

BAB 2 TINJAUAN PUSTAKA

2.1. Pembelajaran Online (*E-learning*)

E-learning adalah pembelajaran yang memungkinkan atau didukung oleh penggunaan alat dan konten digital. *E-learning* biasanya melibatkan beberapa bentuk interaktivitas, termasuk di dalamnya dapat berupa interaksi online antara peserta didik dan pengajar atau kelompok mereka. *E-learning* biasanya diakses melalui internet, namun teknologi lainnya seperti CD-ROM juga dapat digunakan [4]. Walaupun demikian, penggunaan istilah pembelajaran online lebih identik dengan pembelajaran melalui internet atau jaringan.

Mengapa pembelajaran online perlu dilakukan? Untuk membahas hal tersebut lebih dalam, penulis akan memaparkan beberapa butir studi literatur yang perlu diketahui; yaitu aspek personal pembelajar, personalisasi pembelajaran, dan kelebihan pembelajaran online.

2.1.1. Aspek Personal Pembelajar

Proses belajar-mengajar menurut Huitt [26] memiliki berbagai komponen yang berpengaruh, seperti komunitas (ukuran dan wilayah); keluarga (tingkat pendidikan ibu, pendapatan keluarga –buku-buku yang ada di rumah, serta ekspektasi akademik); pengajar yang terdiri dari aspek karakteristik pengajar dan aktivitas/tingkah laku pengajar; pembelajar yang terdiri dari dari aspek karakteristik pembelajar dan aktivitas/tingkah laku pembelajar; kebijakan sekolah; serta kebijakan negara. Gaya belajar dan *prior knowledge* merupakan bagian dari karakteristik pembelajar, selain kecerdasan dan motivasi yang akan mempengaruhi aktivitas pembelajar dalam proses belajar-mengajar [5].

2.1.1.1. Gaya Belajar dalam Pembelajaran

Proses belajar mengajar memiliki sejumlah tujuan yang ingin dicapai [6]. Berdasarkan tujuan-tujuan tersebut, semua kegiatan dirancang untuk memfasilitasi peserta didik dalam memperoleh pengetahuan dan keahlian. Dalam proses tersebut seringkali diasumsikan bahwa peserta didik memiliki gaya belajar (*learning style*) yang sama. Padahal dalam realitasnya tidaklah selalu demikian. Pada beberapa penelitian yang telah dilakukan sebelumnya, terungkap bahwa pengabaian terhadap gaya belajar peserta didik yang berbeda tersebut justru tidak memfasilitasi peserta didik dalam mengoptimalkan potensi yang dimiliki dalam belajar.

Gaya belajar merupakan pendekatan dalam pembelajaran yang menekankan fakta bahwa individu-individu menerima dan memproses informasi dengan cara-cara yang sangat berbeda [7]. Setidaknya terdapat empat model gaya belajar yang telah dikembangkan saat ini, antara lain [8]:

- a. The Myers-Briggs Type Indicator (MBTI),
- b. Kolb's Learning Style Model,
- c. Herrmann Brain Dominance Instrument (HBDI), dan
- d. Felder-Silverman Learning Style Model.

Berikut ini akan dijelaskan mengenai Felder-Silverman Learning Style Model yang akan digunakan pada penelitian ini. Model lainnya tidak dijelaskan karena tidak digunakan dan beberapa penjelasan akan beririsan dengan model Felder-Silverman.

Felder-Silverman Learning Style Model [11] membagi gaya belajar berdasarkan lima dimensi. Gaya belajar tersebut terdiri dari gaya belajar *active-reflective*, gaya belajar *inductive-deductive*, gaya belajar *visual-verbal*, gaya belajar *sensing-intuitive*, dan gaya belajar *sequential-global*.

- a. Gaya Belajar *Active-Reflective* (Dimensi Pemrosesan)

Pembelajar dengan gaya belajar *active* cenderung untuk memelihara dan memahami informasi yang terbaik dengan melakukan keaktifan dengannya –

membahas, menerapkannya, atau menjelaskannya kepada orang lain. Di lain sisi, pembelajar *reflective* memilih untuk memikirkannya dahulu dengan tenang. Sebuah ungkapan dari pembelajar *active* adalah, “Mari mencobanya dan lihat bagaimana ia bekerja”, sedangkan pembelajar *reflective* berkata, “Mari kita pikirkan terlebih dahulu”. Pembelajar *active* lebih menyukai belajar kelompok, sebaliknya pembelajar *reflective* lebih menyukai belajar sendiri. Mendengarkan ceramah tanpa melakukan kegiatan fisik kecuali menulis catatan sulit dilakukan keduanya, namun khususnya bagi pembelajar *active*.

b. Gaya Belajar *Sensing-Intuitive* (Dimensi Persepsi)

Pembelajar bertipe *sensing* cenderung menyukai mempelajari fakta, sedangkan pembelajar bertipe *intuitive* sering kali lebih memilih menemukan kemungkinan dan hubungan. *Sensor* lebih mudah memahami jika diberikan contoh dari konsep dan prosedur dan menemukan bagaimana konsep diaplikasikan pada praktik; sedangkan pembelajar *intuitive* mencoba memahami suatu interpretasi atau teori yang menghubungkan fakta. Oleh karena itu, *sensor* cocok diberi pengajaran dengan aplikasi nyata, sedangkan pembelajar *intuitive* cocok diberikan peta konsep.

c. Gaya Belajar *Visual-Verbal* (Dimensi Input)

Pembelajar *visual* mengingat apa yang dilihatnya dengan baik, seperti gambar, diagram, *time line*, film, atau demonstrasi; sedangkan pembelajar *verbal* lebih banyak mendapat informasi dari kata-kata dan penjelasan terucap sehingga mereka menyukai diskusi dan proyek tulisan.

d. Gaya Belajar *Sequential-Global* (Dimensi Pemahaman)

Pembelajar *sequential* memperoleh pemahaman melalui langkah-langkah linear. Pembelajar *global* berusaha memahami gambaran besar dan memecahkan masalah kompleks sekali ketika mendapatkan gambaran besar.

e. Gaya Belajar *Inductive-Deductive* (Dimensi Organisasi)

Pembelajar *inductive* memproses informasi dari yang spesifik ke general, sedangkan pembelajar *deductive* sebaliknya.

2.1.1.2. *Prior Knowledge* dalam Pembelajaran

Dalam *The Strategic Teaching and Reading Project Guidebook* [30], disebutkan *prior knowledge* dapat dijelaskan sebagai kombinasi dari sikap pembelajar yang telah ada sebelumnya, pengalaman-pengalaman, dan pengetahuan mereka [12]. Alice Moses [25] mengatakan siswa belajar lebih efektif ketika mereka sudah mengetahui sesuatu tentang sebuah area konten serta bila konsep-konsep di dalam area tersebut memiliki suatu makna bagi mereka dan bagi latar belakang dan budaya mereka. Jika pengajar menghubungkan informasi baru dengan *prior knowledge* siswa, mereka mengaktifkan ketertarikan dan rasa ingin tahu siswa, dan memompakan pengajaran dengan makna [13].

Prior knowledge setiap pembelajar bisa sama, hampir sama, atau bahkan secara signifikan tidak sama. Secara eksplisit, perbedaan *prior knowledge* di antara pembelajar dapat diketahui melalui *pre-test* yang dilakukan sebelum pembahasan materi inti. Bila pengajar dalam mengajar menggunakan tempo stabil saja, maka pembelajar yang memiliki *prior knowledge* sedikit atau tidak sejalan, kemungkinan besar ia akan gagal [5].

2.1.2. Personalisasi Pembelajaran

Personalisasi pembelajaran adalah penyediaan berbagai fasilitas belajar seperti bahan ajar dan lingkungan pembelajaran berdasarkan karakteristik masing-masing peserta didik [6].

Sebelum membahas mengenai personalisasi pembelajaran online, lebih dahulu akan ditinjau mengenai personalisasi itu sendiri.

Personalisasi: Asal dan Evolusi [14]

Personalisasi di internet datang sebagai hasil proses perubahan yang panjang yang dipercepat dengan perkembangan pesat World Wide Web pada tahun 1990-an [27].

Dengan internet dan web, alat komunikasi baru telah tersedia, memungkinkan orang-orang dengan karakteristik dan tujuan berbeda mengakses informasi yang banyak dan terus berkembang untuk keperluan pribadi. Namun, dalam kondisi seperti ini, orang-orang menjadi sulit mendapatkan informasi yang tepat pada waktu yang tepat dan tingkat kerincian yang tepat. Untuk mendapatkan solusi dari masalah ini para peneliti dari berbagai komunitas yang berbeda-beda telah mengembangkan sistem dengan kemampuan untuk menyesuaikan perilakunya dengan tujuan, kerja, minat, dan fitur lainnya dari pengguna-pengguna perorangan atau kelompok. Hasilnya adalah apa yang kita sebut *adaptive system* atau *personalized system/hypermedia*. Perbedaan sistem ini dengan *traditional static web* adalah pembuatan model pengguna yang merepresentasikan karakteristik pengguna, menggunakannya dalam pembuatan konten dan presentasi disesuaikan dengan individu-individu yang berbeda [28]. Dengan menerapkannya, personalisasi menjadi sebuah alat yang berguna dalam pemilihan dan penyaringan informasi bagi pengguna, memfasilitasi navigasi dan meningkatkan kecepatan sesuai keinginan bahwa pencarian pengguna berhasil.

Principal application area untuk *adaptive web systems* adalah *e-commerce*, di mana personalisasi telah datang untuk menandakan paradigma baru produksi yang berbeda dengan model produksi massal pada abad ke-20. Peppers dan Rogers [18], dijelaskan dalam sebuah buku *marketing* [17], mengilustrasikan kemungkinan perbedaan pembuatan dan penjualan produk dan jasa sesuai dengan profil pelanggan. *Web personalization* memungkinkan perusahaan untuk menawarkan produk, jasa dan iklan yang mempertimbangkan minat, karakteristik, dan kebutuhan pelanggan, sesuai dengan informasi yang didapat selama navigasi oleh pengguna pada *website* perusahaan, contohnya lihat Personalization Consortium (<http://www.personalization.org>), yang mempunyai kepentingan pada teknologi pemasaran *one-to-one*. Mungkin contoh yang telah memimpin adalah Amazon, retailer dan toko buku online yang memonitor minat pengguna melalui navigasi mereka sebelumnya ke situs web.

2.1.3. Kelebihan Pembelajaran Online [6]

Implementasi konsep personalisasi pembelajaran dalam konteks pembelajaran konvensional ternyata sulit dilakukan. Hal ini disebabkan beragamnya gaya belajar maupun *prior knowledge* serta kecerdasan yang dimiliki pembelajar. Sementara di sisi lain, pengajar hanya menerapkan satu tipe metode pembelajaran dalam satu rentang waktu yang paling tidak sesuai dengan satu tipe gaya belajar saja serta menerapkan standar *prior knowledge* dan kecerdasan yang sama bagi para siswa. Bila hal ini terjadi, maka pembelajar lain yang berbeda gaya belajar akan terabaikan, pembelajar dengan *prior knowledge* kurang sulit mengikuti, sedangkan dengan *prior knowledge* dan kecerdasan lebih baik akan meninggalkan mereka yang kurang. Pengajar dapat menyesuaikan dengan standar *prior knowledge* dan tingkat kecerdasan terendah, tetapi ini tidak efisien serta tidak dapat mengakomodasi siswa yang memiliki kemampuan lebih tinggi. Namun dalam konteks pembelajaran online, personalisasi sangat memungkinkan untuk dilakukan yaitu melakukan serangkaian perlakuan terhadap *learning management system* dan *learning object* yang adaptif terhadap tipe gaya belajar, *prior knowledge*, dan tingkat kecerdasan yang beraneka ragam. Personalisasi mampu menghadirkan perlakuan yang berbeda terhadap sejumlah peserta didik dalam rentang waktu bersamaan (paralel).

2.2. Semantic Web

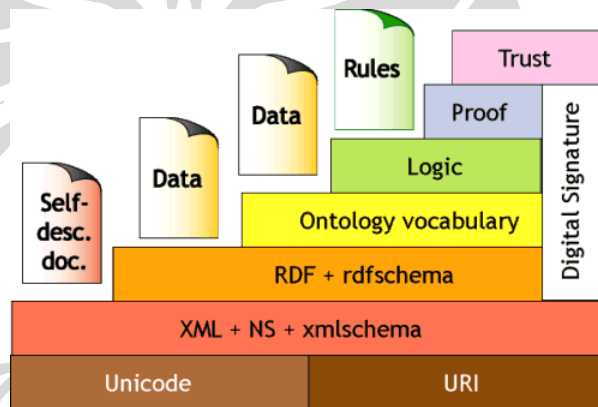
Semantic web adalah web yang dapat mendeskripsikan “sesuatu” dengan suatu cara untuk dapat dimengerti oleh komputer [32]. Contohnya, terdapat pernyataan-pernyataan sebagai berikut:

- a. Ani lahir di Jakarta.
- b. DKI Jakarta adalah ibu kota Negara Republik Indonesia.
- c. Bali adalah sebuah pulau di Indonesia.

Kalimat-kalimat seperti di atas dapat dimengerti oleh manusia, tetapi bagaimana caranya agar dapat dimengerti oleh komputer?

Pernyataan-pernyataan dibangun dengan aturan sintaks. Sintaks dari sebuah bahasa mendefinisikan aturan-aturan untuk membangun pernyataan-pernyataan, tetapi bagaimana sintaks menjadi semantik? Tentang hal inilah *semantic web*, menjelaskan sesuatu dengan suatu cara sehingga dapat dimengerti aplikasi-aplikasi komputer.

Semantic web memungkinkan kita untuk berbagi konten melewati batas aplikasi-aplikasi dan situs-situs web [31]. Dengan kemampuan *machine readable*, mesin dapat menganalisa dan memproses data pada web serta menemukan *link* di antara data tersebut, sehingga memungkinkan integrasi data dari berbagai sumber dan format yang berbeda [20].



Gambar 2.1. *Semantic Web Layer*

<http://www.dlib.org/dlib/november05/bartolo/sweb-stack.gif>

Semantic web terbagi dalam beberapa layer arsitektur [37, 19] seperti dapat dilihat pada Gambar 2.1. yang terdiri atas:

- Unicode dan URI : Unicode adalah standar representasi karakter komputer. URI (Uniform Resource Identifier) adalah standar untuk lokasi dan identitas suatu *resource* (misalnya *web page*).
- XML + NS + XML(S): XML (Extensible Markup Language) dan Namespace serta Schema, merupakan aturan sintaks yang berfungsi menyediakan struktur

- pada web, tetapi tidak menentukan *constraint* semantik pada makna dari dokumen-dokumen tersebut.
- c. RDF adalah model data sederhana yang mengacu pada objek ("*resource*") dan bagaimana mereka berkaitan. Model berdasarkan RDF dapat dinyatakan dalam sintaks XML.
 - d. RDF Schema adalah *vocabulary* untuk menyatakan *property* dari *class* dari *RDF resource*, dengan semantik untuk hierarki generalisasi dari *property* dan *class* tersebut.
 - e. OWL menambahkan *vocabulary* lebih banyak untuk menyatakan *property* dan *class* antara lain hubungan antar *class* (contoh: *disjointness*) dan kardinalitas.
 - f. *Logic* dan *Proof*: Layer ini berupa *rule* dan sistem untuk melakukan *reasoning* pada ontologi sehingga dapat disimpulkan apakah suatu *resource* memenuhi syarat tertentu.
 - g. *Trust*: Layer terakhir dari *semantic web* yang memungkinkan pengguna *web* untuk mempercayai suatu informasi pada web.

Ontologi adalah *backbone* dari *semantic web* [33]. Secara harfiah [16], ontologi adalah cabang filsafat yang membicarakan tentang yang ada. Dalam kaitan dengan ilmu, landasan ontologi mempertanyakan tentang objek yang ditelaah oleh ilmu, bagaimana wujud hakikinya, serta bagaimana hubungannya dengan daya tangkap manusia yang berupa berpikir, merasa, dan mengindera yang membuahkan pengetahuan.

Dalam konteks ilmu komputer dan ilmu informasi, sebuah ontologi mendefinisikan sebuah kumpulan dari primitif-primitif yang merepresentasikan sesuatu untuk memodelkan sebuah domain pengetahuan [34]. Primitif-primitif tersebut dapat berupa kelas-kelas (atau himpunan-himpunan), atribut-atribut (atau properti-properti), dan hubungan-hubungan (atau relasi antara anggota-anggota kelas). Definisi primitif meliputi informasi tentang makna dan batasan-batasan.

Sebuah *ontology language* adalah sebuah bahasa formal yang digunakan untuk mengkodekan ontologi. *Ontology languages* ada beberapa macam, berbeda pada tingkat ekspresivitasnya. *Expressivity* [21] adalah kemampuan suatu bahasa pemodelan untuk menggambarkan aspek-aspek tertentu dunia. Semakin ekspresif suatu bahasa pemodelan dapat mengekspresikan bermacam-macam pernyataan tentang model yang semakin luas. Bahasa-bahasa *semantic web* berurutan dari tingkat ekspresivitas terendah sampai paling tinggi yang telah direkomendasikan W3C sejak tahun 2003 adalah RDF, RDFS, dan OWL. Berikut ini akan dijelaskan satu per satu mengenai ketiganya.

RDF (Resource Description Framework)

RDF adalah *framework* dasar *semantic web*. RDF menyediakan sebuah mekanisme untuk memungkinkan seseorang membuat sebuah pernyataan dasar tentang sesuatu dan melapis pernyataan-pernyataan tersebut menjadi sebuah model tunggal [21].

Dalam *semantic web* kita menghubungkan sesuatu di dunia sebagai *resource*. Sebuah *resource* dapat berupa apa saja yang ingin kita bicarakan. Shakespeare, Jakarta, “nilai dari X” dan “semua sapi di Texas” adalah contoh-contoh dari sesuatu yang ingin kita bicarakan dan dapat dijadikan *resource* pada *semantic web*.

Dalam sebuah web informasi, seseorang dapat berkontribusi kepada pengetahuan kita tentang sebuah *resource*. Ini adalah aspek pada web masa sekarang yang diizinkan berkembang dengan sangat cepat. Untuk mengimplementasi *semantic web*, kita perlu sebuah model data yang membuat informasi terdistribusi pada web.


Ada beberapa strategi untuk menyimpan data pada web. Strategi pertama adalah data disimpan baris per baris sedangkan strategi kedua data disimpan kolom per kolom. Tabel 2.1. berikut adalah contoh data mengenai buku pemrograman.

Tabel 2.1. Contoh Data Tentang Buku Pemrograman


ID	Judul	Pengarang	Tahun
1	Introduction to Algorithms	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein	2001
2	Beginning Programming For Dummies	Wallace Wang	2006
3	The Algorithm Design Manual	Steven S. Skiena	2008
4	Invitation to Computer Science: C++ Version	G. Michael Schneider, Judith Gersting	2004
5	Java Concepts for AP Computer Science	Cay S. Horstmann	2007
6	The Science of Programming	David Gries	2007
7	Introduction Formal Specification And Z	Ben Potter, Jane Sinclair, David Till	1996
8	A Practical Theory of Programming	Eric C.R. Hehner	1993

Pada Gambar 2.2 ditunjukkan strategi pendistribusian data pada banyak mesin (*server*), setiap mesin bertanggungjawab untuk memelihara informasi mengenai beberapa baris dari tabel. Sedangkan pada gambar 2.3 setiap mesin bertanggung jawab untuk memelihara informasi lengkap dari satu kolom atau lebih dari tabel.


Strategi Gambar 2.2 memberikan fleksibilitas karena setiap mesin dapat berbagi beban, namun kekurangannya setiap *server* harus berbagi informasi mengenai kolom karena data pada suatu kolom pada suatu *server* berkorespondensi dengan data pada *server* lainnya pada kolom yang sama. Sedangkan strategi pada Gambar 2.3. cukup memberikan fleksibilitas dengan cara berbeda, di mana jika kita hanya tertarik satu jenis informasi maka kita hanya perlu mengambil informasi dari satu *server*, tidak perlu yang lainnya dan hanya perlu menambah *server* tanpa mengganggu *server* yang lain apabila kita ingin menambah informasi pada suatu entitas. Sama dengan strategi Gambar 2.2, strategi ini juga memerlukan kordinasi antar *server*. Bagaimana kita dapat mengetahui baris 3 pada suatu *server* mengacu kepada entiti yang sama pada baris 3 *server* lainnya?



1	Introduction to Algorithms	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein	2001
---	----------------------------	--	------




4	Invitation to Computer Science: C++ Version	G.Michael Schneider, Judith Gersting	2004
6	The Science of Programming	David Gries	2007



7	Introduction Formal Specification And Z	Ben Potter, Jane Sinclair, David Till	1996
3	The Algorithm Design Manual	Steven S. Skiena	2008


Gambar 2.2 Strategi Distribusi Data pada Web- Baris per Baris



Judul	
Introduction to Algorithms	
Beginning Programming For Dummies	
The Algorithm Design Manual	
Invitation to Computer Science: C++ Version	
Java Concepts for AP Computer Science	
The Science of Programming	
Introduction Formal Specification And Z	
A Practical Theory of Programming	

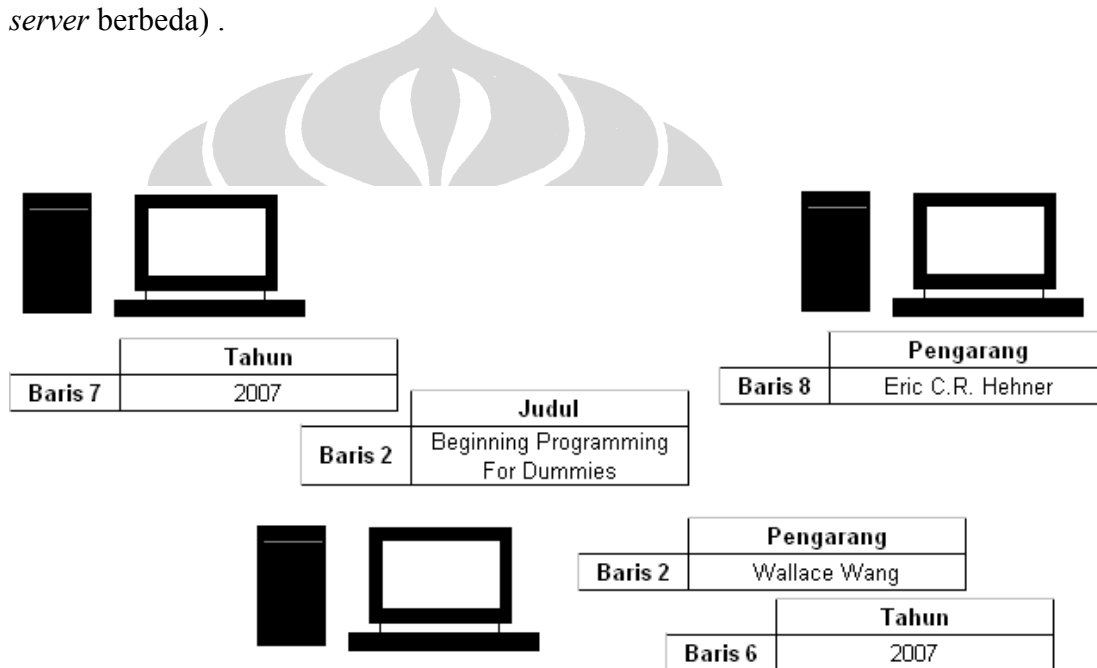
Pengarang	
Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein	
Wallace Wang	
Steven S. Skiena	
G.Michael Schneider, Judith Gersting	
Cay S. Horstmann	
David Gries	
Ben Potter, Jane Sinclair, David Till	
Eric C.R. Hehner	

Tahun
2001
2006
2008
2004
2007
2007
1996
1993



Gambar 2.3 Strategi Distribusi Data pada Web- Kolom per Kolom

Strategi yang ditunjukkan pada Gambar 2.4. adalah kombinasi dua strategi sebelumnya, di mana informasi tidak didistribusikan baris per baris atau kolom per kolom, tetapi *cell per cell*. Setiap mesin bertanggung jawab untuk beberapa *cell* pada tabel. Sistem ini mengkombinasikan fleksibilitas dari strategi sebelumnya. Dua *server* dapat berbagi deskripsi suatu entitas (pada gambar, tahun dan judul dari "Beginning Programming for Dummies" disimpan terpisah), dan dapat pula berbagi penggunaan suatu properti (pada gambar, pengarang dari Baris 2, 6, dan 8 direpresentasikan pada *server* berbeda) .



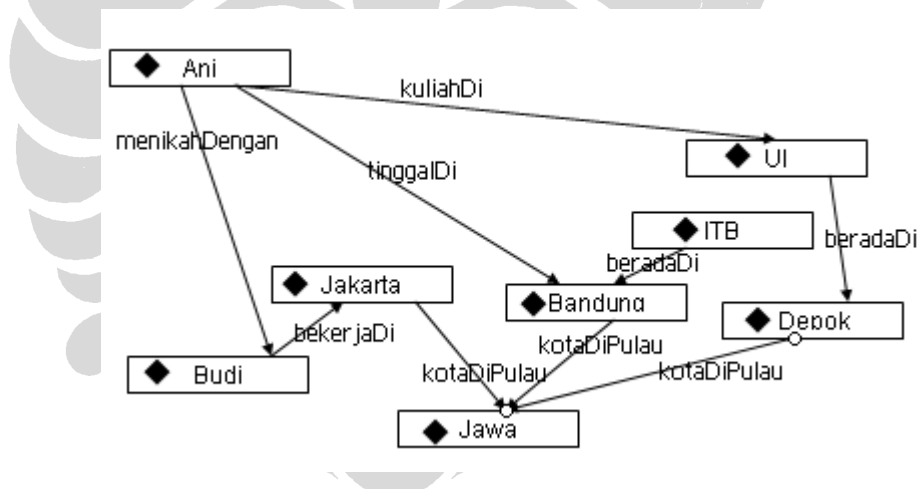
Gambar 2.4. Strategi Distribusi Data pada Web- *Cell per Cell*

Setiap *cell* direpresentasikan dengan tiga nilai pada strategi terakhir; penunjuk untuk baris, penunjuk untuk kolom, dan *value* dari *cell* itu sendiri. Strategi inilah yang diambil oleh RDF. Karena dibangun dengan 3 *value*, *basic building block* untuk RDF disebut dengan *triple*. Penunjuk untuk baris disebut dengan *subject*, penunjuk untuk kolom, disebut dengan *predicate*, sedangkan *value* pada *cell* disebut dengan *object*.

Tabel 2.2. Contoh *Triples*

Subject	Predicate	Object
Baris 7	Tahun	2007
Baris 2	Judul	Beginning Programming For Dummies
Baris 8	Pengarang	Eric C.R. Hehner
Baris 2	Pengarang	Wallace Wang
Baris 6	Tahun	2007

Triple akan semakin menarik apabila terdapat banyak entitas yang saling berhubungan. Misalnya saja Ani kuliahDi UI; Ani tinggalDi Bandung; UI beradaDi Depok; Jakarta, Bandung, Depok kotaDiPulau Jawa; Ani menikahDengan Budi; Budi bekerjaDiJakarta; dan ITB beradaDi Bandung. Hubungan-hubungan tersebut dapat direpresentasikan dengan sebuah graf.

Gambar 2.5. Representasi *Triples* dengan Graf

Ada banyak hal di dunia dan sering kali sesuatu terlihat menunjukkan hal yang sama padahal sama sekali berbeda. Misalnya, ada "Jakarta" yang menunjukkan kota, ada juga "Jakarta" menunjukkan jalan, atau mungkin ada pula "Jakarta" yang menunjuk ke sebuah toko. Oleh karena itu standar penamaan pada RDF menggunakan URI (Uniform Resource Identifier), seperti <http://example.com/index.html> yang dapat diikuti

dengan *fragment identifier*, yaitu setelah tanda '#' misalnya `http://example.com/index.html#book1` [19]. Contoh terakhir dinamakan URI *reference* (URIref) dan digunakan untuk penamaan sesuatu yang dinyatakan dalam RDF, yaitu sebagai *resource ID*. URI dan URL terlihat sama persis, namun URL hanyalah *special case* dari URI. Untuk menyederhanakan penulisan URI, RDF menggunakan *qualified names* (QNnames) yang terdiri atas *prefix* untuk suatu *namespace*, diikuti oleh tanda ':' dan *local name* (ID). Contohnya apabila `ex` adalah *prefix* untuk *namespace* `http://example.com/index.html` maka `http://example.com/index.html#book1` dapat disingkat dengan `ex:book1`.

Cara penulisan RDF ada beberapa, misalnya: RDF/XML, N-Triple, N3, Turtle, dan lain-lain. Format N3 merupakan format yang sederhana dan mudah dipahami sedangkan format RDF/XML disebut juga *serialization format* memiliki sintaks yang lebih rumit. W3C merekomendasikan format RDF/XML karena kebanyakan infrastruktur web merepresentasikan informasi dengan HTML, atau lebih umum, dengan XML. Gambar 2.6. dan 2.7. menunjukkan contoh kedua sintaks RDF tersebut dan memiliki arti yang sama

```
@prefix sm:    <http://www.owl-ontologies.com/> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dc:   <http://purl.org/dc/elements/1.1/> .
sm:student_model.owl dc:creator "Leonny Pramitasari"
```

Gambar 2.6. Contoh Sintaks RDF dalam N3 *Format*

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

  <rdf:Description rdf:about="http://www.owl-
ontologies.com/student_model.owl">
    <dc:creator>Leonny Pramitasari</dc:creator>
  </rdf:Description>
</rdf:RDF>
```

Gambar 2.7. Contoh Sintaks RDF dalam *Serialization Format*

RDF menyediakan cara yang konsisten dalam merepresentasikan data sehingga informasi dari banyak sumber dapat dibawa bersama dan diperlakukan seperti dari satu sumber. *Semantic web* menyediakan *inferencing*, yaitu proses di mana *triples* baru secara sistematis ditambahkan berdasarkan pola *triple* yang sudah ada. *Triple* baru tersebut disebut *inferred triple*, sedangkan *triple* yang dimasukkan adalah *asserted triple*. Contohnya adalah sebagai berikut:

- a. Terdapat RDFS *rules* yang menyatakan apabila A *subclass* dari B kemudian ada suatu X yang bertipe A, maka ia juga bertipe B;

```
IF
?A rdfs:subClassOf ?B.
AND
?X rdf:type ?A.
THEN
?X rdf:type ?B.
```

- b. Apabila terdapat *asserted triple*:

```
tb:BukuKomputer rdfs:subClassOf tb:Buku .
tb:Linux1 rdfs:type tb:BukuKomputer .
```

maka akan terdapat *inferred triple*:

```
tb:Linux1 rdfs:type tb:Buku .
```

RDFS (RDF Schema Language)

RDFS adalah bahasa dengan ekspresivitas untuk menggambarkan *basic notions* mengenai penggunaan komponen yang sama dan variabilitas yang sudah lazim dari *object languages* dan *system class* lainnya, yang dinamakan *classes*, *subclasses*, dan *properties* [21]. RDFS digunakan untuk mendefinisikan kosakata yang dipakai pada RDF. Pada Tabel 2.3. dan 2.4. berisi beberapa daftar kosakata utama berupa *class* dan *property* yang didefinisikan dalam RDFS [35, 19].

Tabel 2.3. RDF *Classes*

<http://www.w3.org/TR/rdf-schema/>

Class name	Comment
rdfs:Resource	The class resource, everything.
rdfs:Literal	The class of literal values, e.g. textual strings and integers.
rdf:XMLLiteral	The class of XML literals values.
rdfs:Class	The class of classes.
rdf:Property	The class of RDF properties.
rdfs:Datatype	The class of RDF datatypes.
rdf:Statement	The class of RDF statements.
rdf:Bag	The class of unordered containers.
rdf:Seq	The class of ordered containers.
rdf:Alt	The class of containers of alternatives.
rdfs:Container	The class of RDF containers.
rdfs:ContainerMembershipProperty	The class of container membership properties, rdf:_1, rdf:_2, ..., all of which are sub-properties of 'member'.
rdf:List	The class of RDF Lists.

Tabel 2.4. RDF *Properties*

<http://www.w3.org/TR/rdf-schema/>

Property name	Comment	domain	range
rdf:type	The subject is an instance of a class.	rdfs:Resource	rdfs:Class
rdfs:subClassOf	The subject is a subclass of a class.	rdfs:Class	rdfs:Class
rdfs:subPropertyOf	The subject is a subproperty of a property.	rdf:Property	rdf:Property
rdfs:domain	A domain of the subject property.	rdf:Property	rdfs:Class
rdfs:range	A range of the subject property.	rdf:Property	rdfs:Class
rdfs:label	A human-readable name for the subject.	rdfs:Resource	rdfs:Literal
rdfs:comment	A description of the subject resource.	rdfs:Resource	rdfs:Literal
rdfs:member	A member of the subject resource.	rdfs:Resource	rdfs:Resource
rdf:first	The first item in the subject RDF list.	rdf:List	rdfs:Resource
rdf:rest	The rest of the subject RDF list after the first item.	rdf:List	rdf:List
rdfs:seeAlso	Further information about the subject resource.	rdfs:Resource	rdfs:Resource
rdfs:isDefinedBy	The definition of the subject resource.	rdfs:Resource	rdfs:Resource
rdf:value	Idiomatic property used for structured values (see the RDF Primer for an example of its usage).	rdfs:Resource	rdfs:Resource
rdf:subject	The subject of the subject RDF statement.	rdf:Statement	rdfs:Resource
rdf:predicate	The predicate of the subject RDF statement.	rdf:Statement	rdfs:Resource
rdf:object	The object of the subject RDF statement.	rdf:Statement	rdfs:Resource

OWL (Web Ontology Language) [36,19]

OWL menyediakan fleksibilitas dan kekayaan ekspresi (hubungan-hubungan dan batasan-batasan) yang dibutuhkan untuk berbagai macam aplikasi berbasis web dengan kebutuhan untuk memungkinkan *logical reasoning* secara otomatis. OWL merupakan ekstensi dari RDFS. Karena terkadang ada kebutuhan yang bertentangan maka dibuat tiga 'species' OWL; OWL Full, OWL Lite dan OWL DL. Pilihan antara OWL Lite dan OWL DL bergantung pada cukup tidaknya konstruksi dengan OWL Lite (apakah perlu menggunakan yang lebih ekspresif dengan 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 *meta classes (classes of classes)*.

Elemen pada OWL terdiri atas *classes, properties, instances of classes*, dan relasi antar *instances*. OWL mendefinisikan *root* dari semua yang ada dengan `owl:Thing`. Jadi semua *class* yang dibuat secara implisit merupakan *subclass* dari `owl:Thing`. Definisi *class* dapat dibedakan menjadi dua bagian, yaitu pemberian nama atau *reference*; dan daftar keterangan atau *restriction* yang berlaku untuk *class* tersebut, misalnya *subClassOf*.

Individuals atau disebut juga *instances* adalah anggota (*members*) dari *classes*. *Instances* ini dapat dipandang sebagai objek yang ada pada domain yang dibahas.

Property merupakan *binary relation*. Ada dua jenis *property* pada OWL, yaitu *ObjectProperty* (relasi antara *instance* dari dua *classes*) dan *DatatypeProperty* (relasi antara *instance* dengan RDF *literal* dan XML *Schema datatypes*). Sama halnya seperti *class* yang dapat dinyatakan secara hierarkis, begitu pula properti dapat dinyatakan sebagai *subPropertyOf* dengan `rdfs:subPropertyOf`. Untuk memberikan batasan pada suatu *property*, dapat digunakan `rdfs:domain` dan `rdfs:range`, yang disebut juga sebagai *global restriction* karena berlaku untuk umum, tidak terbatas pada *class* tertentu.

Ada dua hal terkait dengan *property*, yaitu:

- a. *Characteristics*, memberikan keterangan tambahan untuk *property*, antara lain: ***inverseOf***, ***TransitiveProperty***, ***SymmetricProperty***, dan ***InverseFunctionalProperty***.
- b. *Restriction*, disebut juga sebagai *local restriction* karena memberikan batasan pada definisi suatu *class*. *Restriction* ini ada tiga macam, yaitu: *quantifier*, *cardinality*, dan *hasValue*. Untuk menentukan *quantifier* digunakan *allValuesFrom* dan *someValuesFrom*.

Implementasi pada OWL akan dibahas lebih lanjut pada tugas akhir ini pada butir 4.4.4.3. Pengembangan dengan OWL.

2.3. Penelitian Sebelumnya yang Terkait

Ada banyak sekali penelitian yang pernah dijalankan terkait dengan penelitian ini, sehingga tidak dapat dijelaskan seluruhnya. Penulis hanya akan membahas beberapa penelitian yang menjadi rujukan, dibagi ke dalam penelitian terkait di lingkungan Fasilkom UI dan penelitian tentang *student model ontology*.

2.3.1. Penelitian di Fakultas Ilmu Komputer Universitas Indonesia

Penelitian berbasis *semantic web* yang pernah dilakukan adalah pengembangan prototipe *semantic portal* berbasis ontologi komunitas riset [19] pada tahun 2008. Penelitian ini melakukan *reuse* ontologi yang sudah ada yaitu ontologi komunitas riset SWRC dengan melakukan modifikasi.

Penelitian tentang personalisasi pembelajaran online yang tidak berbasis *semantic web* sudah lebih dahulu dilakukan, salah satunya adalah pemodelan personalisasi pembelajaran berdasarkan aspek gaya belajar pada Student Centered e-Learning Environment [6] pada tahun 2007. Kemudian penelitian serupa dilakukan pada tahun 2008 dengan mempertimbangkan *prior knowledge* [5].

Pada tahun 2008 juga telah dikembangkan sistem untuk *administrator* dan pengajar berupa *authoring tool* yang dinamakan dengan SHECAR (Sharable Content E-Learning Authoring Tool and Repository) [9]. Dengan SHECAR, penyedia materi *e-learning*, yaitu pengajar atau dosen dapat membangun materi ajar yang bersifat *reusable* dan *sharable*. Materi ajar yang dibangun juga diharapkan dapat mendukung personalisasi dengan diberikan informasi berupa *metadata*. Dalam sistem tersebut setiap obyek pembelajaran diberikan sebuah "makna" tertentu dengan menggunakan sebuah metadata [38]. *Metadata* pada SHECAR menjadi dasar pertimbangan pengembangan ontologi *learning object* yang dilakukan secara paralel dengan penelitian *student model ontology* ini. Pengembangan ontologi *learning object* yang dilakukan tentunya untuk saling melengkapi dengan *student model ontology*.

2.3.2. Ontologi *Student Model* yang Pernah Dikembangkan

Berdasarkan hasil pencarian melalui *Google Search Engine* yang didapat penulis, pemodelan ontologi *student* maupun penjelasannya secara lengkap masih sedikit dibandingkan pengembangan ontologi *learning object*. Kemudian penulis mengambil tiga referensi pemodelan yang dianggap dapat menjadi pertimbangan untuk pengembangan *student model ontology* yang disesuaikan dengan kebutuhan Fakultas Ilmu Komputer Universitas Indonesia.

Salah satu referensi adalah *user model ontology* yang termasuk ke dalam LOCO (Learning Object Context Ontologies). LOCO¹ sejauh ini terdiri dari *learning context ontology*, *user model ontology*, *learning object content structure ontology*, dan *domain ontology*. Sedangkan *learning design ontology* sedang dalam pengembangan. *User model ontology* LOCO beserta tiga ontologi kecil yang digunakan; ontologi untuk pemodelan kompetensi, ontologi untuk pemodelan *learner performance*, dan ontologi untuk pemodelan preferensi pengguna menjadi bahan referensi untuk pengembangan *student model ontology*.

¹ <http://iis.fon.rs/LOCO-Analyst/loco.html>

Dalam situsnya dijelaskan, LOCO-Analyst adalah sebuah *educational tool* yang ditujukan untuk memperlengkapi pengajar dengan umpan balik pada aspek yang relevan dengan proses pembelajaran di lingkungan pembelajaran berbasis web, sehingga membantu mereka meningkatkan konten dan struktur mata kuliah berbasis web. LOCO-Analyst memperlengkapi pengajar dengan umpan balik yang memperhatikan semua jenis aktivitas siswa yang dilakukan dan atau diikuti selama proses pembelajaran; kegunaan dan kemampuan untuk dimengerti dari konten pembelajaran yang disiapkan dan dikembangkan pada LCMS; serta interaksi sosial antara siswa yang kontekstual pada lingkungan pembelajaran virtual [23].

Selanjutnya pembahasan mengenai *user model ontology* LOCO dan lainnya akan dibahas pada Bab 4 Pelaksanaan.

