 ISBN: 978-602-60280-1-3

Penilaian Otomatis Uraian Singkat Menggunakan Metode *Semantic Text Similarity*

1st Uswatun Hasanah, 2nd Chyntia Raras Ajeng Widiawati
Program Studi Teknologi Informasi
Universitas Amikom Purwokerto
Purwokerto, Indonesia
1st uswatun_hasanah@amikompurwokerto.ac.id, 2nd chyntiaraw@amikompurwokerto.ac.id

Abstrak— Perkembangan penilaian otomatis uraian singkat telah memacu munculnya berbagai teknik dan metode komputasi untuk mengukur kemiripan jawaban siswa dengan jawaban referensi, baik secara leksikal maupun semantik. Pengukuran kemiripan secara leksikal hanya mengukur teks berdasarkan urutan karakternya, sedangkan pengukuran kemiripan secara semantik menilai kemiripan teks berdasarkan keterkaitan konteksnya. Penelitian ini menerapkan metode Semantic Text Similarity (STS) untuk menilai jawaban uraian singkat. Metode STS merupakan metode hybrid, yaitu gabungan antara metode berbasis string dan metode berbasis korpus. Pengukuran kemiripan secara string dilakukan dalam dua langkah. Langkah pertama menggunakan metode Longest Common Subsequence (LCS) yang dinormalisasi dan mendapat sedikit modifikasi. Langkah kedua menggunakan metode Common-Word Order dengan mempertimbangkan urutan kata pada kedua jawaban. Sedangkan pengukuran kemiripan secara semantik dilakukan dengan menggunakan metode Second Order Co-occurrence PMI (SOC-PMI). Sebanyak tiga assignment yang masing-masing terdiri dari tujuh pertanyaan dijawab oleh 30 siswa dan dinilai oleh dua *annotator*. Hasil menunjukkan metode STS mampu mencapai nilai korelasi Pearson antara 0,39264 - 0,57205 dan nilai *Mean Absolute Error* (MAE) antara 1,87944 - 2,13954. Dengan menggunakan dataset yang sama, metode STS memiliki tingkat *error* yang lebih kecil dari metode berbasis string *Cosine Coefficient*.

Kata kunci—kemiripan semantik, penilaian otomatis, short answer, Semantic Text Similarity

I. PENDAHULUAN

Soal bentuk uraian adalah alat penilaian yang menuntut siswa untuk mengingat, memahami, dan mengorganisasikan gagasan atau hal-hal yang sudah dipelajari, dengan cara mengemukakan atau mengekspresikan gagasan tersebut dalam bentuk uraian tertulis dengan menggunakan kata-kata pilihannya sendiri [1]. Tes uraian dapat dibedakan menjadi dua bentuk [1], yaitu: Tes uraian terbuka atau bebas, artinya butir soal yang ditanyakan hanya menyangkut masalah utama yang dibicarakan. Tes uraian tertutup atau terbatas atau terstruktur, artinya butir soal yang ditanyakan sudah mengarah ke masalah tertentu, sehingga jawaban siswa harus sesuai dengan apa yang dituntut dari soal itu secara terstruktur.

Beberapa keunggulan dari soal uraian adalah mampu mengukur aspek kognitif yang lebih tinggi, mengembangkan kemampuan berbahasa siswa, melatih kemampuan peserta didik untuk berpikir secara kritis, menghindari sifat terkaan dalam menjawab soal, dan mampu memberikan gambaran yang tepat pada bagian-bagian yang belum dikuasai peserta didik [1]. Sedangkan kelemahan dari soal uraian antara lain cara pemeriksaan hasil pekerjaan siswa lebih sulit dan cenderung subjektif, membutuhkan waktu yang lebih lama untuk koreksi, dan berpotensi untuk terjadi inkonsistensi penilaian.

Seiring berkembangnya teknologi informasi, penilaian hasil belajar mulai dilakukan dengan memanfaatkan teknologi e-Learning. Proses ini memungkinkan penilaian hasil belajar siswa secara otomatis, dengan menggunakan sistem penilaian otomatis. Sistem penilaian otomatis memiliki keunggulan, di antaranya mampu menilai jawaban dengan cepat, obyektif, dan konsisten. Burrows [2] memisahkan perbedaan antara pertanyaan jawaban pendek, *fill the gap*, dan esai dalam penilaian otomatis, berdasarkan panjang, fokus, dan sifat keterbukaan jawaban. Soal uraian singkat diidentifikasi memiliki panjang jawaban pada kisaran satu frase sampai satu paragraf, menuntut jawaban yang berfokus pada konten, serta memiliki sifat tertutup yang mana setiap jawaban memiliki suatu batasan tertentu.

Selama beberapa dekade terakhir, penelitian tentang penilaian otomatis uraian singkat menjadi topik yang cukup menarik. Penelitian tersebut berfokus tentang bagaimana membuat mesin, yang dalam hal ini adalah komputer, memungkinkan untuk menilai hasil belajar siswa yang berupa uraian singkat. Pada dasarnya, konsep penilaian otomatis uraian singkat adalah mengukur kemiripan jawaban siswa dengan jawaban kunci yang disediakan oleh guru. Dalam penelitiannya, Gomaa dan Fahmy [3] mengelompokkan pengukuran kemiripan kalimat ke dalam dua kelas utama, yaitu kemiripan leksikal dan kemiripan semantik. Dua kalimat dikatakan mirip secara leksikal apabila kedua kalimat memiliki urutan karakter yang sama. Sedangkan dua kalimat dikatakan mirip secara semantik apabila keduanya memiliki kata-kata pada suatu hal yang sama, saling bertentangan, digunakan dalam konteks yang sama, atau salah satu kata merupakan bagian dari kata yang lain.

Kemiripan semantik dapat dibagi menjadi dua kelompok metode, yaitu metode kemiripan berbasis korpus (*corpus-based similarity*) dan metode kemiripan berbasis pengetahuan (*knowedge-based similarity*). Kemiripan berbasis korpus mengukur kemiripan antara kata-kata menurut informasi yang diperoleh dari korpora besar. Sedangkan kemiripan berbasis pengetahuan adalah pengukuran kemiripan semantik dengan menggunakan informasi yang berasal dari jaringan semantik

kata. Pengukuran kemiripan secara leksikal cenderung mudah untuk dilakukan namun memiliki kelemahan karena karakter pada jawaban siswa harus sama persis dengan jawaban referensi yang disediakan oleh guru. Meskipun demikian, kemiripan leksikal juga memiliki peranan penting dalam proses penilaian otomatis uraian singkat.

Penelitian ini bertujuan untuk menerapkan metode *hybrid* pada sistem penilaian otomatis uraian singkat. Metode *hybrid* yang digunakan adalah metode *Semantic Text Similarity* (STS) yang merupakan gabungan dari metode berbasis korpus dan metode berbasis string. Dalam hal ini, penilaian otomatis uraian singkat diharapkan mampu memfasilitasi nilai kemiripan dari dua sisi, yaitu secara leksikal dan semantik.

II. TINJAUAN PUSTAKA

A. Penilaian Otomatis Uraian Singkat

Burrows [2] membagi sistem penilaian otomatis uraian singkat menjadi lima kelompok berdasarkan teknik yang digunakan yaitu *concept, information extraction, corpus-based, machine learning*, dan *evaluation era* (eval). Roy et al. [4] membagi teknik sistem penilaian otomatis uraian singkat menjadi lima kelompok utama, yaitu *Natural Language Processing* (NLP), *Information Extraction* dan *Pattern Matching, Machine Learning, Document Similarity*, dan *Clustering*. Masing-masing metode tersebut digunakan dan dikelompokkan berdasarkan permasalahan yang terjadi, solusi yang diusulkan, serta mengacu pada *dataset* yang digunakan.

Hasanah et al. [5] menggunakan metode berbasis string dan mempertimbangkan keberadaan kata kunci pada jawaban. Metode berbasis string yang digunakan adalah *Longest Common Subsequence, Cosine Coefficient, Jaccard Coefficient*, dan *Dice Coefficient*. Hasil korelasi *Pearson* diperoleh pada rentang 0,65 – 0,66 dan nilai *Mean Absolute Error* (MAE) pada rentang 0,95 – 1,24. Noorbehbahani dan Kardan [6] mengusulkan metode modifikasi dari algoritme BLEU dengan menyediakan beberapa jawaban referensi pada setiap pertanyaan. Hasil evaluasi menunjukkan bahwa metode yang diusulkan mampu mengungguli metode lain seperti *Latent Semantic Analysis* (LSA) dan *n-gram co-occurrence*. Mihalcea et al. [7] mengukur kemiripan semantik kata dengan menggunakan serangkaian metode dari kelompok *corpus-based similarity* seperti PMI-IR dan LSA, serta metode dari kelompok *knowldege-based similarity* seperti Leacock & Chodorow, Lesk, Wu & Palmer, Resnik, Lin, Jiang & Conrath. Melalui eksperimen yang dilakukan pada kumpulan data parafrase, metode pengukuran semantik mampu mengungguli metode kemiripan leksikal biasa, dan mampu menurunkan tingkat kesalahan sebanyak 13%.

B. Kemiripan Semantik

Keterkaitan semantik mengacu pada sejauh mana hubungan antara dua konsep atau kata, sedangkan kemiripan semantik merupakan kasus khusus atau subset dari keterkaitan semantik [8]. Manusia dapat dengan mudah menilai keterkaitan dari sepasang kata dengan berbagai cara. Budanitsky dan Hirst [9] menunjukkan bahwa kemiripan semantik digunakan ketika entitas serupa seperti apel dan jeruk atau meja dan furnitur dibandingkan. Berbeda dengan manusia, komputer adalah mesin sintaksis yang tidak dapat memahami semantik, sehingga komputer akan merepresentasikan kata-kata semantik sebagai kata-kata sintaksis [10]. Pengukuran kemiripan semantik kata telah digunakan sejak lama dalam aplikasi pengolahan bahasa alami dan bidang terkait lainnya, seperti pembuatan tesaurus otomatis, pengindeksan otomatis, anotasi teks dan peringkasan, klasifikasi teks, *word sense disambiguation*, ekstraksi informasi dan temu balik informasi, pemilihan leksikal, koreksi otomatis kesalahan kata pada teks, dan menemukan *word sense* dari suatu teks.

C. Latent Semantic Analysis

Latent Semantic Analysis (LSA) merupakan model asosiasi linier dimensi tinggi yang menganalisis korpus besar teks dan menghasilkan representasi yang menilai kemiripan kata atau bagian-bagian teks [11], [12]. Ketika LSA digunakan untuk menghitung kemiripan kalimat, vektor untuk setiap kalimat dibentuk dalam *reduced-dimensional space*; dan selanjutnya kemiripan diukur dengan menggunakan kosinus sudut antara vektor-vektor baris yang sesuai [13]. Ukuran dimensi kata dengan matriks konteks terbatas jumlahnya dan sudah ditetapkan menjadi beberapa ratus saja karena batasan komputasi pada *Singular Value Decomposition* (SVD). Sebagai hasilnya, vektor diperbaiki sehingga menjadi representasi teks pendek yang sangat jarang, seperti sebuah kalimat. LSA tidak memperhitungkan informasi sintaksis dan oleh karena itu lebih cocok untuk ukuran teks yang lebih panjang.

D. PMI-IR

PMI-IR [14], merupakan algoritma *unsupervised learning* sederhana untuk mengenali sinonim, dengan menggunakan *Pointwise Mutual Information* seperti yang dituliskan pada Persamaan (1) berikut:

$$\text{score}(choice_i) = \frac{\text{p}(problem \,\&\, choice_i)}{\text{p}(choice_i)} \quad (1)$$

$problem$ merepresentasikan kata yang bermasalah dan $\{choice_1, choice_2, \ldots, choice_n\}$ menyatakan alternatifnya. p($problem$ & $choice_i$) adalah probabilitas di mana $problem$ dan $choice_i$ bertemu. Dengan kata lain, setiap pilihan hanya dinilai dengan probabilitas bersyarat dari kata $problem$, yang ditunjukkan oleh kata $choice_i$, yaitu p($problem \mid choice_i$). Jika $problem$ dan $choice_i$ independen secara statistik, maka probabilitas keduanya terjadi bersamaan ditunjukkan oleh perkalian produk p($problem$) · p($choice_i$). Jika keduanya tidak independen dan memiliki kecenderungan untuk terjadi bersama, maka p($problem$ & $choice_i$) akan lebih besar dari p($problem$) · p($choice_i$).

PMI-IR menggunakan sintaks kueri *AltaVista Advanced Search* untuk menghitung probabilitas. Dalam kasus yang paling sederhana, dua kata muncul bersamaan ketika muncul dalam dokumen yang sama seperti pada Persamaan (2) berikut:

$$\text{score}_1(choice_i) = \frac{\text{hits}(problem \text{ AND } choice_i)}{\text{hits}(choice_i)} \quad (2)$$

hits(x) menjadi jumlah klik (jumlah dokumen yang diambil) ketika kueri x diberikan kepada AltaVista. AltaVista menyediakan berapa banyak dokumen yang mengandung $problem$ dan $choice_i$, dan berapa banyak dokumen yang hanya mengandung $choice_i$. Rasio kedua angka ini adalah skor untuk $choice_i$.

E. Semantic Text Similarity

Metode *Semantic Text Similarity* (STS) diusulkan oleh Islam dan Inkpen [15] dan merupakan metode *hybrid*, yaitu gabungan dari metode berbasis korpus dan metode berbasis string. Tiga fungsi kemiripan digunakan untuk mendapatkan

metode kemiripan yang lebih umum. Pertama, kemiripan secara string dan semantik diukur, dan selanjutnya fungsi kemiripan urutan kata (*common-word order similarity*) diterapkan untuk mendapatkan informasi sintaksis, namun penggunaannya dapat menjadi opsional. Berikut ini merupakan tiga fungsi kemiripan yang digunakan pada metode STS:

1) Kemiripan String Antar Kata

Metode STS mengukur kemiripan string antar kata dengan menggunakan metode *Longest Common Subsequence* (LCS) yang didasarkan pada penelitian yang dilakukan oleh [16] dengan beberapa normalisasi dan sedikit modifikasi. Metode LCS dinormalisasikan sehingga menjadi metode *Normalized Longest Common Subsequence* (NLCS) seperti yang ditunjukkan oleh persamaan (3) berikut:

$$v_1 = NLCS(r_i, s_j) = \frac{length(LCS(r_i,s_j))^2}{length(r_i)\times length(s_j)} \quad (3)$$

Sedangkan modifikasi NLCS ditunjukkan pada persamaan (4) dan (5) berikut:

$$v_2 = NMCLCS_1(r_i, s_j) = \frac{length(MCLCS_1(r_i,s_j))^2}{length(r_i)\times length(s_j)} \quad (4)$$

$$v_3 = NMCLCS_n(r_i, s_j) = \frac{length(MCLCS_n(r_i,s_j))^2}{length(r_i)\times length(s_j)} \quad (5)$$

Nilai v_1, v_2, dan v_3 diberi bobot (w_1, w_2, dan w_3) secara individu untuk menentukan nilai kemiripan, di mana w_1+w_2+$w_3 = 1$. Pada akhirnya, nilai kemiripan secara string ditentukan oleh persamaan (6) berikut:

$$\alpha = w_1 v_1 + w_2 v_2 + w_3 v_3 \quad (6)$$

2) Kemiripan Semantik Kata

Kemiripan semantik kata pada metode STS didasarkan pada metode SOC-PMI, yang merupakan hasil pengembangan dari metode PMI-IR. Misalkan W_1 dan W_2 adalah dua kata yang dibutuhkan untuk menentukan kemiripan semantik dan $C = \{c_1, c_2, \dots, c_m\}$ menyatakan suatu korpus teks yang besar (setelah beberapa *preprocessing*, misalnya penghapusan *stop word* dan lematisasi) yang mengandung kata-kata m (token). Selanjutnya, misalkan $T = \{t_1, t_2, \dots, t_n\}$ adalah himpunan semua kata unik yang muncul pada korpus C. Berbeda dengan korpus C yang merupakan daftar berurutan yang mengandung banyak kemunculan kata yang sama, T adalah himpunan yang tidak mengandung kata-kata yang diulang. Pada bagian ini, W digunakan untuk menyatakan W_1 dan W_2.

Parameter α ditetapkan, yang menentukan berapa banyak kata sebelum dan sesudah kata target W yang akan dimasukkan dalam jendela konteks. Jendela juga mengandung kata target W itu sendiri, menghasilkan *window size* $2\alpha + 1$ kata. Langkah-langkah dalam menentukan kemiripan semantik melibatkan pemindaian korpus dan kemudian mengekstraksi beberapa fungsi yang terkait dengan penghitungan frekuensi. Persamaan (7) berikut mendefinisikan fungsi frekuensi jenis:

$$f^t(t_i) = |\{k: c_k = t_i\}|, \qquad i = 1, 2, \dots, n \quad (7)$$

Fungsi frekuensi jenis menyatakan berapa banyak tipe t_i muncul pada seluruh korpus. Misalkan $f^b(t_i, W) = |\{k: t_k = W \text{ dan } t_{k\pm j} = t_i\}|$, di mana $i = 1,2, \dots, n$ dan $-\alpha \le j \le \alpha$, menjadi fungsi frekuensi bigram. $f^b(t_i, W)$ menyatakan berapa banyak kata t_i muncul dengan kata W dalam jendela dengan ukuran $2\alpha + 1$ kata. Kemudian fungsi *pointwise mutual information* dinyatakan pada Persamaan (8) untuk kata-kata yang memiliki $f^b(t_i, W) > 0$,

$$f^{pmi}(t_i, W) = \log_2 \frac{f^b(t_i,W)\times m}{f^t(t_i) f^t(W)} \quad (8)$$

Di mana $f^t(t_i) f^t(W) > 0$ dan m adalah total jumlah token pada korpus C seperti yang telah disebutkan sebelumnya. Untuk kata W_1, suatu himpunan kata X didefinisikan, diurutkan secara menurun (*descending*) berdasarkan nilai PMI dengan W_1 dan mengambil kata-kata teratas β_1 yang memiliki $f^{pmi}(t_i, W_1) > 0$.

$X = \{X_i\}$, di mana $i = 1, 2, \dots, \beta_1$ dan $f^{pmi}(t_1, W_1) \ge f^{pmi}(t_2, W_1) \ge \cdots f^{pmi}(t_{\beta_1 - 1}, W_1) \ge f^{pmi}(t_{\beta_1}, W_1)$.

Demikian juga untuk kata W_2, suatu himpunan kata Y didefinisikan pada urutan menurun berdasarkan nilai PMI dengan W_2 dan mengambil kata-kata teratas β_2 yang memiliki $f^{pmi}(t_i, W_2) > 0$.

$Y = \{Y_i\}$, di mana $i = 1, 2, \dots, \beta_2$ dan $f^{pmi}(t_1, W_2) \ge f^{pmi}(t_2, W_2) \ge \cdots f^{pmi}(t_{\beta_2 - 1}, W_2) \ge f^{pmi}(t_{\beta_2}, W_2)$.

Nilai untuk β (baik untuk β_1 dan β_2) belum ditentukan, yang mana nilainya bergantung pada kata W dan jumlah jenis kata pada korpus. Selanjutnya, fungsi $\beta - PMI\ summation$ didefinisikan. Untuk kata W_1, fungsi $\beta - PMI\ summation$ ditunjukkan oleh Persamaan (9) berikut:

$$f^{\beta}(W_1) = \sum_{i=1}^{\beta_1} (f^{pmi}(X_i, W_2))^{\gamma} \quad (9)$$

Di mana $f^{pmi}(X_i, W_2) > 0$ dan $f^{pmi}(X_i, W_1) > 0$ yang menjumlahkan seluruh nilai PMI positif kata-kata pada himpunan Y juga untuk kata-kata yang sama pada himpunan X. Fungsi ini bersifat *semantically-close* karena semua kata ini memiliki nilai PMI tinggi dengan W_2 dan hal ini tidak menjamin kedekatan sehubungan dengan jarak dalam ukuran jendela. Begitu juga untuk kata W_2, fungsi $\beta - PMI\ summation$ ditunjukkan oleh Persamaan (10) berikut:

$$f^{\beta}(W_2) = \sum_{i=1}^{\beta_2} (f^{pmi}(Y_i, W_1))^{\gamma} \quad (10)$$

Di mana $f^{pmi}(Y_i, W_1) > 0$ dan $f^{pmi}(Y_i, W_2) > 0$ yang menjumlahkan seluruh nilai PMI positif kata-kata pada himpunan X juga untuk kata-kata yang sama pada himpunan Y. Fungsi kemiripan semantik PMI atara dua kata W_1 dan W_2 didefinisikan pada Persamaan (11) berikut:

$$Sim(W_1, W_2) = \frac{f^{\beta}(W_1)}{\beta_1} + \frac{f^{\beta}(W_2)}{\beta_2} \quad (11)$$

Nilai β terkait dengan berapa kali kata W muncul dalam korpus. Nilai β didefinisikan dalam persamaan (12) berikut:

$$\beta_1 = (\log\ (f^t(W_i)))^{2\,\frac{(\log_2(n))}{\delta}}, \quad i = 1,2 \quad (12)$$

Di mana δ adalah konstanta dan pada penelitian yang dilakukan oleh Islam dan Inkpen [8] nilai $\delta = 6{,}5$. Nilai δ bergantung pada ukuran korpus. Semakin kecil korpus yang digunakan, maka nilai δ juga semakin kecil. Jika nilai β diturunkan, maka konsekuensinya akan ada beberapa kata penting yang hilang, dan jika nilai β ditingkatkan maka kemungkinan akan lebih banyak kata yang sama pada W_1 dan W_2 sehingga akan menurunkan hasil kemiripan secara signifikan. γ harus memiliki nilai lebih besar dari 1. Semakin tinggi nilai γ maka semakin besar titik berat pada kata-kata yang memiliki nilai PMI sangat tinggi dengan W. Pada penelitiannya, Islam dan Inkpen [8] memilih nilai $\gamma = 3$.

3) Kemiripan Urutan Kata pada Kalimat

Jika dua teks memiliki beberapa kata yang sama, maka urutan kata di dalamnya dapat diukur. Kata-kata ini dapat muncul dalam urutan yang sama, atau urutan yang hampir sama, atau urutan yang sangat berbeda. Misalkan ada sepasang kalimat P dan R, yang masing-masing memiliki token sejumlah m dan n. Hitung jumlah p_i (sebut saja δ) untuk $p_i = r_j$, untuk semua $p \in P$ dan $r \in R$. Sehingga, terdapat δ token pada P yang sama persis dengan R, di mana $\delta \leq m$. Hapus semua δ token pada P dan letakkan pada X dan hapus semua δ token pada R dan letakkan pada Y, dengan urutan posisi yang sama seperti saat token muncul pada kalimat. Sehingga, $X = \{x_1, x_2, \dots, x_\delta\}$ dan $Y = \{y_1, y_2, \dots, y_\delta\}$. Selanjutnya, ubah X dengan memasukkan indeks angka unik untuk setiap token pada X dimulai dari 1 sampai δ, sehingga, $X = \{1,2, \dots, \delta\}$. Berdasarkan indeks angka unik untuk setiap token pada X, ubah juga Y d mana $X = Y$. Selanjutnya pengukuran kemiripan urutan kata umum (common-word order similarity) dari dua teks ditunjukkan oleh persamaan (13) berikut:

$$S_o = 1 - \frac{|x_1 - y_1| + |x_2 - y_2| + \cdots + |x_\delta - y_\delta|}{|x_1 - x_\delta| + |x_2 - x_{\delta-1}| + \cdots + |x_\delta - x_1|} \quad (13)$$

Keseluruhan fungsi kemiripan pada metode STS akan menghasilkan kemiripan gabungan yang ditunjukkan oleh Persamaan (14) berikut:

$$S(P,R) = \frac{(\delta(1 - w_f + w_f S_o) + \sum_{i=1}^{|\rho|} \rho_i) \times (m+n)}{2mn} \quad (14)$$

Jika urutan kemiripan kata diabaikan, maka atur $w_f = 0$.

III. METODOLOGI PENELITIAN

A. Alat dan Bahan

Penelitian dilakukan dengan menggunakan beberapa alat yang terdiri dari perangkat lunak dan perangkat keras komputer sebagai berikut:

- Perangkat keras: Laptop dengan spesifikasi prosesor intel core i-3, RAM 6 GB, dan Sistem Operasi Windows 8.1 64-bit sebagai media untuk pengumpulan data, programming, pengolahan hasil, dan penulisan laporan.
- Perangkat lunak: Bahasa pemrograman Python versi 3.6.0 sebagai alat untuk menghitung nilai *similarity*, Spyder IDE versi 2.3.7 digunakan sebagai editor bahasa pemrograman Python, aplikasi pengolah kata Notepad++ sebagai wadah pengolahan dataset, dan aplikasi *spreadsheet* Microsoft Office Excel 2010 sebagai media untuk mengolah hasil eksperimen dan hasil pengujian.

Sedangkan bahan penelitian yang digunakan dalam penelitian ini adalah dataset berdasarkan penelitian yang dilakukan oleh [17]. Dataset tersebut terdiri dari tiga penugasan (*assignment*) yang terdiri dari 7 pertanyaan pada masing-masing penugasan, dan diberikan kepada kelas untuk matakuliah pengantar ilmu komputer di University of North Texas. Data berbentuk *plaintext*, dan setiap tugas mencakup pertanyaan, jawaban guru dan sekumpulan jawaban siswa. Masing-masing jawaban siswa telah dinilai oleh dua guru dan rata-rata nilai *annotator* juga disertakan. Korelasi *inter-annotator* adalah 0,6443 menggunakan koefisien Pearson. Gambar 1 berikut merupakan contoh potongan data yang digunakan:

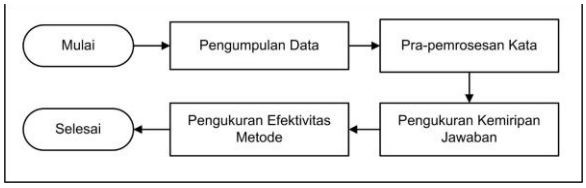
```
##################################################
##################################################
###################
        Question: What is the role of a prototype
        program in problem solving?
        Answer: To simulate the behaviour of
        portions of the desired software product.

3.5 [6] High risk problems are address in the
prototype program to make sure that the program is
feasible.  A prototype may also be used to show a
company that the software can be possibly
programmed.<br><br>
5   [5] To simulate portions of the desired final
product with a quick and easy program that does a
small specific job. It is a way to help see what the
problem is and how you may solve it in the final
project.
4   [8] A prototype program simulates the behaviors
of portions of the desired software product to allow
for error checking.
```

Gambar 1 Contoh potongan data

B. Metode Penelitian

Setelah mengumpulkan data, langkah-langkah penelitian selanjutnya ditunjukkan oleh Gambar 2 berikut:



Gambar 2 Alur penelitian

Data soal dan jawaban yang telah dikumpulkan kemudian melalui proses pra-pemrosesan kata. Teknik pra-pemrosesan yang digunakan adalah membuang seluruh angka dan simbol dari dataset. Selanjutnya *case folding* digunakan untuk mengubah semua huruf menjadi huruf kecil. *Stopwords removal* digunakan untuk membuang *stoplist* atau daftar kata-kata yang frekuensi kemunculannya cukup banyak namun tidak memberi dampak yang signifikan pada pengukuran kemiripan kalimat. Penelitian ini menggunakan 179 kata yang termasuk dalam *stoplist*. Setelah melalui tahap pra-pemrosesan teks, jawaban siswa diukur kemiripannya dengan jawaban guru menggunakan metode STS. Setelah semua jawaban dinilai, pengukuran efektivitas metode dilakukan dengan menghitung nilai korelasi Pearson (r) dan *Mean Absolute Error* (MAE). Pada tahap ini, metode pengukuran kemiripan berbasis string *Cosine Coefficient* juga digunakan sebagai metode pembanding.

IV. HASIL DAN PEMBAHASAN

Pada bagian ini akan dibahas mengenai contoh penggunaan metode STS pada penilaian otomatis uraian singkat. Terdapat jawaban referensi yang disediakan oleh guru (P) dan jawaban dari siswa (R).

P = To simulate the behaviour of portions of the desired software product.

R = it simulates the behavior of portions of the desired software product

Langkah 1: setelah melalui tahapan pra-pemrosesan teks, maka akan didapatkan P = {*simulate, behaviour, portion, desired, software, product*} dan R = {*simulates, behavior, portion, desired, software, product*} di mana $m = 6$ dan $n = 6$.

Langkah 2: terdapat empat token pada *P* (*portion, desired, software, product*) yang juga sama persis dengan token pada *R*, sehingga nilai $\delta = 4$. Selanjutnya, eliminasi token portion, desired, software, product dari *P* dan *R*. Sehingga *P* = {*simulate, behaviour*} dan *R* = {*simulates, behavior*). Karena nilai $m - \delta \neq 0$, maka lakukan proses selanjutnya.

Langkah 3: buat matriks pencocokan string (M_1) berukuran 2×2.

$$M_1 = \begin{matrix} \\ simulate \\ behaviour \end{matrix} \begin{matrix} simulates & behavior \\ \end{matrix} \begin{bmatrix} 0{,}88000 & 0.00516 \\ 0{,}01630 & 0.74250 \end{bmatrix}$$

Langkah 4: buat matriks kemiripan semantik (M_2), dan atur $\lambda = 25$ untuk menghitung nilai SOC-PMI untuk tiap pasang kata.

$$M_2 = \begin{matrix} \\ simulate \\ behaviour \end{matrix} \begin{matrix} simulates & behavior \\ \end{matrix} \begin{bmatrix} 1{,}00000 & 0{,}16667 \\ 0{,}00000 & 0{,}30769 \end{bmatrix}$$

Langkah 5: buat matriks gabungan (M) dan masukkan faktor bobot $\psi = 0{,}5$ dan $\varphi = 0{,}5$.

$$M = \begin{matrix} \\ simulate \\ behaviour \end{matrix} \begin{matrix} simulates & behavior \\ \end{matrix} \begin{bmatrix} \mathbf{0,94000} & 0{,}08591 \\ 0{,}00815 & 0{,}52510 \end{bmatrix}$$

Dapat diketahui bahwa elemen matriks nilai tertinggi, $\gamma_{ij} = 0{,}94000$ dan tambahkan nilai tersebut untuk ρ karena $\gamma_{ij} \geq 0$. Matriks *M* yang baru setelah mengeliminasi baris ke-i ($i = 1$) dan kolom ke-j ($j = 1$) adalah:

$$M = \begin{matrix} \\ behaviour \end{matrix} \begin{matrix} behavior \\ \end{matrix} \begin{bmatrix} \mathbf{0, 52510} \end{bmatrix}$$

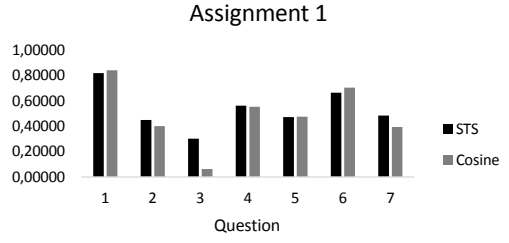
Diketahui elemen matriks nilai tertinggi selanjutnya $\gamma_{ij} = 0{,}52510$ dan tambahkan nilai tersebut untuk ρ karena $\gamma_{ij} \geq 0$, sehingga $\rho = \{0{,}94000, 0{,}52510\}$.

Langkah 6: dari langkah sebelumnya diketahui bahwa $m = 6, n = 6, \delta = 4$, dan $|\rho| = 2$.

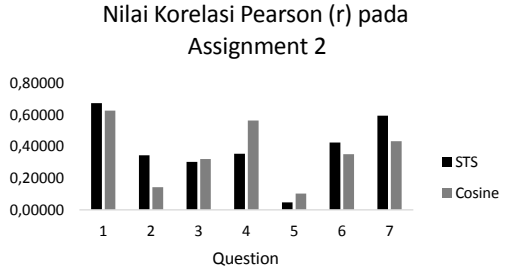
$$S(P,R) = \frac{(\delta + \sum_{i=1}^{|\rho|} \rho_i) \times (m+n)}{2mn}$$

$$= \frac{(4+1{,}4651) \times 12}{72} = 0{,}91085$$

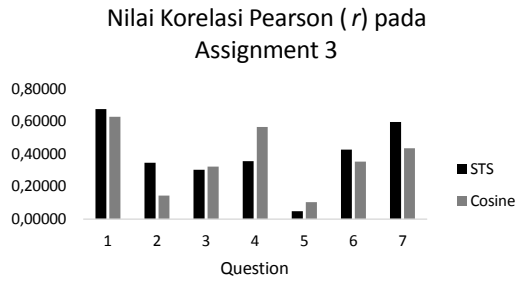
Untuk contoh ini, tidak menjadi masalah berapa pun nilai yang dipilih untuk *w* (faktor *common-word order*), karena nilai kemiripan *common-word order* adalah 1. Sehingga, $S(P,R) = 0{,}91085$ untuk semua $w_f \in [0, 0{,}5)$. Selanjutnya, langkah 1 sampai dengan langkah 6 diterapkan pada seluruh *dataset*. Sebanyak 3 *assignment* diukur kemiripannya menggunakan metode STS. Selain itu, metode *cosine coefficient* juga digunakan untuk menghitung nilai kemiripan jawaban siswa dan jawaban referensi. Gambar 3, Gambar 4, Gambar 5, Gambar 6, Gambar 7 dan Gambar 8 berikut menunjukkan perbandingan nilai korelasi dan MAE pada ketiga *assignment*.



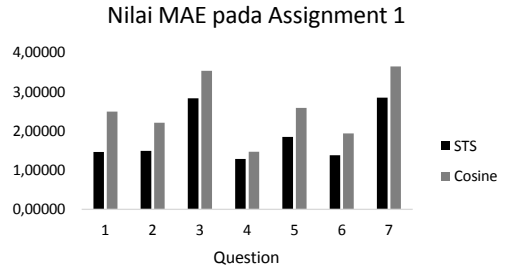
Gambar 3 Nilai Korelasi Pearson pada *Assignment* 1



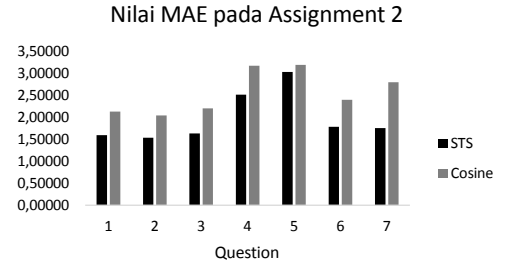
Gambar 4 Nilai Korelasi Pearson pada *Assignment* 2



Gambar 5 Nilai Korelasi Pearson pada *Assignment* 3



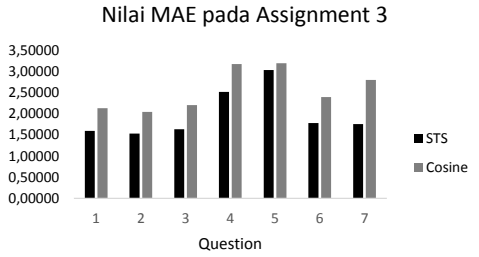
Gambar 6 Nilai MAE pada *Assignment* 1



Gambar 7 Nilai MAE pada *Assignment* 2

Tabel 1 menunjukkan rata-rata nilai korelasi Pearson (*r*) dan MAE pada ketiga *assignment*. Hasil menunjukkan bahwa dalam beberapa pertanyaan metode STS mampu mencapai nilai korelasi yang lebih tinggi dibanding metode *cosine coefficient*, meskipun pada beberapa pertanyaan nilai korelasi metode STS tidak lebih unggul dari *cosine coefficient*.

Namun, secara umum metode STS memiliki tingkat *error* yang lebih kecil.



Gambar 8 Nilai MAE pada *Assignment* 3

Tabel 1. rata-rata Nilai Korelasi Pearson dan MAE Keterkaitan konteks pada pengukuran kemiripan semantik

Assignment	***r***		***MAE***	
	STS	***Cosine***	***STS***	***Cosine***
1	0,53714	0,49075	1,87944	2,55668
2	0,39264	0,36428	1,97866	2,56317
3	0,57205	0,57704	2,13954	2,87473

kalimat dapat menjadi langkah yang menjanjikan dalam pengembangan model sistem penilaian otomatis uraian singkat. Meskipun demikian, metode SOC-PMI yang digunakan pada penelitian ini belum mampu menangani makna ekspresi yang digunakan untuk suatu maksud tertentu. Sebagai contoh, pada pertanyaan nomor 5 untuk *assignment* 2, jawaban yang ditargetkan oleh guru adalah "*unlimited number*" di mana siswa membuat ekspresi yang berbeda dalam menjawab, meskipun pada dasarnya makna tersebut sama. Sebagai contoh, ketika siswa menjawab dengan "*any number you want*" atau "*as many as you need*", maka sebenarnya jawaban tersebut memiliki makna yang sama dengan "*unlimited number*". Namun metode SOC-PMI tidak akan memberikan nilai yang memuaskan karena ada kemungkinan bahwa kata-kata tersebut tidak muncul dalam jendela konteks yang sama di dalam korpus.

V. KESIMPULAN

Metode STS yang digunakan pada penelitian ini mampu mengukur kemiripan leksikal dan semantik pada data jawaban uraian singkat dengan nilai korelasi Pearson antara 0,39264 - 0,57205 dan nilai *Mean Absolute Error* (MAE) antara 1,87944 - 2,13954. Dengan menggunakan dataset yang sama, metode STS memiliki tingkat *error* yang lebih kecil dari metode berbasis string *Cosine Coefficient*. Meskipun demikian, metode SOC-PMI yang digunakan sebagai metode pengukuran semantik kata belum mampu menangani eskpresi jawaban yang berbeda meskipun jawaban tersebut memiliki maksud/arti yang sama. Dalam hal ini, kesalahan dapat terjadi apabila kata-kata yang digunaan tidak muncul bersama dalam jendela konteks yang sama di dalam korpus.

DAFTAR PUSTAKA

[1] D. Kunandar, "Penilaian Autentik (Penilaian Hasil Belajar Peserta Didik Berdasarkan Kurikulum 2013)," *Jakarta Rajawali Pers*, 2013.

[2] S. Burrows, I. Gurevych, and B. Stein, "The eras and trends of automatic short answer grading," *Int. J. Artif. Intell. Educ.*, vol. 25, no. 1, pp. 60–117, 2015.

[3] W. H.Gomaa and A. A. Fahmy, "A Survey of Text Similarity Approaches," *Int. J. Comput. Appl.*, vol. 68, no. 13, pp. 13–18, 2013.

[4] S. Roy, Y. Narahari, and O. D. Deskhmukh, "A perspective on computer assisted assessment techniques for short free-text answers.," in *International Computer Assisted Assessment Conference*, 2015, vol. 571, pp. 96–109.

[5] U. Hasanah, A. E. Permanasari, and S. S. Kusumawardani, "A scoring rubric for automatic short answer grading system," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 17, no. 2, pp. 763–770, 2019.

[6] F. Noorbehbahani and A. A. Kardan, "The automatic assessment of free text answers using a modified BLEU algorithm," *Comput. Educ.*, vol. 56, no. 2, pp. 337–345, 2011.

[7] A. M. Ben Omran and M. J. Ab Aziz, "Automatic essay grading system for short answers in English language," in *Journal of Computer Science*, 2013, Third Edit., vol. 9, no. 10, pp. 1369–1382.

[8] A. Islam and D. Inkpen, "Second order co-occurrence PMI for determining the semantic similarity of words," in *LREC*, 2006, pp. 1033–1038.

[9] A. Budanitsky and G. Hirst, "Evaluating WordNet-based Measures of Lexical Semantic Relatedness," *Comput. Linguist.*, vol. 32, no. 1, 2006.

[10] A. Maind, A. Deorankar, and C. Prashant, "Measurement of semantic similarity between words: A survey," *Int. J. Comput. Sci. Eng. Inf. Technol.*, vol. 2, no. 6, pp. 51–56, 2012.

[11] T. K. Landauer and S. T. Dumais, "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge.," *Psychol. Rev.*, vol. 104, no. 2, p. 211, 1997.

[12] S. Dennis, T. Landauer, W. Kintsch, and J. Quesada, "Introduction to latent semantic analysis," in *25th Annual Meeting of the Cognitive Science Society. Boston, Mass*, 2003.

[13] P. W. Foltz, W. Kintsch, and T. K. Landauer, "The measurement of textual coherence with latent semantic analysis," *Discourse Process.*, vol. 25, no. 2–3, pp. 285–307, 1998.

[14] P. D. Turney, "Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL," in *Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001)*, 2001, pp. 491–502.

[15] A. Islam and D. Inkpen, "Semantic Text Similarity Using Corpus-Based Word Similarity and String Similarity," *ACM Trans. Knowl. Discov. Data*, vol. 2, no. 2, pp. 1–25, 2008.

[16] L. Allison and T. I. Dix, "A bit-string longest-common-subsequence algorithm," *Inf. Process. Lett.*, vol. 23, no. 5, pp. 305–310, 1986.

[17] M. Mohler and R. Mihalcea, "Text-to-text Semantic Similarity for Automatic Short Answer Grading," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL '09)*, 2009, no. April, pp. 567–575.