis kode seluruh objek, dapat dilanjutkan dengan menulis kode untuk relasi setiap objek. Penulisan relasi objek dimulai dengan `<ObjectPropertyAssertion>` dan ditutup dengan `</ObjectPropertyAssertion>`. Setelah mendeklarasi properti, dilanjutkan dengan menulis nama properti penghubung, kemudian nama kedua objek yang direlasikan. Contoh: `<ObjectPropertyAssertion><ObjectProperty:IRI="#hasMethods"/> <NamedIndividual:IRI="#ActionModeling"/><NamedIndividual:IRI="#AssociationRule"/> </ObjectPropertyAssertion>`.

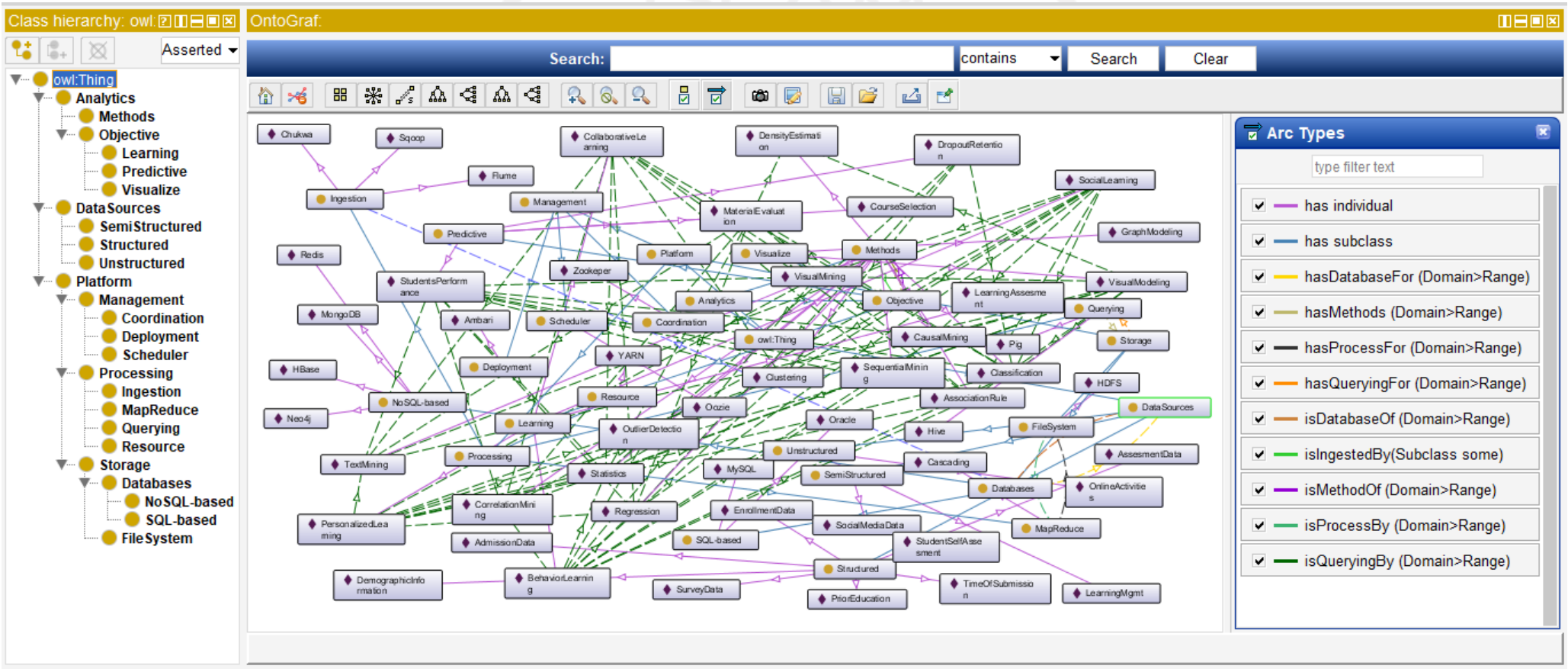
```
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#ActionModeling"/> <NamedIndividual
IRI="#AssociationRule"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#ActionModeling"/> <NamedIndividual IRI="#CausalMining"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty
IRI="#hasMethods"/> <NamedIndividual IRI="#ActionModeling"/> <NamedIndividual IRI="#Classification"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#ActionModeling"/> <NamedIndividual
IRI="#Clustering"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#ActionModeling"/> <NamedIndividual IRI="#CorrelationMining"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion>
<ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#ActionModeling"/> <NamedIndividual IRI="#OutlierDetection"/>
</ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#ActionModeling"/>
<NamedIndividual IRI="#Regression"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#ActionModeling"/> <NamedIndividual IRI="#SequentialMining"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#ActionModeling"/> <NamedIndividual
IRI="#Statistics"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#ActionModeling"/> <NamedIndividual IRI="#VisualMining"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty
IRI="#hasMethods"/> <NamedIndividual IRI="#CollaborativeLearning"/> <NamedIndividual IRI="#AssociationRule"/>
</ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#CollaborativeLearning"/> <NamedIndividual IRI="#Classification"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion>
<ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#CollaborativeLearning"/> <NamedIndividual IRI="#Clustering"/>
</ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#CollaborativeLearning"/> <NamedIndividual IRI="#CorrelationMining"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion>
<ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#CollaborativeLearning"/> <NamedIndividual IRI="#Regression"/>
</ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#CollaborativeLearning"/> <NamedIndividual IRI="#SequentialMining"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion>
<ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#CollaborativeLearning"/> <NamedIndividual IRI="#Statistics"/>
</ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#CollaborativeLearning"/> <NamedIndividual IRI="#TextMining"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion>
<ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#CollaborativeLearning"/> <NamedIndividual IRI="#VisualMining"/>
</ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#DropoutRetention"/>
<NamedIndividual IRI="#Classification"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#DropoutRetention"/> <NamedIndividual IRI="#Clustering"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion>
<ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#DropoutRetention"/> <NamedIndividual IRI="#Regression"/>
</ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#DropoutRetention"/>
<NamedIndividual IRI="#Statistics"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#KnowledgeModeling"/> <NamedIndividual IRI="#AssociationRule"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#KnowledgeModeling"/> <NamedIndividual
IRI="#CausalMining"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#KnowledgeModeling"/> <NamedIndividual IRI="#Classification"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion>
<ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#KnowledgeModeling"/> <NamedIndividual IRI="#Clustering"/>
</ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#KnowledgeModeling"/>
<NamedIndividual IRI="#CorrelationMining"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#KnowledgeModeling"/> <NamedIndividual IRI="#OutlierDetection"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#KnowledgeModeling"/> <NamedIndividual
IRI="#Regression"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/> <NamedIndividual
IRI="#KnowledgeModeling"/> <NamedIndividual IRI="#SequentialMining"/> </ObjectPropertyAssertion> <ObjectPropertyAssertion>
<ObjectProperty IRI="#hasMethods"/> <NamedIndividual IRI="#KnowledgeModeling"/> <NamedIndividual IRI="#Statistics"/>
```

Gambar 4.7 Kode OWL/XML dari Relasi Setiap Objek

Hasil penulisan kode dapat dilihat pada Gambar 4.7. Setelah mendeklarasi properti, dilanjutkan dengan menulis nama properti penghubung, kemudian nama kedua objek yang direlasikan. Contoh:

`<ObjectPropertyAssertion><ObjectProperty:IRI="#hasMethods"/> <NamedIndividual:IRI="#ActionModeling"/><NamedIndividual:IRI="#AssociationRule"/> </ObjectPropertyAssertion>`.

Hasil penulisan kode dapat dilihat pada Gambar 4.7. Untuk memetakan relasi antar objek, dapat dilakukan visualisasi. Adapun hasil visualisasi dari elemen ontologi dapat dilihat pada Gambar 4.8.




Gambar 4.8 Visualisasi Relasi Antar Objek

## 4.3 Evaluasi

### 4.3.1 Analisis Ontologi

Dilakukan evaluasi schema metrics terhadap ontologi pada penelitian ini yang mengukur Relationship Richness (RR), Inheritance Richness (IR), dan Attribute Richness (AR). Nilai yang digunakan untuk melakuakan pengukuran didapatkan dari nilai ontology metrics seperti yang terdapat pada Tabel 4.6.

Tabel 4.6 *Ontology Metrics*

| No | Ontology Element | Count |
|---|---|---|
| 1 | Superclass | 9 |
| 2 | Subclass | 16 |
| 3 | Object Property | 10 |
| 4 | Datatype Property | 1 |

Adapun pengukuran ontologi sebagai berikut.

#### 4.3.1.1 Pengukuran RR

Pada ontologi ini terdapat 10 relasi *object property* dan 16 relasi *subclass*. Hasil perhitungan RR pada ontologi ini, yaitu:

$$RR = \frac{16}{17 + 10} = 0{,}62$$

Hasil pengukuran *Relationship Richness* (RR) menunjukkan bahwa nilai mendekati 1,00 yang dapat disimpulkan bahwa pengetahuan yang terdapat dalam ontologi sudah kaya. Hal ini dapat ditinjau dari banyaknya hubungan kelas dan sub-kelas yaitu 16 buah dan properti untuk menghubungkan antara kelas yang satu dengan lainnya sebanyak 17 buah. Dengan banyaknya relasi kelas-subkelas mendukung banyak dan beragam pengetahuan yang dapat disimpan dalam ontologi.

#### 4.3.1.2 Pengukuran IR

Pada ontologi ini terdapat 16 *subclass* dan 9 *superclass*. Hasil perhitungan IR, yaitu:

$$IR = \frac{16}{9} = 1{,}88$$

Hasil pengukuran *Inheritance Richness* (IR) menunjukkan bahwa ontologi yang telah dibangun termasuk ontologi yang cukup spesifik, mengingat hasil pengukuran mendekati angka 0,00. Keadaan ini terjadi karena ontologi mempunyai banyak relasi *inheritance* yang mempunyai sub-kelas, yaitu 16 buah. Ontologi yang makin spesifik mengindikasikan seberapa baik konsep/pengetahuan dikelompokkan dalam berbagai kategori maupun sub-kategori.

#### 4.3.1.3 Pengukuran AR

Pada ontologi ini terdapat 01 *attribute* (*datatype property*) dan 16 *class*. Hasil perhitungan AR, yaitu.

$$AR = \frac{1}{16} = 0{,}06$$

Pengukuran *Attribute Richness* (AR) menghasilkan skor 0,06 yang menunjukkan bahwa informasi dari pengetahuan yang tersimpan dalam ontologi dapat ditingkatkan lagi. Untuk merangkum hasil pengukuran ontologi, dapat dilihat pada Tabel 4.7 berikut.

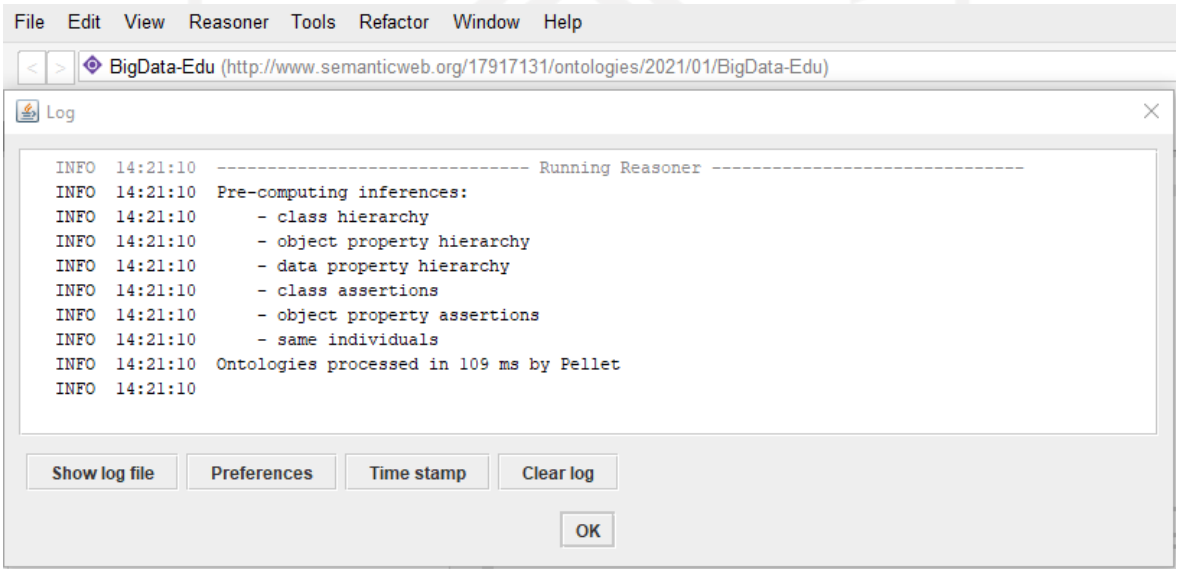
Tabel 4.7 Hasil Pengukuran Ontologi

| **Jenis Pengukuran** | **Nilai** | **Kesimpulan** |
|---|---|---|
| *Relationship Richness* (RR) | 0.62 | Keberagaman pengetahuan yang dimuat ontologi ini sudah kaya |
| *Inheritance Richness* (IR) | 1.78 | Ontologi ini termasuk dalam kategori yang cukup spesifik |
| *Attribute Richness* (AR) | 0.06 | Informasi yang dimuat ontologi dapat ditingkatkan lagi |

### 4.3.2 Analisis *Query*

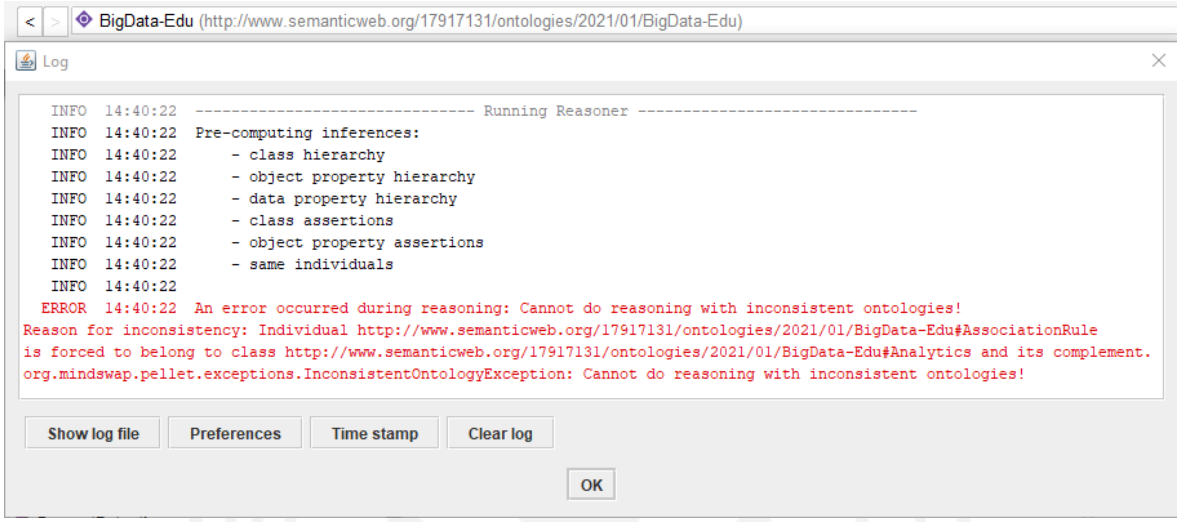
#### 4.3.2.1 Pengujian Konsistensi

Telah dilakukan pengujian konsistensi pada model ontologi yang telah dibangun menggunakan *reasoner* Pellet dari perangkat lunak Protégé. Ketika dilakukan uji konsistensi, tidak terdapat pesan *error* yang ditampilkan oleh *reasoner* Pellet melalui *software* Protégé. Hal ini berarti ontologi dapat dikatakan telah konsisten. Gambar 4.9 menunjukkan hasil pengujian konsistensi dari ontologi.




Gambar 4.9 Hasil Uji Konsistensi pada Ontologi

Pada Gambar 4.10, dilakukan skenario jika pada uji konsistensi ontologi menghasilkan *error*. Hasil *error* terjadi ketika adanya inkonsistensi pada ontologi. Hal tersebut terjadi karena tidak konsistennya nilai yang ditetapkan ketika membentuk ontologi dengan batasan yang telah ditetapkan pada ontologi. Gambar 4.10 menunjukkan hasil *error* dari uji konsistensi.




Gambar 4.10 Kondisi Uji Konsistensi Gagal

Kondisi tersebut terjadi karena objek `AssociationRule` yang merupakan objek dari kelas `Methods` dari superkelas `Analytics` seharusnya saling tidak berhubungan (*disjoint*) dengan kelas di luar kelas tersebut superkelas tersebut. Jika kondisi tersebut dipaksakan maka kegiatan *query* tidak dapat dilakukan.

#### 4.3.2.2 Pengujian *Query*

Setelah memastikan model ontologi yang dibangun telah konsisten, maka dapat dilanjutkan untuk melakukan *query* guna menarik pengetahuan yang tersimpan dalam ontologi. Dilakukan uji coba *query* menggunakan SPARQL untuk menarik pengetahuan yang tersimpan dalam ontologi. Pada model ontologi yang telah dibangun, terdapat pengetahuan mengenai sasaran *analytics* yang bertujuan untuk menganalisis kinerja akademik mahasiswa. Pengetahuan mengenai sasaran tersebut disimpan dalam bentuk objek ontologi yaitu `StudentsPerformance`. Pada penelitian ini, dilakukan penarikan pengetahuan berdasarkan beberapa pertanyaan, yaitu:

1. Metode apa yang dapat digunakan untuk menganalisis kinerja akademik mahasiswa?
2. Sumber data apa saja yang bisa digunakan untuk mencapai tujuan tersebut?
3. Teknologi basis data apa saja yang dapat digunakan untuk menampung data tersebut?
4. Sistem *file* apa yang bisa digunakan untuk menyimpan *big data* agar dapat diproses?

5. Teknik apa yang dapat digunakan untuk memproses *big data* tersebut?

Penulisan *query* diawali dengan deklarasi `PREFIX rdf`, `owl`, `rdfs`, `xsd`, dan `bigdata`. Deklarasi `PREFIX` memuat URI (*Uniform Resource Identifier*) dari ontologi web semantik. Setelah melakukan deklarasi prefix, dilanjutkan dengan menuliskan isi dari klausa SELECT. Fungsi pernyataan `SELECT` yaitu untuk mengembalikan nilai suatu tabel dari variabel dan nilai yang dideskripsikan di dalam *query*. Kemudian, variabel didahului dengan tanda tanya ("`?`"), yang merupakan *node* (*resource* atau *literal*) di dalam RDF. Untuk penamaan variabel tidak ada aturan khusus yang mengikat, tetapi harus diawali dengan tanda tanya dan tidak mengandung spasi.

Variabel `SELECT` yang digunakan untuk menampilkan pengetahuan yang tersimpan dalam ontologi yaitu:

```
SELECT ?objective ?methods ?datasources ?databases
?filesystem ?processing.
```

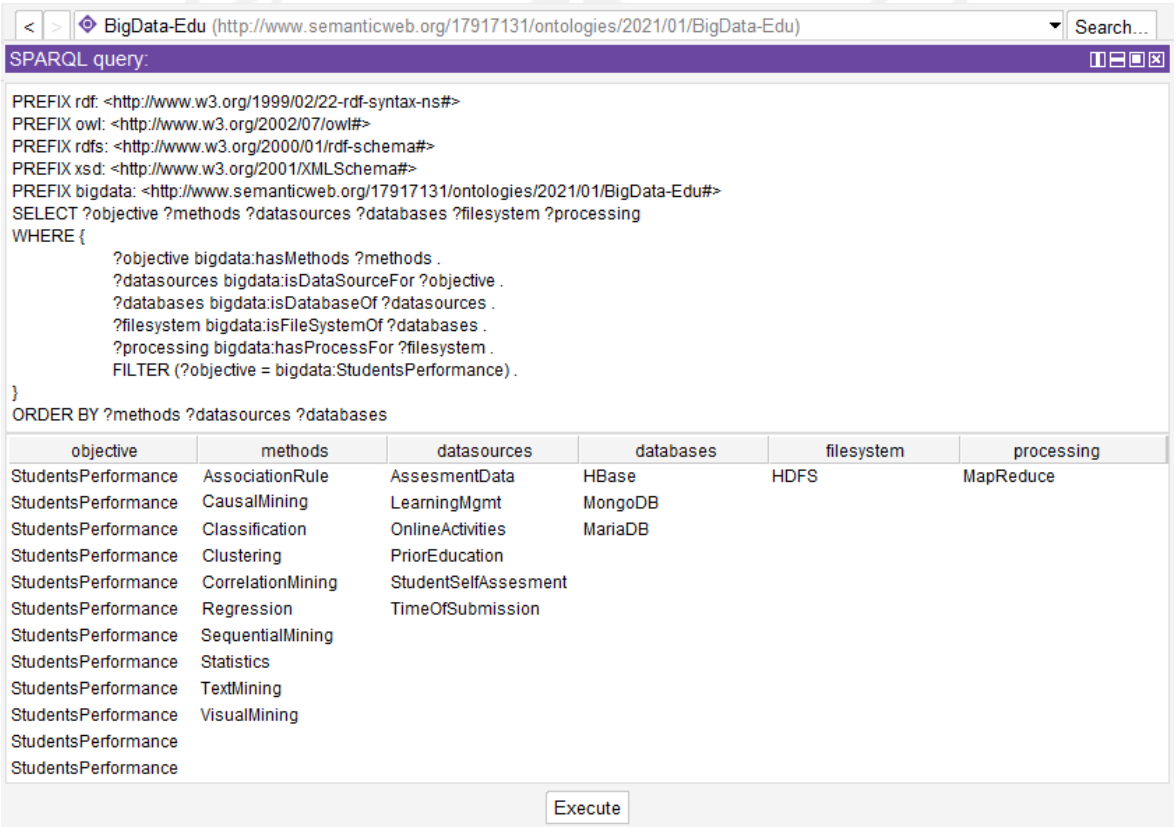
Variabel `?objective` digunakan untuk menampilkan objek pengetahuan mengenai cara untuk menganalisis kinerja mahasiswa. Variabel `?methods` untuk menampilkan metode apa saja yang dapat digunakan untuk mencapai tujuan analisis tersebut. Variabel `?datasources` untuk menampilkan sumber data apa saja yang bisa digunakan untuk mencapai tujuan tersebut. Variabel `?databases` untuk menunjukkan teknologi basis data yang dapat digunakan untuk menampung data tersebut. Variabel `?filesystem` untuk menampilkan sistem *file* yang bisa digunakan untuk menyimpan *big data*. Kemudian variabel `?processing` untuk mengetahui teknik apa yang dapat digunakan untuk memproses *big data* tersebut.

Setelah menuliskan klausa `SELECT` beserta isinya, dilanjutkan dengan menuliskan isi dari klausa `WHERE`. Pada klausa `WHERE`, harus diisi berdasarkan pola *tripple*. Pola *tripple* merupakan pernyataan yang terdiri dari subjek, predikat dan objek. Klausa `WHERE` yang digunakan untuk menarik pengetahuan dari ontologi sesuai pertanyaan yang diberikan, yaitu:

```
WHERE {
    ?objective bigdata:hasMethods ?methods .
    ?datasources bigdata:isDataSourceFor ?objective .
    ?databases bigdata:isDatabaseOf ?datasources .
    ?filesystem bigdata:isFileSystemOf ?databases .
    ?processing bigdata:hasProcessFor ?filesystem .
    FILTER (?objective = bigdata:StudentsPerformance) .
}
```

Pernyataan `bigdata:hasMethods` merupakan predikat yang menghubungkan `?objective` dengan `?methods`. Ini berdasarkan properti objek yang telah ditetapkan dalam ontologi. Properti tersebut menghubungkan kelas `Objective` sebagai *domain* dengan kelas `Methods` sebagai *range*. Pernyataan `bigdata:isDataSourceFor` merupakan predikat yang menghubungkan `?datasources` dengan `?objective`. ini berdasarkan properti yang menghubungkan kelas `datasources` sebagai *domain* dengan kelas `objective` sebagai *range*.

Setelah menulis semua pernyataan dari klausa yang akan dipakai melakukan query, dilakukan pengecekan pernyataan dari *query* tersebut. Setelah memastikan pernyataan *query* tersebut sudah benar, maka dapat dilakukan eksekusi *query*. Hasil *query* SPARQL dapat dilihat pada Gambar 4.11.

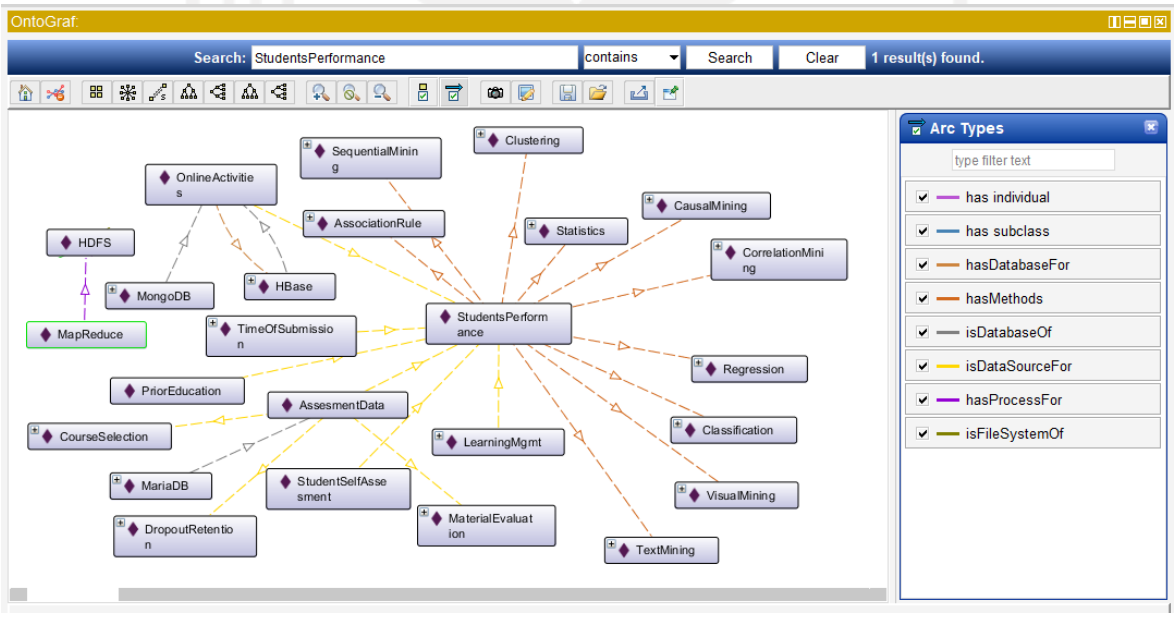



Gambar 4.11 Uji Coba *Query* `StudentsPerformance`

Berdasarkan hasil uji coba query seperti yang telah ditunjukkan pada Gambar 4.11, diketahui bahwa untuk menganalisis kinerja akademik mahasiswa, dapat dilakukan analisis melalui beberapa pilihan metode seperti yang ditampilkan dari hasil *query*, yaitu: 1) *association rule*, 2) *causal mining*, 3) klasifikasi *(classification)*, 4) pengelompokkan *(clustering)*, 5) penggalian korelasi *(correlation mining)*, 6) regresi *(regression)*, 7) *sequential mining*, 8) statistika, 9) *text mining*, dan 10) *visual mining*.

Beberapa sumber data dapat digunakan yaitu: 1) data penilaian akademik, 2) data dari LMS (*learning management system*), 3) aktivitas daring, 4) pendidikan sebelumnya, 5) penilaian diri sendiri, dan 6) ketepatan waktu mengumpulkan tugas. Sumber-sumber data tersebut dapat digunakan sebagai sumber data untuk menganalisis kinerja mahasiswa. Sumber data tersebut dapat disimpan menggunakan beberapa pilihan teknologi basis data, yaitu MongoDB yang lebih sesuai untuk data tidak terstruktur seperti kumpulan data dari LMS dan aktivitas daring mahasiswa. Untuk data terstruktur dapat menggunakan MySQL, seperti data dari penilaian akademik, pendidikan sebelumnya. Sedangkan dalam menyimpan *big data* dapat menggunakan teknologi *filesystem* dari Hadoop yaitu HDFS, yang dapat diproses menggunakan teknik MapReduce.

Untuk memvalidasi pengetahuan yang ditarik melalui *query* SPARQL sebelumnya, perlu dilakukan peninjauan kembali terhadap ontologi yang telah dibangun. Objek `StudentsPerformance` mempunyai beberapa metode yang dapat digunakan untuk mencapai sasaran yang diinginkan. Pengetahuan yang terhubung dengan objek tersebut dapat dilihat pada Gambar 4.12. Berdasarkan pengetahuan yang tersimpan dalam ontologi, dapat dikatakan bahwa hasil *query* yang telah dilakukan telah sesuai dengan pengetahuan yang terdapat dalam ontologi.




Gambar 4.12 Relasi dari `StudentsPerformance`

Berdasarkan uji coba *query* yang telah dilakukan, ontologi yang telah dibangun mampu memberikan pengetahuan yang tersimpan dalam ontologi. Pengetahuan yang telah direpresentasikan ke dalam bentuk model ontologi dapat dimanfaatkan oleh perguruan tinggi guna mendukung kesiapan institusi dalam menghadapi era *big data*.

# BAB 5
# Kesimpulan dan Saran

## 5.1 Kesimpulan

Penelitian ini melakukan pengembangan model ontologi untuk merepresentasikan pengetahuan mengenai *big data analytics* di perguruan tinggi berdasarkan aspek *analytics, data sources,* dan *platforms*. Sehingga berdasarkan hasil penelitian yang telah dilakukan dapat disimpulkan bahwa:

1. Domain pengetahuan *big data analytics* di perguruan tinggi mampu dimodelkan menggunakan teknik ontologi berdasarkan tahapan konseptualisasi, implementasi, dan evaluasi. Hasil pemodelan mampu merepresentasikan pengetahuan ke dalam bahasa OWL/RDF sebagai standar bahasa yang dapat dimengerti oleh komputer.
2. Model ontologi dapat menyediakan pengetahuan terstruktur yang dapat digunakan dalam memahami potensi pemanfaatan *big data analytics* di bidang pendidikan khususnya perguruan tinggi.

Evaluasi model ontologi dilakukan dengan cara menganalisis kualitas ontologi menggunakan OntoQA dan uji coba *query* menggunakan SPARQL. Berdasarkan hasil analisis kualitas ontologi diperoleh nilai RR (*Relationship Richness*) sebesar 0,62 yang menunjukkan bahwa keberagaman pengetahuan yang dimuat ontologi ini sudah kaya. Nilai IR (*Inheritance Richness*) sebesar 1,78 yang menunjukkan bahwa menunjukkan bahwa ontologi yang telah dibangun termasuk ontologi yang cukup spesifik. Namun, nilai AR (Attribute Richness) sebesar 0,06 menunjukkan bahwa informasi yang dimuat dalam ontologi masih dapat ditingkatkan lagi. Melalui analisis hasil uji coba *query*, ontologi dinilai mampu menunjukkan pengetahuan yang tersimpan di dalamnya dengan akurat berdasarkan validasi yang menyesuaikan hasil *query* dengan ontologi. Model ontologi tersebut dapat dimanfaatkan guna membangun sistem berbasis pengetahuan maupun sistem manajemen pengetahuan agar dapat mendukung kesiapan perguruan tinggi dalam menghadapi dan memanfaatkan *big data*.

## 5.2 Saran

Dalam penelitian ini disadari masih banyak kekurangan yang sebaiknya dapat disempurnakan lagi oleh penelitian berikutnya. Adapun saran pada penelitian ini yaitu:

1. Pengetahuan yang tersimpan dapat dikembangkan menyesuaikan perkembangan pengetahuan yang dibutuhkan berdasarkan konseptualisasi yang telah dilakukan pada penelitian ini.
2. Informasi yang dimuat dalam ontologi dapat ditingkatkan lagi. Informasi tersebut dapat diperoleh dari berbagai studi/kajian mengenai pemanfaatan *big data analytics* di perguruan tinggi.
3. Penyesuaian perkembangan pengetahuan dan pengumpulan informasi dapat dilakukan secara otomatis menggunakan metode ekstraksi informasi terhadap berbagai studi relevan yang kemudian dapat disimpan pada ontologi ini.
4. Penelitian ini hanya membangun model ontologi. Penelitian selanjutnya dapat membangun sistem berbasis pengetahuan yang menggunakan ontologi ini, maupun melakukan integrasi dengan ontologi yang sudah ada agar dapat meningkatkan interoperabilitas pengetahuan.

# Daftar Pustaka


Abubakar, A. M., Elrehail, H., Alatailat, M. A., & Elçi, A. (2019). Knowledge management, decision-making style and organizational performance. *Journal of Innovation & Knowledge*, *4*(2), 104-114.

Aldowah, H., Al-Samarraie, H., & Fauzy, W. M. (2019). Educational data mining and learning analytics for 21st century higher education: A review and synthesis. *Telematics and Informatics*, *37*, 13-49.

Alsheikh, N. A. (2019). Developing an Integrated Framework to Utilize Big Data for Higher Education Institutions in Saudi Arabia. *International Journal of Computer Science & Information Technology (IJCSIT) Vol*, *11*.

Ang, K. L. M., Ge, F. L., & Seng, K. P. (2020). Big educational data & analytics: Survey, architecture and challenges. *IEEE access*, *8*, 116392-116414.

Antoniou, G., & Van Harmelen, F. (2004). Web ontology language: Owl. In *Handbook on ontologies* (pp. 67-92). Springer, Berlin, Heidelberg.

Anuradha, J. (2015). A brief introduction on Big Data 5Vs characteristics and Hadoop technology. *Procedia computer science*, *48*, 319-324.

Ashaari, M. A., Amran, A., Ahmad, N. H., Bakri, H., & Nazri, S. (2020, June). Big Data Analytics Technology Capability and Data-Driven Decision Making in Malaysian Higher Education Institutions: A Conceptual Framework. In *IOP Conference Series: Materials Science and Engineering* (Vol. 874, No. 1, p. 012021). IOP Publishing.

Attaran, M., Stark, J., & Stotler, D. (2018). Opportunities and challenges for big data analytics in US higher education: A conceptual model for implementation. *Industry and Higher Education*, *32*(3), 169-182.

Badron, Y. F., Agus, F., & Hatta, H. R. (2017, Maret). Studi Tentang Pemodelan Ontologi Web Semantik Dan Prospek Penerapan Pada Bibliografi Artikel Jurnal Ilmiah. Dalam *Prosiding SAKTI (Seminar Ilmu Komputer dan Teknologi Informasi)* (Vol. 2, No. 1, pp. 164-169).

Bai, X., Zhang, F., Li, J., Guo, T., Aziz, A., Jin, A., & Xia, F. (2021). Educational Big Data: Predictions, Applications and Challenges. *Big Data Research*, 100270.

Baig, M. I., Shuib, L., & Yadegaridehkordi, E. (2020). Big data in education: a state of the art, limitations, and future research directions. *International Journal of Educational Technology in Higher Education*, *17*(1), 1-23.



Baig, M. I., Shuib, L., & Yadegaridehkordi, E. (2020). Big data in education: a state of the art, limitations, and future research directions. *International Journal of Educational Technology in Higher Education*, *17*(1), 1-23.



Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific american*, *284*(5), 34-43.

Cen, L., Ruta, D., & Ng, J. (2015, July). Big education: Opportunities for big data analytics. In *2015 IEEE international conference on digital signal processing (DSP)* (pp. 502-506). IEEE.

Chaurasia, S. S., & Rosin, A. F. (2017). From Big Data to Big Impact: analytics for teaching and learning in higher education. *Industrial and Commercial Training*.

Da Silva, A. R. (2015). Model-driven engineering: A survey supported by the unified conceptual model. *Computer Languages, Systems & Structures, 43*, 139-155.

Daniel, B. K. (2017). Big data in higher education: The big picture. In *Big data and learning analytics in higher education* (pp. 19-28). Springer, Cham.

De Long, D. (1997). Building the knowledge-based organization: How culture drives knowledge behaviors. *Centers for Business Innovation–Working Paper*, 1-29.

De Silva, M., Howells, J., & Meyer, M. (2018). Innovation intermediaries and collaboration: Knowledge–based practices and internal value creation. *Research Policy*, *47*(1), 70-87.

Deshpande, P. S., Sharma, S. C., & Peddoju, S. K. (2019). Predictive and prescriptive analytics in Big-data Era. In *Security and data storage aspect in cloud computing* (pp. 71-81). Springer, Singapore.

Devedžić, V. (1999). A survey of modern knowledge modeling techniques. *Expert systems with applications*, *17*(4), 275-294.

Devedzic, V. (2001). Knowledge modeling–state of the art. *Integrated Computer-Aided Engineering*, *8*(3), 257-281.

El-Diraby, T. E. (2013). Domain ontology for construction knowledge. *Journal of Construction Engineering and Management*, *139*(7), 768-784.

Fischer, C., Pardos, Z. A., Baker, R. S., Williams, J. J., Smyth, P., Yu, R., ... & Warschauer, M. (2020). Mining big data in education: Affordances and challenges. *Review of Research in Education*, *44*(1), 130-160.

Fudholi, D. H., Rahayu, W., & Pardede, E. (2016, March). Ontology-based information extraction for knowledge enrichment and validation. In 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA) (pp. 1116-1123). IEEE.

Fudholi, D. H., Rahayu, W., & Pardede, E. (2016, March). Ontology-based information extraction for knowledge enrichment and validation. In 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA) (pp. 1116-1123). IEEE.

Fung, K. W., & Bodenreider, O. (2019). Knowledge representation and ontologies. In Clinical research informatics (pp. 313-339). Springer, Cham.

Gangemi, A., Catenacci, C., Ciaramita, M., & Lehmann, J. (2006, June). Modelling ontology evaluation and validation. In European Semantic Web Conference (pp. 140-154). Springer, Berlin, Heidelberg.

Gartner. 2019. Gartner's Four Analytics Types. https://blogs.gartner.com/jason-mcnellis/2019/11/05/youre-likely-investing-lot-marketing-analytics-getting-right-insights (Diakses pada Juni 2021)

Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge acquisition, 5(2), 199-220.

He, Y., Hao, C., Wang, Y., Li, Y., Wang, Y., Huang, L., & Tian, X. (2020). An ontology-based method of knowledge modelling for remanufacturing process planning. Journal of Cleaner Production, 258, 120952.

Hung, R. Y. Y., Lien, B. Y. H., Fang, S. C., & McLean, G. N. (2010). Knowledge as a facilitator for enhancing innovation performance through total quality management. *Total Quality Management*, *21*(4), 425-438.

Kollia, I., & Glimm, B. (2013). Optimizing SPARQL query answering over OWL ontologies. Journal of artificial intelligence research, 48, 253-303.

Kurilovas, E. (2018). On data-driven decision-making for quality education. Computers in Human Behavior.

Lee, I. (2017). Big data: Dimensions, evolution, impacts, and challenges. Business Horizons, 60(3), 293-303.

Mandinach, E., & Gummer, E. (2015). Data-driven decision making: Components of the enculturation of data use in education. Teachers College Record, 117(4), 1-12.

Marín-Marín, J. A., López-Belmonte, J., Fernández-Campoy, J. M., & Romero-Rodríguez, J. M. (2019). Big data in education. A bibliometric review. *Social Sciences*, *8*(8), 223.

McGuinness, D. L., & Van Harmelen, F. (2004). OWL web ontology language overview. W3C recommendation, 10(10), 2004.

Noy, N. F., & McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology.

Oussous, A., Benjelloun, F. Z., Lahcen, A. A., & Belfkih, S. (2018). Big Data technologies: A survey. *Journal of King Saud University-Computer and Information Sciences*, *30*(4), 431-448.

Punlumjeak, W., Rachburee, N., & Arunrerk, J. (2017). Big data analytics: Student performance prediction using feature selection and machine learning on microsoft azure platform. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, *9*(1-4), 113-117.

Quadir, B., Chen, N. S., & Isaias, P. (2020). Analyzing the educational goals, problems and techniques used in educational big data research from 2010 to 2018. Interactive Learning Environments, 1-17.

Rozas, I., Rolliawati, D., Wahyudi, N., & Ratodi, M. (2019, November). Readiness of Indonesian Higher Education Programs in the Big Data Era. In *International Conference on Social Science 2019 (ICSS 2019)* (pp. 1216-1221). Atlantis Press.

Saggi, M. K., & Jain, S. (2018). A survey towards an integration of big data analytics to big insights for value-creation. *Information Processing & Management*, *54*(5), 758-790.

Schneider, M., Carroll, J., Herman, I., & Patel-Schneider, P. F. (2009). OWL 2 web ontology language RDF-based semantics. W3C Recommendation (October 27 2009).

Seufert, S., Meier, C., Soellner, M., & Rietsche, R. (2019). A pedagogical perspective on big data and learning analytics: A conceptual model for digital learning support. *Technology, Knowledge and Learning*, *24*(4), 599-619.

Shi-Nash, A., & Hardoon, D. R. (2017). Data analytics and predictive analytics in the era of big data. Internet of things and data analytics handbook, 329-345.

Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: Principles and methods. Data & knowledge engineering, 25(1-2), 161-197.

Tartir, S., & Arpinar, I. B. (2007, September). Ontology evaluation and ranking using OntoQA. In *International conference on semantic computing (ICSC 2007)* (pp. 185-192). IEEE.

Tsai, C. W., Lai, C. F., Chao, H. C., & Vasilakos, A. V. (2015). Big data analytics: a survey. Journal of Big data, 2(1), 1-32

Tulasi, B. (2013). Significance of Big Data and analytics in higher education. International Journal of Computer Applications, 68(14).

Waheed, H., Hassan, S. U., Aljohani, N. R., Hardman, J., Alelyani, S., & Nawaz, R. (2020). Predicting academic performance of students from VLE big data using deep learning models. *Computers in Human behavior*, *104*, 106189.

Wang, Y. (2016). Big opportunities and big concerns of big data in education. TechTrends, 60(4), 381-384.

World Wide Web Consortium (W3C). 2003. Semantic Web Stack. https://www.w3.org/2003/Talks/simo-semwebapp/all.htm (Diakses pada Juni 2021)

Yun, W., Zhang, X., Li, Z., Liu, H., & Han, M. (2021). Knowledge modeling: A survey of processes and techniques. International Journal of Intelligent Systems, 36(4), 1686-1720.

Zagzebski, L. (2017). What is knowledge? The Blackwell guide to epistemology, 92-116.

# LAMPIRAN

Tabel 1. Kode OWL Pembentukan Kelas

class.owl

```
<!--Class Declaration-->
<Declaration> <Class BigData-Edu="#Analytics"/> </Declaration>
<Declaration> <Class BigData-Edu="#Objective"/> </Declaration>
<Declaration> <Class BigData-Edu="#Behavior"/> </Declaration>
<Declaration> <Class BigData-Edu="#Learning"/> </Declaration>
<Declaration> <Class BigData-Edu="#Predictive"/> </Declaration>
<Declaration> <Class BigData-Edu="#Methods"/> </Declaration>
<Declaration> <Class BigData-Edu="#DataSource"/> </Declaration>
<Declaration> <Class BigData-Edu="#SemiStructured"/> </Declaration>
<Declaration> <Class BigData-Edu="#Structured"/> </Declaration>
<Declaration> <Class BigData-Edu="#Unstructured"/> </Declaration>
<Declaration> <Class BigData-Edu="#Platforms"/> </Declaration>
<Declaration> <Class BigData-Edu="#Management"/> </Declaration>
<Declaration> <Class BigData-Edu="#Coordination"/> </Declaration>
<Declaration> <Class BigData-Edu="#Deployment"/> </Declaration>
<Declaration> <Class BigData-Edu="#Scheduler"/> </Declaration>
<Declaration> <Class BigData-Edu="#Processing"/> </Declaration>
<Declaration> <Class BigData-Edu="#Ingestion"/> </Declaration>
<Declaration> <Class BigData-Edu="#MapReduce"/> </Declaration>
<Declaration> <Class BigData-Edu="#Querying"/> </Declaration>
<Declaration> <Class BigData-Edu="#Resource"/> </Declaration>
<Declaration> <Class BigData-Edu="#Storage"/> </Declaration>
<Declaration> <Class BigData-Edu="#FileSystem"/> </Declaration>
<Declaration> <Class BigData-Edu="#Databases"/> </Declaration>
<Declaration> <Class BigData-Edu="#SQL-based"/> </Declaration>
<Declaration> <Class BigData-Edu="#NoSQL-based"/> </Declaration>
<!--Subclass Declaration-->
<SubClassOf> <Class BigData-Edu="#SemiStructured"/> <Class BigData-
Edu="#DataSource"/> </SubClassOf> <SubClassOf> <Class BigData-
Edu="#Structured"/> <Class BigData-Edu="#DataSource"/> </SubClassOf>
<SubClassOf> <Class BigData-Edu="#Unstructured"/> <Class BigData-
Edu="#DataSource"/> </SubClassOf> <SubClassOf> <Class BigData-
Edu="#Objective"/> <Class BigData-Edu="#Analytics"/> </SubClassOf>
<SubClassOf> <Class BigData-Edu="#Methods"/> <Class BigData-
Edu="#Analytics"/> </SubClassOf>          <SubClassOf> <Class BigData-
Edu="#Behavior"/> <Class BigData-Edu="#Objective"/> </SubClassOf>
<SubClassOf> <Class BigData-Edu="#Learning"/> <Class BigData-
Edu="#Objective"/>   </SubClassOf> <SubClassOf> <Class BigData-
Edu="#Predictive"/> <Class BigData-Edu="#Objective"/> </SubClassOf>
<SubClassOf> <Class BigData-Edu="#Management"/> <Class BigData-
Edu="#Platforms"/> </SubClassOf> <SubClassOf> <Class BigData-
Edu="#Processing"/> <Class BigData-Edu="#Platforms"/> </SubClassOf>
<SubClassOf> <Class BigData-Edu="#Storage"/> <Class BigData-
Edu="#Platforms"/> </SubClassOf> <SubClassOf> <Class BigData-
Edu="#Coordination"/> <Class BigData-Edu="#Management"/> </SubClassOf>
<SubClassOf> <Class BigData-Edu="#Deployment"/> <Class BigData-
Edu="#Management"/> </SubClassOf> <SubClassOf> <Class BigData-
Edu="#Scheduler"/> <Class BigData-Edu="#Management"/> </SubClassOf>
<SubClassOf> <Class BigData-Edu="#Ingestion"/> <Class BigData-
Edu="#Processing"/> </SubClassOf> <SubClassOf> <Class BigData-
Edu="#MapReduce"/> <Class BigData-Edu="#Processing"/> </SubClassOf>
<SubClassOf> <Class BigData-Edu="#Querying"/> <Class BigData-
Edu="#Processing"/> </SubClassOf> <SubClassOf> <Class BigData-
```

class.owl

```
Edu="#Resource"/> <Class BigData-Edu="#Processing"/> </SubClassOf>
<SubClassOf> <Class BigData-Edu="#FileSystem"/> <Class BigData-
Edu="#Storage"/> </SubClassOf> <SubClassOf> <Class BigData-
Edu="#Databases"/> <Class BigData-Edu="#Storage"/> </SubClassOf>
<SubClassOf> <Class BigData-Edu="#SQL-based"/> <Class BigData-
Edu="#Databases"/> </SubClassOf> <SubClassOf> <Class BigData-
Edu="#NoSQL-based"/> <Class BigData-Edu="#Databases"/> </SubClassOf>
```

Tabel 2. Kode OWL Pembentukan Properti

property.owl

```
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#ActionModeling"/> <NamedIndividual
IRI="#AssociationRule"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#ActionModeling"/> <NamedIndividual
IRI="#CausalMining"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#ActionModeling"/> <NamedIndividual
IRI="#Classification"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#ActionModeling"/> <NamedIndividual
IRI="#Clustering"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#ActionModeling"/> <NamedIndividual
IRI="#CorrelationMining"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#ActionModeling"/> <NamedIndividual
IRI="#OutlierDetection"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#ActionModeling"/> <NamedIndividual
IRI="#Regression"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#ActionModeling"/> <NamedIndividual
IRI="#SequentialMining"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#ActionModeling"/> <NamedIndividual
IRI="#Statistics"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#ActionModeling"/> <NamedIndividual
IRI="#VisualMining"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#CollaborativeLearning"/> <NamedIndividual
IRI="#AssociationRule"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#CollaborativeLearning"/> <NamedIndividual
IRI="#Classification"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#CollaborativeLearning"/> <NamedIndividual
IRI="#Clustering"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#CollaborativeLearning"/> <NamedIndividual
IRI="#CorrelationMining"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#CollaborativeLearning"/> <NamedIndividual
IRI="#Regression"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
```

property.owl

```
<NamedIndividual IRI="#CollaborativeLearning"/> <NamedIndividual
IRI="#SequentialMining"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#CollaborativeLearning"/> <NamedIndividual
IRI="#Statistics"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#CollaborativeLearning"/> <NamedIndividual
IRI="#TextMining"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#CollaborativeLearning"/> <NamedIndividual
IRI="#VisualMining"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#DropoutRetention"/> <NamedIndividual
IRI="#Classification"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#DropoutRetention"/> <NamedIndividual
IRI="#Clustering"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#DropoutRetention"/> <NamedIndividual
IRI="#Regression"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#DropoutRetention"/> <NamedIndividual
IRI="#Statistics"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#KnowledgeModeling"/> <NamedIndividual
IRI="#AssociationRule"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#KnowledgeModeling"/> <NamedIndividual
IRI="#CausalMining"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#KnowledgeModeling"/> <NamedIndividual
IRI="#Classification"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#KnowledgeModeling"/> <NamedIndividual
IRI="#Clustering"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#KnowledgeModeling"/> <NamedIndividual
IRI="#CorrelationMining"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#KnowledgeModeling"/> <NamedIndividual
IRI="#OutlierDetection"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#KnowledgeModeling"/> <NamedIndividual
IRI="#Regression"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#KnowledgeModeling"/> <NamedIndividual
IRI="#SequentialMining"/> </ObjectPropertyAssertion>
<ObjectPropertyAssertion> <ObjectProperty IRI="#hasMethods"/>
<NamedIndividual IRI="#KnowledgeModeling"/> <NamedIndividual
IRI="#Statistics"/> </ObjectPropertyAssertion>
```

Tabel 1. Kode OWL Pembentukan Objek

object.owl

```
<ClassAssertion> <Class IRI="#Behavior"/> <NamedIndividual
IRI="#ActionModeling"/> </ClassAssertion> <ClassAssertion> <Class
IRI="#Structured"/> <NamedIndividual IRI="#AdmissionData"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Deployment"/>
```

object.owl

```
<NamedIndividual IRI="#Ambari"/> </ClassAssertion> <ClassAssertion>
<Class IRI="#Structured"/> <NamedIndividual IRI="#AssesmentData"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Methods"/>
<NamedIndividual IRI="#AssociationRule"/> </ClassAssertion>
<ClassAssertion> <Class IRI="#MapReduce"/> <NamedIndividual
IRI="#Cascading"/> </ClassAssertion> <ClassAssertion> <Class
IRI="#Methods"/> <NamedIndividual IRI="#CausalMining"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Ingestion"/>
<NamedIndividual IRI="#Chukwa"/> </ClassAssertion> <ClassAssertion>
<Class IRI="#Methods"/> <NamedIndividual IRI="#Classification"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Methods"/>
<NamedIndividual IRI="#Clustering"/> </ClassAssertion> <ClassAssertion>
<Class IRI="#Learning"/> <NamedIndividual
IRI="#CollaborativeLearning"/> </ClassAssertion> <ClassAssertion>
<Class IRI="#Methods"/> <NamedIndividual IRI="#CorrelationMining"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Predictive"/>
<NamedIndividual IRI="#CourseSelection"/> </ClassAssertion>
<ClassAssertion> <Class IRI="#Structured"/> <NamedIndividual
IRI="#DemographicInformation"/> </ClassAssertion> <ClassAssertion>
<Class IRI="#Methods"/> <NamedIndividual IRI="#DensityEstimation"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Predictive"/>
<NamedIndividual IRI="#DropoutRetention"/> </ClassAssertion>
<ClassAssertion> <Class IRI="#Structured"/> <NamedIndividual
IRI="#EnrollmentData"/> </ClassAssertion> <ClassAssertion> <Class
IRI="#Ingestion"/> <NamedIndividual IRI="#Flume"/> </ClassAssertion>
<ClassAssertion> <Class IRI="#NoSQL-based"/> <NamedIndividual
IRI="#HBase"/> </ClassAssertion> <ClassAssertion> <Class
IRI="#FileSystem"/> <NamedIndividual IRI="#HDFS"/> </ClassAssertion>
<ClassAssertion> <Class IRI="#Querying"/> <NamedIndividual
IRI="#Hive"/> </ClassAssertion> <ClassAssertion> <Class
IRI="#Behavior"/> <NamedIndividual IRI="#KnowledgeModeling"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Learning"/>
<NamedIndividual IRI="#LearningAssesment"/> </ClassAssertion>
<ClassAssertion> <Class IRI="#Unstructured"/> <NamedIndividual
IRI="#LearningMgmt"/> </ClassAssertion> <ClassAssertion> <Class
IRI="#Predictive"/> <NamedIndividual IRI="#MaterialEvaluation"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#NoSQL-based"/>
<NamedIndividual IRI="#MongoDB"/> </ClassAssertion> <ClassAssertion>
<Class IRI="#SQL-based"/> <NamedIndividual IRI="#MySQL"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#NoSQL-based"/>
<NamedIndividual IRI="#Neo4j"/> </ClassAssertion> <ClassAssertion>
<Class IRI="#Unstructured"/> <NamedIndividual IRI="#OnlineActivities"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Scheduler"/>
<NamedIndividual IRI="#Oozie"/> </ClassAssertion> <ClassAssertion>
<Class IRI="#SQL-based"/> <NamedIndividual IRI="#Oracle"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Methods"/>
<NamedIndividual IRI="#OutlierDetection"/> </ClassAssertion>
<ClassAssertion> <Class IRI="#Behavior"/> <NamedIndividual
IRI="#PatternModeling"/> </ClassAssertion> <ClassAssertion> <Class
IRI="#Learning"/> <NamedIndividual IRI="#PersonalizedLearning"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Querying"/>
<NamedIndividual IRI="#Pig"/> </ClassAssertion> <ClassAssertion> <Class
IRI="#Structured"/> <NamedIndividual IRI="#PriorEducation"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#NoSQL-based"/>
<NamedIndividual IRI="#Redis"/> </ClassAssertion> <ClassAssertion>
<Class IRI="#Methods"/> <NamedIndividual IRI="#Regression"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Methods"/>
<NamedIndividual IRI="#SequentialMining"/> </ClassAssertion>
<ClassAssertion> <Class IRI="#Learning"/> <NamedIndividual
```

object.owl

```
IRI="#SocialLearning"/> </ClassAssertion> <ClassAssertion> <Class
IRI="#SemiStructured"/> <NamedIndividual IRI="#SocialMediaData"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Ingestion"/>
<NamedIndividual IRI="#Sqoop"/> </ClassAssertion> <ClassAssertion>
<Class IRI="#Methods"/> <NamedIndividual IRI="#Statistics"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Structured"/>
<NamedIndividual IRI="#StudentSelfAssesment"/> </ClassAssertion>
<ClassAssertion> <Class IRI="#Predictive"/> <NamedIndividual
IRI="#StudentsPerformance"/> </ClassAssertion> <ClassAssertion> <Class
IRI="#Structured"/> <NamedIndividual IRI="#SurveyData"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Methods"/>
<NamedIndividual IRI="#TextMining"/> </ClassAssertion> <ClassAssertion>
<Class IRI="#Structured"/> <NamedIndividual IRI="#TimeOfSubmission"/>
</ClassAssertion> <ClassAssertion> <Class IRI="#Methods"/>
<NamedIndividual IRI="#VisualMining"/> </ClassAssertion>
<ClassAssertion> <Class IRI="#Resource"/> <NamedIndividual
IRI="#YARN"/> </ClassAssertion> <ClassAssertion> <Class
IRI="#Coordination"/> <NamedIndividual IRI="#Zookeper"/>
</ClassAssertion>
```