

## **SISTEM PENILAIAN OTOMATIS KEMIRIPAN KALIMAT MENGUNAKAN *SYNTACTIC-SEMANTIC SIMILARITY* PADA SISTEM *E-LEARNING***

**\*Deddy Winarsono, \*\* Daniel O Siahaan, \*\*\* Umi Yuhana**

Jurusan Teknik Informatika, ITS

Jl.Raya ITS, Kampus ITS, Sukolilo, Surabaya, 60111

E-Mail: \* deddysw@cs.its.ac.id, \*\* daniel@if.its.ac.id, \*\*\* yuhana@cs.its.ac.id

### **Abstract**

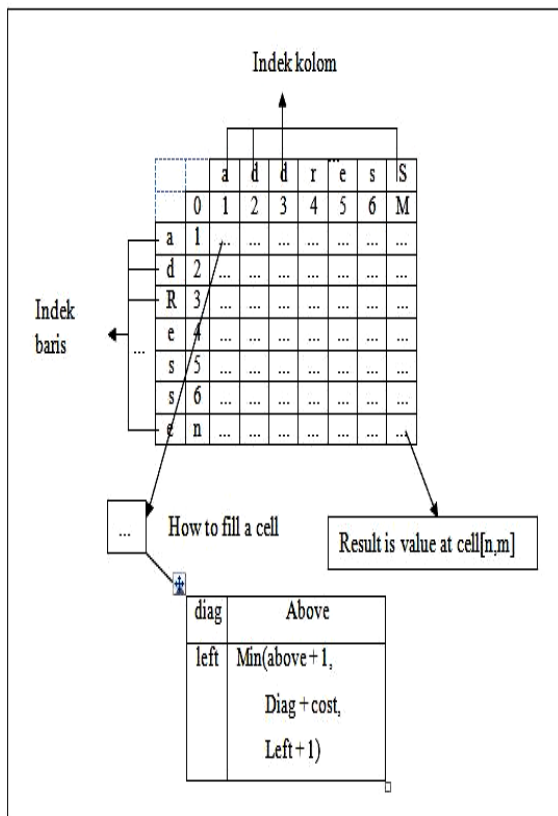
*Autograding system on the essay answer has become the main focus to be a research since past up to now. The problems faced is how to assess every answer of the students Objectively, consistenly, and fast. The variety essay answers cause the difficulty to evaluate automatically with correcting among words in an answer of the sentences as of the key answers sentences. One of the ways to solve this problem is comparing among words based on the sentence structure. This research designed and constructed a new method of the problem on accounting sentence similarity syntatically and semantically by identifying the sentence fisrt. This method is known as SynSemSim method. The result of this research is a software calculating sentence similarity syntatically and semantically which knows the sentence structure by using Penn treebank model. Try out of this model used sintatic and questionnaire data. Syntatic data had nine scenarios, every scenario has two till six simple sentences which are compared to the value of the sentence similarity. The result of try out of syntatic data of SymSemSim method closed to the definition of try out scenario. On the other hand, the result of try out of questionnaire data of SysmseSym method resulted in deviation mean having 2,61 as the the worst case.*

*Key words: Autograding, Semantic Similarity, Sintatic Similarity, Penn-Treebank.*

### **PENDAHULUAN**

Penggunaan *e-Learning* memungkinkan pembelajaran dilakukan dengan jumlah mahasiswa yang banyak, sehingga diperlukan sistem penilaian otomatis (*autograding*) pada jawaban dari soal-soal *e-Learning*. Model soal dengan jawaban pendek (*short answer*) merupakan nilai penting dalam mengatur dan menyatukan ide, selain itu juga mampu menginterpretasikan data dan aplikasi dari suatu data. Namun kesulitan terbesar dari model ini adalah bagaimana subyektifitas dari masing-masing jawaban dapat dimengerti [1]. Sistem penilaian otomatis pada jawaban pendek harus memegang prinsip konsisten, konsumsi waktu dan biaya yang kecil sehingga nilai obyektifitasnya akan dapat dipertanggungjawabkan.

Kemiripan *string* (*string similarity*) adalah sebuah nilai mutlak yang menunjukkan hubungan arti antar dua *string*. Masing-masing *string* biasanya terdiri dari beberapa kata atau akronim [2]. Kemiripan sintaktik kata (*syntactic string similarity*) adalah menghitung nilai kemiripan antar dua kata (*string*) berdasarkan struktur huruf penyusun kata [3] (*eats dan eats* adalah sama). Kemiripan arti kata (*syntactic string similarity*) adalah menghitung nilai kemiripan antar dua kata berdasarkan arti katanya (*eats dan consumes* adalah sama), sedangkan kemiripan arti antar kalimat adalah menghitung nilai kemiripan arti antar dua kalimat berdasarkan arti keseluruhan kalimat (bukan arti tiap katanya). Nilai



Gambar 1. Ilustrasi Algoritma *Levenshtein Distance* kata “adresse” dengan “address”.

kemiripan (*similarity score*) diasumsikan pada rentang 0 (nol) hingga 1 (satu), yang artinya nilai 1 adalah nilai maksimum yang menunjukkan bahwa dua kata adalah sama identik. Penelitian sebelumnya yang menggunakan metode *ontology* [4] menghasilkan nilai rata deviasi hasil uji coba dengan data kuisisioner pada kasus terburuk sebesar 4,23. Penelitian lainnya dengan teknik sintaktik dan semantik [2] [3] menghasilkan nilai 1 (sama identik) untuk dua kalimat yang secara arti sebenarnya berbeda (“*John eats apple*” dengan “*Apple eats John*”).

Penelitian ini merancang dan membangun suatu metode baru untuk menyelesaikan permasalahan menghitung kemiripan kalimat secara sintaktik dan semantik dengan mengenali struktur kalimat terlebih dahulu. Penggabungan dua metode (sintaktik dan semantik) dan pengenalan struktur kalimat dengan menggunakan model *penn-treebank* adalah teknik yang digunakan pada penelitian ini.

## KEMIRIPAN SINTAKTIK DAN SEMANTIK

### Kemiripan Sintaktik Antar Kata

Teknik menghitung kemiripan kata secara sintaktik pada penelitian sebelumnya [2] menggunakan algoritma *Levenshtein Distance*. Algoritma ini menghitung kemiripan antar kata berdasarkan total biaya terkecil dari transformasi salah satu kata menjadi kata yang lain dengan menggunakan *edit-rules*, yaitu penambahan karakter (*insertion*), penggantian karakter (*substitution*), dan penghapusan karakter (*deletion*). Contoh ilustrasi dapat dilihat pada Gambar 1.

Algoritma *Levenshtein Distance* dimulai dari elemen pojok kiri atas sebuah larik (*array*) dua dimensi dengan indeks baris adalah karakter sumber dan indeks kolom adalah karakter target yang dibandingkan dengan kata sumber. Tiap  $cell[i,j]$  merepresentasikan biaya terkecil dari transformasi karakter ke- $i$  dari kata sumber ke karakter ke- $j$  dari kata target. Biaya *edit-distance* adalah nilai dari  $cell[n,m]$ . Untuk contoh pada Gambar 1, biaya *edit-distance* adalah pada  $cell[n,m]$  yaitu 2. Setelah mendapatkan biaya *edit-distance* maka untuk menghitung nilai *Levenshtein Distance* menggunakan Persamaan (1).

$$Levenshtein\ Distance(str1, str2) = 1 - \frac{edit\ distance}{maxLength(str1, str2)} \quad (1)$$

Nilai kemiripan (*similarity score*) diasumsikan pada rentang 0 (nol) hingga 1 (satu), yang artinya nilai 1 adalah nilai maksimum yang menunjukkan bahwa dua kata adalah sama identik. Pendekatan yang digunakan oleh penelitian ini mampu mengukur nilai kemiripan antar dua *string* berdasarkan pada susunan karakter

### Kemiripan Semantik Antar Kalimat

Teknik menghitung kemiripan antar kalimat adalah menghitung kemiripan arti tiap kata dengan seluruh kata pada kalimat lain [3]. Sebagai contoh:

Kalimat 1: “*John eats apple*”.

Kalimat 2: “*Apple eats John*”.

Proses *tokenizing* dari kedua kalimat tersebut menghasilkan *token*:

$Tokens\ 1 = \{“John”, “eats”, “apple”\}$ .

$Tokens\ 2 = \{“Apple”, “eats”, “John”\}$ .

Tabel 1. *Matrix Similarity* Kemiripan Semantik.

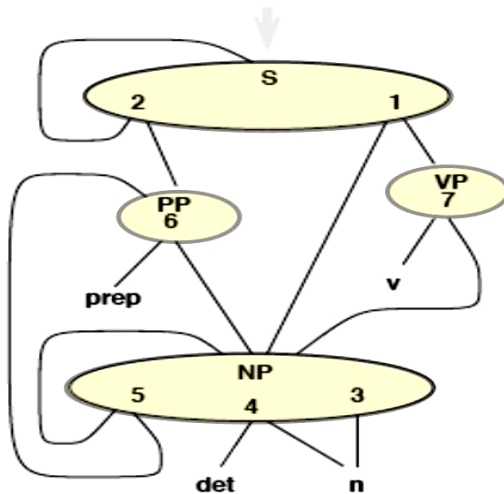
		Kolom 1 <i>Apple</i>	Kolom 2 <i>Eat</i>	Kolom 3 <i>John</i>
Baris 1	<i>John</i>	0.4	0.0	1.0
Baris 2	<i>Eat</i>	0.0	1.0	0.0
Baris 3	<i>Apple</i>	1.0	0.0	0.4

$$\text{SumX} = \max(\text{Cell}[1,1], \text{Cell}[1,2], \text{Cell}[1,3]) + \max(\text{Cell}[2,1], \text{Cell}[2,2], \text{Cell}[2,3]) + \max(\text{Cell}[3,1], \text{Cell}[3,2], \text{Cell}[3,3]) = 1 + 1 + 1 = 3$$

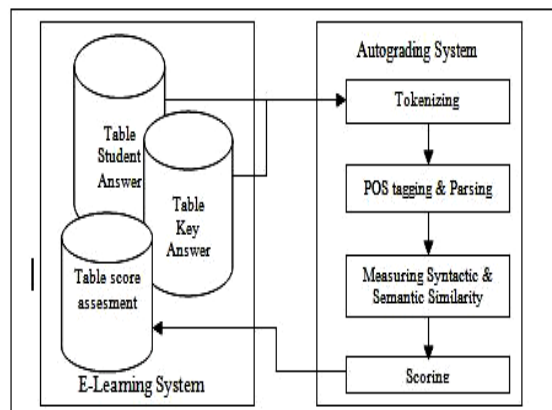
$$\text{SumY} = \max(\text{Cell}[1,1], \text{Cell}[2,1], \text{Cell}[3,1]) + \max(\text{Cell}[1,2], \text{Cell}[2,2], \text{Cell}[3,2]) + \max(\text{Cell}[1,3], \text{Cell}[2,3], \text{Cell}[3,3]) = 1 + 1 + 1 = 3$$

$$\text{Matching Average} = \frac{\text{Match}(X,Y)}{|X| + |Y|} = \frac{3+3}{3+3} = 1$$

Gambar 2. Proses *Fast Heuristic*.



Gambar 3. Hubungan Antar Kata untuk Membentuk Kalimat Baku.



Gambar 4. Rancangan dan Arsitektur Sistem Penilaian Otomatis.

Proses kemiripan semantik dengan menggunakan *thesaurus WordNet* menghasilkan *matrix similarity* seperti yang ditunjukkan dalam Tabel 1. Setelah *matrix similarity* didapatkan, maka dilanjutkan proses *fast heuristic* dan *Matching Average* yang ditunjukkan pada Gambar 2 dengan hasil nilai kemiripan kedua kalimat adalah 1 yang artinya adalah sama identik secara semantik.

### STRUKTUR KALIMAT

Struktur kalimat adalah bentuk dari suatu kalimat yang tersusun dari beberapa kata yang saling terhubung. Secara umum hubungan antar kata digambarkan seperti pada Gambar 3.

Hubungan relasi tersebut bila diterjemahkan menjadi aturan *part of speech* adalah sebagai berikut:

1.  $S \rightarrow NP . VP$
2.  $S \rightarrow S . VP$
3.  $S \rightarrow S conj S$
4.  $NP \rightarrow n$
5.  $NP \rightarrow det . n$
6.  $NP \rightarrow NP . PP$
7.  $NP \rightarrow n$
8.  $NP \rightarrow NP conj NP$
9.  $VP \rightarrow Verb$
10.  $VP \rightarrow Verb . NP . NP$
11.  $VP \rightarrow VP . PP$
12.  $VP \rightarrow v . NP$
13.  $PP \rightarrow prep . NP$

Sebuah kalimat dapat terdiri dari *frase NOUN (NP)* dan *frase VERB (VP)* atau terisi dari *SENTENCE (S)* dan *frase VERB (VP)*. *Frase NOUN (NP)* dapat terdiri dari sebuah kata *noun (n)*, *determiner (DET)* dengan *noun (n)* atau *frase NOUN (NP)* dengan *frase PREPOSITION (PP)*. *Frase PREPOSITION* dapat terdiri *preposition* yang dikombinasikan dengan *frase NOUN (NP)*. Sedangkan *frase VERB* terdiri kata yang bertindak sebagai *verb* dan *frase NOUN (NP)*.

### PERANCANGAN DAN ARSITEKTUR SISTEM SISTEM PENILAIAN OTOMATIS

Gambar 4 memperlihatkan sistem utama yaitu sistem *e-Learning* berbasis *web* dan sistem penilaian otomatis (*autograding system*). Sistem penilaian otomatis mengambil masukan kalimat jawaban yang tersimpan pada tabel jawaban (*table student answer*) dan masukan kalimat kunci jawaban dari tabel kunci jawaban (*table key answer*) dari sistem *e-Learning*

berbasis *web*. Kalimat jawaban mahasiswa dan kunci jawaban diproses pada sistem penilaian otomatis melalui proses *tokenizing*, *POS tagging* dan *parsing*, pengukuran kemiripan kalimat (*measuring syntactic&semantic similarity*), dan penilaian (*scoring*). Nilai hasil proses penilaian disimpan kembali ke tabel hasil nilai ujian (*table score assesment*) pada sistem *e-Learning* berbasis *web*.

**IMPLEMENTASI SISTEM PENILAIAN OTOMATIS MENGGUNAKAN SYNSEMSIM**

Implementasi sistem penilaian otomatis menggunakan *metode syntactic-semantic similarity (SynSemSim)* terdiri dari beberapa bagian yaitu *tokenizing* (memecah kalimat menjadi beberapa *token*), *POS tagging* dan *parsing* (mengidentifikasi struktur kalimat), *measuring similarity* (menghitung kemiripan kalimat dengan menggunakan metode *SynSemSim* menghasilkan *matrix similarity*), *scoring* (menghitung nilai akhir dari *matrix similarity* menggunakan metode *fast heuristic* dan *matching average*). Pada contoh Kalimat 1 “*John eats apple*” dan Kalimat 2 “*Apple eats John*”, proses *tokenizing*:

*Tokens 1* = {“*John*”, “*eats*”, “*apple*”, “*.*”}.  
*Tokens 2* = {“*Apple*”, “*eats*”, “*John*”, “*.*”}.

Setelah mendapatkan *tokens*, maka proses *POS tagging* dan *Parsing* menghasilkan struktur kalimat seperti yang ditunjukkan pada Gambar 5. Dari proses ini didapatkan *matrix similarity* seperti yang ditunjukkan pada Gambar 6. Proses *fast heuristic* masing-masing *matrix*:

- 1.*Matrix 1* menghasilkan nilai 0,4 + 0,4 = 0,8
- 2.*Matrix 2* menghasilkan nilai 1,0 + 1,0 = 2
- 3.*Matrix 3* menghasilkan nilai 0,4 + 0,4 = 0,8
- 4.*Matrix 4* menghasilkan nilai 1,0 + 1,0 = 2

Maka nilai kemiripan adalah

$$\frac{(Matrix\ 1 + Matrix\ 2 + Matrix\ 3 + Matrix\ 4)}{n} = \frac{(0,8 + 2 + 0,8 + 2)}{8} = 0,7$$

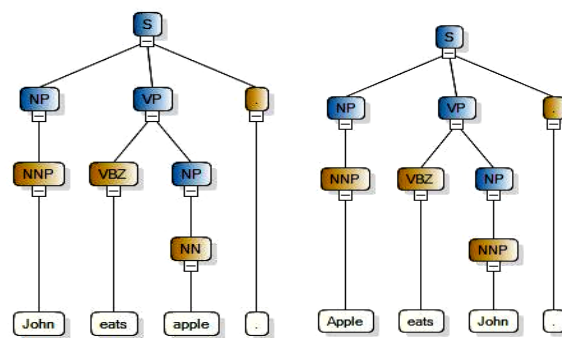
Proses penilaian (*scoring*) dilakukan dengan mengalikan nilai kemiripan dengan angka 10 atau 100 disesuaikan dengan standar penilaian dari dosen.

**HASIL DAN PEMBAHASAN**

Tabel 2. Hasil Uji Coba dengan Data *Sintesa*.

Uji coba pada penelitian ini dilakukan dengan menggunakan dua jenis data, yaitu data sintesis dan data kuisisioner mahasiswa. Pengujian dengan menggunakan data *sintesis* adalah pengujian dengan data yang dibuat untuk membandingkan hasil uji coba penelitian ini dengan penelitian sebelumnya [2][3]. Pengujian dengan data kuisisioner adalah pengujian dengan data pada penelitian sebelumnya [4] yang didapatkan dari kuisisioner dengan soal menggunakan Bahasa Inggris yang dibagikan kepada mahasiswa Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember, pada mata kuliah Pengantar Teknologi Informasi. Kuisisioner ini dijawab dengan menggunakan Bahasa Inggris.

Pada perbandingan hasil pengujian dengan data kuisisioner dilakukan dengan cara membandingkan nilai deviasi antara metode *SynSemSim* dengan metode sebelumnya [2][3][4]. Nilai deviasi didapatkan dengan cara menghitung nilai *absolut* dari selisih antara nilai kemiripan hasil tiap metode dengan nilai manual penilaian dosen. Semakin rendah nilai deviasi maka hasil perhitungan *autograding* semakin mendekati kenyataan atau nilai manual dosen.



Gambar 5. Struktur Kalimat Hasil Proses POS Tagging dan Parsing.

<b>Matrik 1</b>	Kolom 1 Apple	<b>Matrik 2</b>	Kolom 1 eats
Baris 1   John	0,4	Baris 1   Eats	1.0
<b>Matrik 3</b>	Kolom 1 John	<b>Matrik 3</b>	Kolom 1 titik
Baris 1   Apple	0,4	Baris 1   Titik	1.0

Gambar 6. *Matrix Similarity* dari Metode *Synsemsim*.

		<i>SynSemSim</i>	[4]	[3]
Skenario 1	<i>Syntactic</i> dan <i>Semantic</i> adalah sama indentik. <i>Sim</i> = 1			
	1. <i>John eats apple.</i> <i>John eats apple.</i>	1,000	1,000	1,000
	2. <i>I will go to Jakarta.</i> <i>I will go to Jakarta.</i>	1,000	1,000	1,000
	3. <i>I do not understand with your instructions.</i> <i>I do not understand with your instructions.</i>	1,000	1,000	1,000
Skenario 2	<i>Syntactic</i> diasumsikan mirip dan <i>semantic</i> sama. <i>Sim</i> mendekati/samadengan 1			
	1. <i>John eats apple.</i> <i>John consumes apple.</i>	1,000	0,707	1,000
	2. <i>Deddy eats apple.</i> <i>Deddy consumes apple.</i>	1,000	0,720	0,667
	3. <i>Deddy eats a banana.</i> <i>Deddy consumes a banana.</i>	1,000	0,796	0,750
	4. <i>He drives a car.</i> <i>He rides a car.</i>	1,000	0,934	0,800
	5. <i>You'd rather go home</i> <i>You'd better go home</i>	0,917	0,875	0,584
	6. <i>David goes to Jakarta.</i> <i>David comes to Jakarta.</i>	0,934	0,900	0,668
Skenario 3	<i>Syntactic</i> diasumsikan mirip dan <i>semantic</i> sama (terjadi kesalahan eja). <i>Sim</i> mendekati/sama dengan 1			
	1. <i>John eats apple.</i> <i>John eat apple.</i>	1,000	0,917	1,000
	2. <i>John eats apple.</i> <i>John eats apel.</i>	0,900	0,867	0,738
	3. <i>John eats apple.</i> <i>John eats appel.</i>	0,900	0,867	0,738
	4. <i>John eats apple.</i> <i>John eat appel.</i>	0,900	0,783	0,733
Skenario 4	<i>Syntactic</i> dan <i>semantic</i> diasumsikan mirip. <i>Sim</i> mendekati 1			
	1. <i>John eats apple and banana.</i> <i>John eats apple.</i>	0,908	0,803	0,860
	2. <i>John eats apple and banana.</i> <i>John eats banana.</i>	0,921	0,813	0,860
	3. <i>John eats apple and banana.</i> <i>John eats apple, orange, and banana.</i>	0,773	0,939	0,802
	4. <i>John eats apple and banana.</i> <i>John eats banana and apple.</i>	0,960	1,000	0,800
Skenario 5	<i>Semantic</i> sama. Beda model penulisan. <i>Sim</i> = 1			
	1. <i>I will go to Jakarta.</i> <i>I'll go to Jakarta.</i>	1,000	0,806	0,541
	2. <i>I do not understand with your instructions.</i> <i>I don't understand with your instructions.</i>	1,000	0,862	0,508
	3. <i>I'd like to hire someone.</i> <i>I would like to hire someone.</i>	1,000	0,805	0,587
	4. <i>I'll go to Jakarta.</i> <i>I'm going to go to Jakarta.</i>	0,939	0,840	0,500
Skenario 6	<i>Syntactic</i> diasumsikan mirip dan <i>semantic</i> tidak sama. <i>Sim</i> mendekati 0.			

Skenario 7	1. <i>John eats apple then banana.</i> <i>John eats banana then apple.</i>	0,960	1,000	1,000
	Syntactic dan semantic tidak sama tetapi salah satu bukan kalimat yang baku. <i>Sim</i> mendekati 0.			
	1. <i>John eats apple.</i> <i>Apple eats John.</i>	0,700	1,000	1,000
Skenario 8	2. <i>John eats apple.</i> <i>Eats John apple?</i>	0,750	1,000	1,000
	Syntactic dan semantic adalah tidak sama keduanya adalah kalimat baku. <i>Sim</i> mendekati 0.			
	1. <i>John eats apple.</i> <i>Dog catches cat.</i>	0,558	0,370	0,557
Skenario 9	Membandingkan dua kalimat dengan tipe yang berbeda.. <i>Sim</i> mendekati/samadengan 0.			
	1. <i>John eats apple.</i> <i>Does john eats apple?</i>	0,720	0,893	0,926
	2. <i>John eats apple.</i> <i>John doesn't eat apple.</i>	0,867	0,899	0,857
	3. <i>John doesn't eat apple.</i> <i>Does John eats apple?</i>	0,750	0,893	0,810

Tabel 3. Hasil Uji Coba dengan Data Kuisisioner Nilai Rata Tiap Mahasiswa.

No	Nama	Dosen Rata	SynSemSim		[4]		[3]		[2]	
			Rata	Deviasi	Rata	Deviasi	Rata	Deviasi	Rata	Deviasi
1	Afwanul	7,00	6,84	0,19	5,31	1,69	5,53	1,47	3,00	4,00
2	Rio	6,33	7,35	1,01	5,91	0,96	6,18	0,85	3,10	3,23
3	patrisius	6,00	6,85	0,99	5,33	0,68	5,53	0,90	3,10	2,90
4	Arya	7,33	6,70	0,64	5,34	1,99	5,43	1,90	3,10	4,23
5	Lena	7,00	7,02	0,90	5,61	1,39	5,82	1,18	3,10	3,90
6	Dading	5,67	7,06	2,24	5,41	2,24	5,54	2,53	3,10	2,57
7	Noor	5,67	6,88	2,04	5,34	1,87	5,53	2,37	3,10	2,57
8	Ahmad	5,67	7,40	2,61	5,99	1,85	6,17	2,43	3,10	2,57
9	Ferdi	7,00	7,17	1,14	5,60	1,80	5,85	1,16	3,10	3,90
10	Gunarto	6,67	7,41	1,13	5,95	0,75	5,83	1,00	3,10	3,57
11	Rianto	5,33	7,63	2,29	6,32	1,11	6,08	0,80	3,30	2,03
12	Hendrik	6,33	6,95	0,64	5,43	1,12	5,40	0,93	3,50	2,83
13	Holil	6,33	7,52	1,19	6,14	0,73	6,22	0,87	3,00	3,33
14	Husnun	5,33	7,39	2,06	6,01	0,90	6,27	1,07	2,80	2,53
15	Dewa	6,33	7,10	0,83	5,64	0,77	5,97	0,53	3,70	2,63
	AVG	6,27	7,15	1,33	5,69	1,32	5,82	1,33	3,15	3,12
	MIN	5,33	6,70	0,19	5,31	0,68	5,40	0,53	2,80	2,03
	MAX	7,33	7,63	2,61	6,32	2,24	6,27	2,53	3,70	4,23

Tabel 4. Contoh Hasil Uji Coba dengan Data Kuisisioner Tiap Jawaban Mahasiswa.

No	Nama	Jawaban Mahasiswa	Dosen SynSemSim	[4]	[3]	[2]
----	------	-------------------	-----------------	-----	-----	-----

		Nilai	Nilai	Deviasi	Nilai	Deviasi	Nilai	Deviasi	Nilai	Deviasi
1. Afwanul	1. <i>Software is an application created using programming languages for certain purposes.</i>	7,00	7,04	0,04	5,19	1,81	5,95	1,05	-	-
	2. <i>Hardware is electronic devices which have physical dimension.</i>	6,00	5,81	0,19	4,40	1,60	4,81	1,19	-	-
	3. <i>Information is Data that is processed to be another form which is more useful.</i>	8,00	7,66	0,34	6,33	1,67	5,82	2,18	-	-
	RATA	7,00	6,84	0,19	5,31	1,69	5,53	1,47	3,00	4,00
2. Rio	1. <i>Software is a program that runs in a computer.</i>	6,00	7,73	1,73	6,80	0,80	7,04	1,04	-	-
	2. <i>Hardware is a comprehensive form of all physical parts of a computer.</i>	6,00	6,84	0,84	5,06	0,94	5,30	0,70	-	-
	3. <i>Information is data that given from users before input in data processing.</i>	7,00	7,47	0,47	5,87	1,13	6,19	0,81	-	-
	RATA	6,33	7,35	1,01	5,91	0,96	6,18	0,85	3,10	3,23
3. Patrisius	1. <i>Software is an application created by programming languages for certain purposes.</i>	5,00	6,78	1,78	5,02	0,02	5,66	0,66	-	-
	2. <i>Hardware is a device that contains physical components, for example mouse, cpu, monitor, and keyboard.</i>	6,00	6,99	0,99	5,76	0,24	5,68	0,32	-	-
	3. <i>Information is processed data which can be understood on computer by user and can be useful for the real life.</i>	7,00	6,80	0,20	5,22	1,78	5,27	1,74	-	-
	RATA	6,00	6,85	0,99	5,33	0,68	5,53	0,90	3,10	2,90
4. arya	1. <i>Software is a device that run in computer, processes a given input, and usually called a device driver, which functions for computing data, for example: Microsoft Office.</i>	7,00	6,91	0,09	5,51	1,49	6,50	0,50	-	-
	2. <i>Hardware is a kind of computer's component which has physical form and can be seen by everyone.</i>	9,00	7,27	1,73	6,05	2,95	4,95	4,05	-	-
	3. <i>Information is something/data which is manipulated for many purposes.</i>	6,00	5,91	0,09	4,46	1,54	4,84	1,16	-	-
	RATA	7,33	6,70	0,64	5,34	1,99	5,43	1,90	3,10	4,23

Hasil uji coba dengan data sintesis ditunjukkan pada Tabel 2. Hasil uji coba dengan metode *SynSemSim* ini lebih mendekati definisi skenario daripada hasil penelitian sebelumnya [2][3].

Hasil uji coba dengan data kuisioner ditunjukkan pada Tabel 3 dan Tabel 4. Hasil yang diperoleh metode *SynSemSim* pada uji coba dengan data kuisioner relatif lebih tinggi daripada penelitian sebelumnya [2][3][4]. Hal ini dikarenakan sebagai berikut:

1. Metode *SynSemSim* mengambil nilai terbaik antara sintaktik dan semantik. Jika nilai kemiripan kata yang dihasilkan metode sintaktik lebih baik daripada semantik maka nilai sintaktik akan dipakai pada *matrix similarity (vice versa)*. Nilai kemiripan yang dihasilkan semakin tinggi berbanding lurus dengan banyak kosakata selain *noun, verb, adjective, adverb* yang dikandung oleh kalimat jawaban. Sebagai contoh kalimat pertama "*Information is something that we*

*can use for our needs*” dibandingkan dengan kalimat kedua *“Information is a result of an activity of data processing that able to give more valuable meaning to particular event”*. Maka nilai yang dihasilkan metode *SynSemSim* untuk kalimat kedua akan lebih tinggi dari kalimat pertama. Hal ini dikarenakan secara sintaktik kata *“Information”, “is”, “that”, “for”, “of”* memiliki nilai 1.0 sehingga metode sintaktik lebih dominan daripada metode *semantic* pada *matrix similarity*.

2. Metode *SynSemSim* mayoritas menghasilkan nilai kemiripan yang tinggi, bahkan pada kasus kalimat jawaban yang salah (seharusnya 0), nilai kemiripan yang dihasilkan metode *SynSemSim* tetap berada di rentang angka 5 hingga 7. Hal ini dikarenakan dua kalimat yang dibandingkan banyak mengandung kata *stopword* (kata yang tidak memiliki arti penting). Penggunaan kata *stopword* mengakibatkan perubahan pada struktur kalimat sehingga dua kalimat dengan struktur berbeda namun kosakata banyak yang sama menyebabkan metode sintaktik lebih dominan daripada metode semantik.

## SIMPULAN DAN SARAN

Dari penelitian yang telah dilakukan, maka dapat diambil beberapa simpulan:

1. Metode *SynSemSim* memiliki nilai rata deviasi untuk kasus terburuk mencapai 2,61 dan pada kasus terbaik nilai deviasi mencapai angka 0,19. Jika dibandingkan dengan hasil percobaan penelitian sistem penilaian otomatis sebelumnya [4] metode *SynSemSim* lebih mendekati kenyataan atau nilai manual dari Dosen seperti yang ditunjukkan Tabel 4.
2. Metode *SynSemSim* memiliki rata deviasi yaitu 1,33; yang berarti lebih baik dari penelitian sebelumnya [4], akan tetapi kurang lebih sama dengan dua penelitian lainnya [2][3] seperti yang ditunjukkan Tabel 3.
3. Kelebihan metode *SynSemSim* adalah membandingkan dua kalimat dengan struktur kalimat yang mirip. Sedangkan kelemahan dari metode *SynSemSim* adalah membandingkan kalimat majemuk atau kalimat yang banyak mengandung kata *stopword*.

Saran untuk penelitian selanjutnya, sistem penilaian otomatis dapat diintegrasikan langsung dengan perangkat lunak *e-Learning*. Perhitungan kemiripan akan lebih optimal bila perbandingan antar kata dilakukan di *level* struktur kalimat yang semakin dalam, terutama untuk kasus kalimat majemuk.

## DAFTAR PUSTAKA

- [1] Valenti S, Neri F and Cucchiarelli A. An overview of current research on automated essay grading. *Journal of Information Technology Education*. 2:319-330. 2003.
- [2] Dao TN. *An Improvement On Capturing Similarity Between String*. July 2005. URL: <http://www.codeproject.com/KB/recipes/improvestringsimilarity.aspx>, diakses tanggal 15 Juli 2009.
- [3] Dao TN and Simpson T. *Measuring Similarity Between Sentences*. October 2005. URL: <http://www.codeproject.com/KB/string/semanticssimilaritywordnet.aspx>, diakses tanggal 15 Juli 2009.
- [4] Firdausiah AB, Siahaan DO dan Yuhana UL. Sistem Penilaian Otomatis Jawaban Essay Menggunakan *Ontology* Pada *Moodle*. *TELKOMNIKA*. 6:167-172. 2008.