

Pengukuran Keserupaan pada *Communication Diagram* menggunakan Metode *Graph Edit Distance*

Hidayatul Munawaroh^a, Rizqi Amrullah Wildan Yani^b, Liris Aditya Ningsih^c

^aDepartemen Sistem Informasi, Universitas Internasional Semen Indonesia, hidayatul.munawaroh@uisi.ac.id

^bDepartemen Sistem Informasi, Universitas Internasional Semen Indonesia, rizqi.yani19@student.uisi.ac.id

^cDepartemen Sistem Informasi, Universitas Internasional Semen Indonesia, liris.ningsih19@student.uisi.ac.id

Submitted: 19-06-2023, Reviewed: 12-07-2023, Accepted 01-08-2023
<https://doi.org/10.47233/jteksis.v5i4.900>

Abstract

E-learning is an electronic learning system that is created as a learning media so that the delivery of learning material and assessment of answers to practice questions and exams can be done online. Learning with e-learning method provides opportunities for students to take part in learning at flexible locations and times. The e-learning model is widely used in learning, so it is necessary to increase the need to automate the evaluation process in e-learning. In practice, some learning topics include descriptive answers or pictures in the form of Unified Modeling Language (UML) diagrams such as exam questions on Software Requirements Engineering (SRE). One of the diagrams used is communication diagram, so it is necessary to develop a method for evaluating answers in the form of a communication diagram. This study proposes graph modeling on communication diagrams for measuring the similarity between two diagrams and measuring the similarity between communication diagrams. The communication diagram is modeled into a graph before its similarity is measured. Furthermore, the measurement of the similarity between the two representational graphs is carried out based on structural aspects which are divided into two components, namely inter-structural and intra-structural. Each component is measured for similarity using the Graph Edit Distance method. The GED calculation is performed to find the minimum cost to transform the initial graph to the desired graph. From the results of this study, it is proven that the similarity of communication diagrams can be measured by the Graph Edit Distance method by modeling the diagram into the previous graph form.

Keywords: *Communication Diagram, E-learning, Graph Edit Distance*

Abstrak

E-learning merupakan sistem pembelajaran elektronik yang dibuat sebagai media belajar sehingga penyampaian materi pembelajaran dan penilaian jawaban soal latihan maupun ujian secara online. Pembelajaran dengan metode e-learning memberikan kesempatan bagi peserta untuk mengikuti pembelajaran di lokasi dan waktu yang fleksibel. Model e-learning banyak digunakan pada pembelajaran, sehingga diperlukan peningkatan kebutuhan untuk mengotomatisasi proses evaluasi pada e-learning. Dalam penerapannya, beberapa topik pembelajaran menyertakan jawaban deskriptif atau gambar dalam bentuk diagram Unified Modeling Language (UML) seperti soal pada Rekayasa Kebutuhan Perangkat Lunak (RKPL). Salah satu diagram yang digunakan adalah communication diagram, sehingga diperlukan pengembangan metode evaluasi jawaban dalam bentuk communication diagram. Penelitian ini mengajukan pemodelan graf pada communication diagram untuk pengukuran keserupaan antara dua diagram, dan melakukan pengukuran keserupaan antar communication diagram. Communication diagram dimodelkan kedalam graf sebelum diukur keserupaannya. Selanjutnya pengukuran keserupaan antara dua graf representasi communication diagram dilakukan berdasarkan aspek struktural yang dibagi kedalam dua komponen yaitu inter-struktur dan intra-struktur. Setiap komponen diukur keserupaannya menggunakan metode Graph Edit Distance. Perhitungan GED dilakukan untuk mencari cost perubahan terkecil yang perlu dilakukan untuk merubah graf awal ke graf tujuan. Dari hasil penelitian ini terbukti bahwa keserupaan communication diagram dapat diukur dengan metode Graph Edit Distance dengan memodelkan diagram kedalam bentuk graf sebelumnya.

Keywords: *Communication Diagram, E-learning, Graph Edit Distance*

This work is licensed under Creative Commons Attribution License 4.0 CC-BY International license



PENDAHULUAN

Seiring dengan perkembangan teknologi, proses pembelajaran tidak hanya dilakukan pada ruang kelas dan dalam satu waktu saja. Saat ini, *e-learning* menjadi salah satu komponen penting dalam Pendidikan tinggi dan terus meningkat penggunaannya. *E-learning* merupakan media pendukung dalam penyampaian dan evaluasi pembelajaran secara elektronik [1]. Penggunaan *e-learning* memerlukan batasan ruang atau waktu

yang lebih sedikit, dan lebih mudah dalam membuat catatan jika dibandingkan dengan ruang kelas tradisional.

Penilaian merupakan proses dalam memberikan nilai terhadap suatu obyek dengan mengacu pada ukuran tertentu, seperti tinggi atau rendah. Sedangkan pengukuran merupakan proses pemberian angka terhadap objek ukur. Pengukuran dilakukan dengan membandingkan objek yang ingin diukur dengan fakta yang ada [2]. Pengukuran selalu

memberikan hasil kuantitatif sedangkan hasil penilaian dapat bersifat kualitatif. Hasil pengukuran dapat digunakan sebagai pembandingan atau dasar dalam penilaian.

Pembelajaran dengan metode e-learning memberikan kesempatan bagi peserta didik untuk mengikuti pembelajaran di lokasi dan waktu yang fleksibel. Jumlah peserta yang dapat mengikuti pembelajaran menjadi lebih banyak bila dibandingkan dengan kelas konvensional, sehingga diperlukan peningkatan kebutuhan untuk mengotomatisasi proses evaluasi pada *e-learning*. Sistem penilaian pada *e-learning* dibagi menjadi dua jenis, yaitu penilaian jawaban yang dipilih (*selected response*) dan penilaian jawaban yang dibangun (*constructed response*) [3]. *Selected response* meliputi jawaban benar/salah, pencocokan, pilihan ganda, dan daftar pilihan jawaban. *Constructed response* meliputi jawaban isian dan uraian [4], [5]. Jawaban uraian merupakan jawaban yang dituliskan dalam bentuk uraian, berbeda dengan soal uraian. Soal uraian adalah soal yang dituliskan dalam bentuk cerita atau uraian dengan tipe jawaban yang variatif sesuai kebutuhan, bisa dalam bentuk *selected response* atau *constructed response*. Otomatisasi pada *constructed response* yang sudah ada hingga saat ini adalah untuk jawaban isian teks. Penilaian jawaban isian teks yang ada sudah reliabel terhadap penilaian dari pengampu mata pelajaran [6]–[8].

Dalam penerapannya, beberapa topik pembelajaran menyertakan jawaban deskriptif atau gambar dalam bentuk diagram *Unified Modeling Language* (UML) seperti soal ujian pada Rekayasa Kebutuhan Perangkat Lunak (RKPL). UML merupakan *tools* yang dapat digunakan untuk membantu perancangan perangkat lunak [9]–[11] dan digunakan dalam pembelajaran. Sehingga diperlukan pengembangan metode evaluasi jawaban dalam bentuk diagram UML. Penelitian sebelumnya telah mengembangkan metode pengukuran keserupaan diagram UML untuk penilaian jawaban uraian dalam bentuk diagram. Diagram UML yang sudah ditangani adalah *class diagram* [12], *sequence diagram* [13], *use case diagram* [14], dan *statechart diagram* [15] yang menggunakan *Finite State Machine* dan *Cuckoo Search Algorithm* (CSA) untuk mencocokkan dan mengambil *statechart diagram* dari repositori. Hasil eksperimen menunjukkan bahwa metode ini efektif untuk mengambil *statechart diagram* paling mirip dari repositori terhadap kueri yang diberikan. Penelitian lain tentang keserupaan *statechart diagram* juga telah dilakukan dengan tujuan penggunaan kembali rancangan perangkat lunak [16]. Sedangkan belum ada pengukuran keserupaan untuk *communication diagram*.

Pada penelitian sebelumnya sudah terdapat beberapa metode untuk menghitung keserupaan diagram UML. Tetapi fokus utama dari metode-metode tersebut adalah untuk penggunaan kembali rancangan perangkat lunak atau hanya memperhatikan salah satu aspek baik semantik [12]–[14], [17]–[19] ataupun struktural [16], [20]. Pengukuran keserupaan diagram UML dengan tujuan penggunaan kembali rancangan perangkat lunak dilakukan dengan membandingkan satu diagram UML dengan beberapa diagram UML yang ada pada repositori [15], [21], [22]. Meskipun perangkat lunak memiliki domain yang berbeda, rancangan perangkat lunak dapat digunakan kembali untuk pengembangan perangkat lunak baru. Peringkat keserupaan ditentukan dari nilai keserupaan diagram yang didapatkan. Sedangkan pengukuran keserupaan diagram UML untuk penilaian jawaban soal dilakukan dengan membandingkan beberapa diagram jawaban dengan diagram kunci. Diagram jawaban dan kunci harus memiliki domain yang sama dan penilaian yang diberikan belum tentu sama persis dengan nilai keserupaan yang didapatkan. Sehingga pengaruh dari aspek-aspek pengukuran keserupaan diagram UML akan berbeda bergantung tujuan yang ingin dicapai.

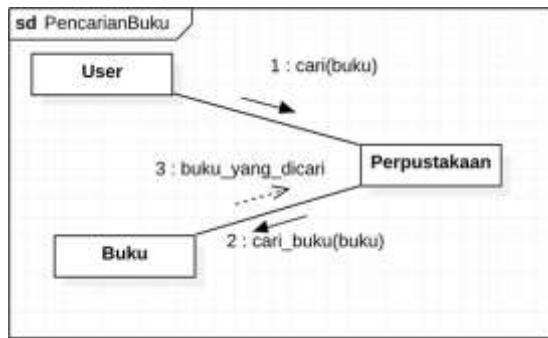
Penelitian ini mengajukan pemodelan graf pada *communication diagram* untuk pengukuran keserupaan antara dua buah diagram, dan melakukan pengukuran keserupaan antar *communication diagram*. Pengukuran keserupaan *communication diagram* dilakukan dengan memperhatikan aspek struktural dengan tujuan untuk penilaian jawaban soal. Hasil pengukuran keserupaan diagram jawaban dan diagram kunci dapat dijadikan landasan atau pertimbangan bagi pengampu mata pelajaran untuk memberikan nilai.

METODE PENELITIAN

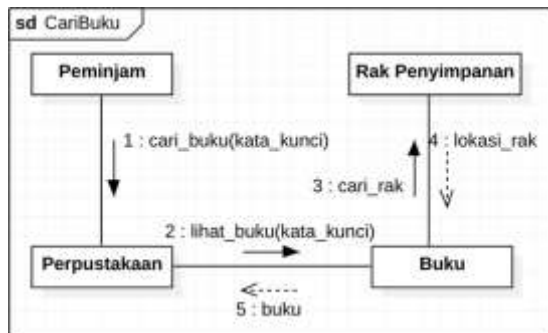
Beberapa tahapan yang dilakukan pada penelitian ini adalah studi literatur, pengumpulan data *communication diagram*, pengukuran keserupaan antara dua *communication diagram*, hingga pelaporan hasil akhir.

1) Studi Literatur

Studi literatur dilakukan melalui jurnal, buku, dan berbagai sumber lainnya untuk mencari informasi tentang perkembangan penelitian pada ranah pengukuran keserupaan diagram UML. Tahapan ini dilakukan untuk mengetahui dan memahami metode-metode yang telah diterapkan sebelumnya. Studi literatur dilakukan dengan meninjau topik yang berkaitan dengan karakteristik diagram UML, pemodelan diagram UML menjadi bentuk graf, dan metode untuk mengukur keserupaan antara dua diagram UML.



Gambar 1 Contoh Communication Diagram 1 (CD1)



Gambar 2 Contoh Communication Diagram 2 (CD2)

2) Pengumpulan Data Communication Diagram

Data yang dikumpulkan merupakan rancangan perangkat lunak dari deskripsi perangkat lunak yang diberikan. Rancangan perangkat lunak dibuat oleh orang yang sebelumnya telah diberikan pengetahuan tentang *communication diagram*. Rancangan tersebut nantinya akan dibandingkan dan diukur keserupaannya dengan kunci jawaban rancangan yang telah dibuat sebelumnya.

Sebelum dilakukan pengukuran keserupaan, setiap diagram diolah menjadi bentuk XML Metadata Interchange (XMI). Gambar 1 merupakan contoh *communication diagram* jawaban, dan Gambar 2 merupakan contoh kunci jawaban.

3) Metode Pengukuran Keserupaan antara Dua Communication Diagram menggunakan Graph Edit Distance (GED)

Keserupaan *communication diagram* diukur dengan memperhatikan aspek struktural. Keserupaan struktural diukur dengan memodelkan *communication diagram* dalam sebuah graf. Keserupaan struktural dibagi menjadi dua komponen, inter-struktur dan intra-struktur. Selanjutnya masing-masing komponen diukur keserupaannya menggunakan metode GED.

Keserupaan struktural *communication diagram* diukur dengan memodelkan *communication diagram* dalam bentuk graf. Tabel 1 merupakan penandaan elemen graf yang dipakai untuk membuat graf representasi *communication diagram*.

Tabel 1 Penandaan Elemen Graf pada Communication Diagram

No	Tipe Elemen	Nama	Penandaan
1	Simpul	Simpul Lifeline	Vl
2	Simpul	Simpul Connector	Vc
3	Simpul	Simpul Self Connector	Vsc
4	Simpul	Simpul Forward Message	Vfm
5	Simpul	Simpul Reverse Message	Vrm
6	Simpul	Simpul Arguments	Va
7	Simpul	Simpul Assignment Target	Vat
8	Simpul	Simpul Guard	Vg
9	Simpul	Simpul Iteration	Vi
10	Tepian	Tepian Connector	e _c
11	Tepian	Tepian Self Connector	e _{sc}
12	Tepian	Tepian Forward Message	e _{fm}
13	Tepian	Tepian Reverse Message	e _{rm}
14	Tepian	Tepian Arguments	e _a
15	Tepian	Tepian Assignment Target	e _{at}
16	Tepian	Tepian Guard	e _g
17	Tepian	Tepian Iteration	e _i

Pengukuran keserupaan struktural *communication diagram* didapatkan dari dua kategori, yaitu inter-struktur dan intra-struktur. Inter-struktur merupakan pengukuran keserupaan antara dua *communication diagram* berdasarkan hubungan antar *lifeline* pada masing-masing diagram. Sedangkan intra-struktur merupakan pengukuran keserupaan antara dua *communication diagram* berdasarkan atribut-atribut yang dimiliki oleh setiap *lifeline*, *connector*, *forward message*, atau *reverse message* pada masing-masing diagram. Perhitungan keserupaan struktural antara dua *communication diagram* ($strucSim(d_1, d_2)$) dapat dilihat pada (1).

$$strucSim(d_1, d_2) = w_{inter} \times interSim(d_1, d_2) + w_{intra} \times intraSim(d_1, d_2) \quad (1)$$

Keterangan:

d_1 = *communication diagram* pertama,

d_2 = *communication diagram* kedua,

w_{inter} = bobot keserupaan inter-struktur,

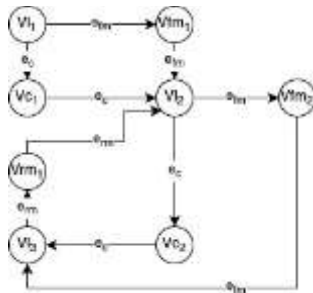
w_{intra} = bobot keserupaan intra-struktur,

$interSim$ = keserupaan inter-struktur *communication diagram*,

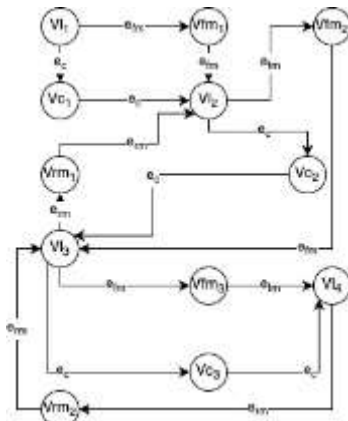
$intraSim$ = keserupaan intra-struktur *communication diagram*.

Contoh perhitungan keserupaan struktural dengan asumsi bobot keserupaan inter-struktur (w_{inter}) dan bobot keserupaan intra-struktur (w_{intra}) adalah sama sehingga masing-masing memiliki nilai 0,5.

a. Pengukuran Keserupaan Inter-Struktur
Communication Diagram



Gambar 3 Model Graf Inter-Struktur dari CD1



Gambar 4 Model Graf Inter-Struktur dari CD2

Pada proses ini dilakukan beberapa tahap, yaitu:

1. Pemodelan *communication diagram* dalam bentuk graf

Tahap pertama yang dilakukan untuk mengukur keserupaan inter-struktur dari *communication diagram* adalah memodelkan *communication diagram* dalam bentuk *graf*. Gambar 3 dan 4 merupakan contoh pemodelan dari CD1 (Gambar 1) dan CD2 (Gambar 2).

Pemodelan *communication diagram* dalam bentuk graf untuk pengukuran keserupaan inter-struktur hanya memperhatikan hubungan antar *lifeline* saja dan mengabaikan atribut-atribut yang ada pada *lifeline*, *connector*, *forward message*, dan *reverse message*. Pemodelan dilakukan mengikuti penandaan pada Tabel 1.

2. Pengukuran keserupaan menggunakan *GED-greedy*

2.1. Pembentukan matriks *cost C*

Matriks *cost C* merupakan matriks yang berisikan *cost* yang dibutuhkan untuk mengubah graf pertama (model graf dari *communication diagram* 1) menjadi graf kedua (model graf dari *communication diagram* 2). *Cost* merupakan langkah yang dibutuhkan untuk mengubah dari simpul menjadi simpul'. Matriks memiliki ukuran $(m + n) \times (m + n)$, dimana m merupakan jumlah

simpul pada graf 1 dan n merupakan jumlah simpul pada graf 2. Sebagai contoh, graf 1 (Gambar 3) memiliki himpunan simpul $V_1 = \{Vl_1, Vl_2, Vl_3, Vc_1, Vc_2, Vfm_1, Vfm_2, Vrm_1\}$ dan graf 2 (Gambar 4) memiliki himpunan simpul $V_2 = \{Vl_1, Vl_2, Vl_3, Vl_4, Vc_1, Vc_2, Vc_3, Vfm_1, Vfm_2, Vfm_3, Vrm_1, Vrm_2\}$. Sehingga diperoleh $m = 8$ dan $n = 12$.

Selanjutnya diperoleh V_1^+ yang merupakan himpunan simpul dari graf 1 yang telah ditambah dengan simpul kosong sejumlah n , sehingga $V_1^+ = \{Vl_1, Vl_2, Vl_3, Vc_1, Vc_2, Vfm_1, Vfm_2, Vrm_1, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon\}$. Selanjutnya diperoleh V_2^+ yang merupakan himpunan simpul dari graf 2 yang telah ditambah dengan simpul kosong sejumlah m , sehingga $V_2^+ = \{Vl_1, Vl_2, Vl_3, Vl_4, Vc_1, Vc_2, Vc_3, Vfm_1, Vfm_2, Vfm_3, Vrm_1, Vrm_2, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon\}$. Apabila anggota V_1^+ dan V_2^+ sudah berjumlah $m + n$ selanjutnya dibuat matriks *cost C*. matriks *cost C* dibagi menjadi empat seperti pada (2):

$$C = \begin{bmatrix} Q_1 & Q_3 \\ Q_2 & Q_4 \end{bmatrix} \quad (2)$$

Keterangan:

$Q_1 = cost$ yang didapatkan untuk substitusi dari u_i menjadi v_j

$Q_2 = cost$ yang didapatkan untuk menghapus u_i (dari u_i menjadi ε)

$Q_3 = cost$ yang didapatkan untuk menambahkan v_j (dari ε menjadi v_j)

$Q_4 = cost$ yang didapatkan untuk substitusi dari ε menjadi ε

$u_i = simpul$ pada g1

$v_j = simpul$ pada g2

Selanjutnya dihitung *cost* dari masing-masing simpul dan hasilnya dimasukkan ke dalam matriks *cost C*.

2.2. Perhitungan *cost total* menggunakan *greedy*

Perhitungan keserupaan dilakukan dengan menambahkan algoritma *greedy* pada metode *Graph Edit Distance (GED)*. Algoritma *greedy* digunakan pada pemilihan permutasi untuk mencari jumlah *cost* terkecil untuk mendapatkan graf 2 dari graf asal graf 1. Untuk mencari *cost minimum* pada GED perlu dilakukan perhitungan dari masing-masing permutasi yang ada sehingga pada GED perlu dilakukan perhitungan sebanyak $(m + n)!$ Untuk mengurangi jumlah percobaan yang dilakukan maka algoritma *greedy* digunakan pada tahap ini.

Algoritma *greedy* dilakukan dua kali, yang pertama dilakukan adalah: apabila $m > n$ maka pemilihan *cost* pertama pada matriks C dari quadran 2 (Q_2). Apabila $m < n$ maka pemilihan *cost* pertama pada matriks C dari quadran 3 (Q_3). Apabila $m = n$ maka pemilihan *cost* pertama pada matriks C dari quadran 1 (Q_1). Berdasarkan aturan yang telah

disebutkan sebelumnya, karena $m < n$ maka pemilihan cost pertama diambil dari Q_3 . Apabila nilai minimum yang telah didapatkan untuk merubah $V_{S_1} \rightarrow V_{S_1}'$ maka semua nilai dari kolom dan baris tersebut tidak dapat dipakai lagi pada matriks C untuk mencari cost minimum.

b. Pengukuran Keserupaan Intra-Struktur Communication Diagram

Keserupaan intra-struktur dibagi menjadi tiga komponen yaitu *lifeline*, *connector*, *forward message*, dan *reverse message*. Untuk mendapatkan nilai keserupaan intra-struktur ($intraSim(d_1, d_2)$) dilakukan dengan mengikuti (3).

$$intraSim(d_1, d_2) = w_{lf} \times lfSim(d_1, d_2) + w_{cn} \times cnSim(d_1, d_2) + w_{fm} \times fmSim(d_1, d_2) + w_{rm} \times rmSim(d_1, d_2) \quad (3)$$

Keterangan:

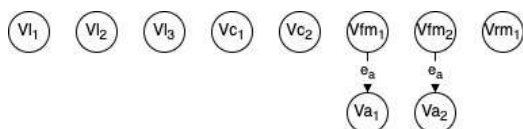
- d_1 = communication diagram pertama,
- d_2 = communication diagram kedua,
- w_{lf} = bobot keserupaan lifeline,
- w_{cn} = bobot keserupaan connector,
- w_{fm} = bobot keserupaan forward message,
- w_{rm} = bobot keserupaan reverse message,
- $lfSim$ = keserupaan lifeline communication diagram,
- $cnSim$ = keserupaan connector communication diagram,
- $fmSim$ = keserupaan forward message communication diagram,
- $rmSim$ = keserupaan reverse message communication diagram.

Nilai keserupaan *lifeline*, *connector*, *forward message*, dan *reverse message* yang telah didapatkan digunakan untuk menghitung keserupaan intra-struktur mengikuti (3). Bobot keserupaan *lifeline* (w_{lf}), bobot keserupaan *connector* (w_{cn}), bobot keserupaan *forward message* (w_{fm}), dan bobot keserupaan *reverse message* (w_{rm}) diasumsikan adalah sama sehingga masing-masing memiliki nilai 0,25.

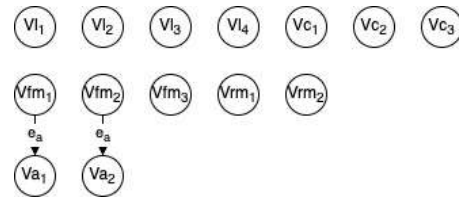
Pada proses ini dilakukan beberapa tahap yaitu:

1. Pemodelan setiap *lifeline*, *connector*, *forward message*, dan *reverse message* dalam bentuk graf

Tahap pertama untuk mengukur keserupaan intra-struktur dari *communication diagram* adalah memodelkan setiap *lifeline*, *connector*, *forward message*, dan *reverse message* dari *communication diagram* dalam bentuk graf.



Gambar 5 Model Graf Intra-Struktur dari CD1



Gambar 6 Model Graf Intra-Struktur dari CD2

Graf 1 memiliki 3 simpul *lifeline*, 2 simpul *connector*, 2 simpul *forward message*, dan 1 simpul *reverse message*. Graf 2 memiliki 4 simpul *lifeline*, 3 simpul *connector*, 3 simpul *forward message*, dan 2 simpul *reverse message*. Dari pemasangan *lifeline*, *connector*, *forward message*, dan *reverse message* dari graf 1 ke graf 2 maka akan terdapat 7 pasang *lifeline*, 5 pasang *connector*, 5 pasang *forward message*, dan 3 pasang *reverse message* yang perlu dihitung menggunakan GED yang telah ditambah dengan algoritma *greedy*.

2. Perhitungan keserupaan intra-struktur

Nilai keserupaan *lifeline* ($lfSim(d_1, d_2)$) diperoleh dengan menerapkan algoritma *greedy* untuk mencari permutasi dengan jumlah *cost* minimum dari kumpulan nilai *cost* pasangan *lifeline*. Nilai tersebut disimpan dalam bentuk matriks *cost* L sesuai (4).

$$L = \begin{bmatrix} Vl_1Vl_1 & \dots & Vl_iVl_j \\ \vdots & \ddots & \vdots \\ Vl_iVl_j & \dots & Vl_xVl_y \end{bmatrix} \quad (4)$$

Keterangan:

x = jumlah simpul *lifeline* pada *communication diagram* pertama

y = jumlah simpul *lifeline* pada *communication diagram* kedua

Nilai keserupaan *connector* ($cnSim(d_1, d_2)$) diperoleh dengan menerapkan algoritma *greedy* untuk mencari permutasi dengan jumlah *cost* minimum dari kumpulan nilai *cost* pasangan *connector*. Nilai tersebut disimpan dalam bentuk matriks *cost* Cn sesuai (5)

$$Cn = \begin{bmatrix} Vcn_1Vcn_1 & \dots & Vcn_iVcn_j \\ \vdots & \ddots & \vdots \\ Vcn_iVcn_j & \dots & Vcn_uVcn_v \end{bmatrix} \quad (5)$$

Keterangan:

u = jumlah simpul *connector* pada *communication diagram* pertama

v = jumlah simpul *connector* pada *communication diagram* kedua

Nilai keserupaan *forward message* ($fmSim(d_1, d_2)$) diperoleh dengan menerapkan algoritma *greedy* untuk mencari permutasi dengan jumlah *cost* minimum dari kumpulan nilai *cost* pasangan *forward message*. Nilai tersebut disimpan dalam bentuk matriks *cost* F sesuai (6)

didapatkan adalah 2. Karena *cost* minimum yang didapatkan adalah 2, maka nilai keserupaan *forward message* ($fmSim(d_1, d_2)$) yang didapatkan adalah 0,5.

Nilai keserupaan *reverse message* ($rmSim(d_1, d_2)$) didapatkan dari matriks *cost* R berikut.

$$R = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Setelah dilakukan greedy permutasi yang didapatkan adalah 1, 1 sehingga *cost* GED yang didapatkan adalah 0. Karena *cost* minimum yang didapatkan adalah 0, maka dapat dikatakan bahwa *reverse message* graf 1 dan graf 2 sepenuhnya sama. Sehingga nilai keserupaan *reverse message* ($rmSim(d_1, d_2)$) yang didapatkan adalah 1, atau sepenuhnya serupa.

Untuk mendapatkan nilai keserupaan intra-struktur, dilakukan dengan menggunakan (3). Bobot keserupaan *lifeline* (w_{lf}), bobot keserupaan *connector* (w_{cn}), bobot keserupaan *forward message* (w_{fm}), dan bobot keserupaan *reverse message* (w_{rm}) diasumsikan adalah sama sehingga masing-masing memiliki nilai 0,25. Sehingga didapatkan hasil perhitungan $intraSim(d_1, d_2)$ yaitu 0,875.

Untuk menghitung keserupaan struktural *communication diagram* dilakukan dengan menggunakan (1). Bobot keserupaan inter-struktur (w_{inter}) dan bobot keserupaan intra-struktur (w_{intra}) adalah sama sehingga masing-masing memiliki nilai 0,5. Dengan nilai keserupaan inter-struktur adalah 0,696 dan nilai keserupaan intra-struktur adalah 0,875 maka nilai keserupaan *communication diagram* yang didapatkan adalah 0,785.

SIMPULAN

Metode *Graph Edit Distance* dapat digunakan untuk mengukur keserupaan *communication diagram* dengan bantuan algoritma *greedy* untuk penentuan *cost* minimum. Keserupaan *communication diagram* diukur menggunakan aspek struktural dengan tujuan untuk penilaian jawaban soal. Hasil penilaian antara diagram jawaban dan diagram kunci jawaban selanjutnya dapat digunakan untuk membantu pengampu mata pelajaran dalam memberikan nilai. Pada pengukuran keserupaan struktural dibagi menjadi dua yaitu inter-struktur dan intra-struktur. Sebelum melakukan pengukuran keserupaan struktural, *communication diagram* dimodelkan terlebih dahulu ke bentuk graf.

DAFTAR PUSTAKA

- [1] D. R. Garrison, *E-learning in The 21st Century: A Framework for Research and Practice*. Taylor & Francis, 2011.
- [2] H. Djaali and P. Muljono, *Pengukuran dalam Bidang Pendidikan*. Jakarta: Grasindo, 2008.
- [3] M. A. Hussein, H. Hassan, and M. Nassef, "Automated language essay scoring systems: A literature review," *PeerJ Comput Sci*, vol. 5, p. e208, 2019.
- [4] T. Isaacs, C. Zara, G. Herbert, and S. Coombs, *Key Concepts in Educational Assessment*. London: Thousand Oaks: Sage Publications Ltd, 2013. doi: 10.4135/9781473915077.
- [5] B. Cope and M. Kalantzis, *E-learning Ecologies: Principles for New Learning and Assessment*. Taylor & Francis, 2017.
- [6] Y. Attali and J. Burstein, "Automated essay scoring with E-Rater R V.2.0," *ETS Research Report Series*, pp. i-21, 2004.
- [7] S. Valenti, F. Neri, and A. Cucchiarelli, "An overview of current research on automated essay grading," *Journal of Information Technology Education: Research*, vol. 2, pp. 319-330, 2017, doi: 10.28945/331.
- [8] H. Zhang and D. Litman, "Co-attention based neural network for source-dependent essay scoring," *arXiv preprint arXiv:1908.01993*, 2019.
- [9] S. R. Ningsih, E. Erdisna, and F. Suryana, "Aplikasi E-Task Berbasis Problem Based Learning Pada Mata Kuliah Perancangan Basis Data Di Perguruan Tinggi," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 4, no. 1, pp. 12-19, Jan. 2022, doi: 10.47233/JTEKSIS.V4I1.347.
- [10] H. Andrianof and M. Kom, "Perancangan Aplikasi M-Learning Untuk Pembelajaran Ebusiness Bagi Mahasiswa Dengan Menggunakan Perangkat Mobile Berbasis Android," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 2, no. 2, pp. 92-98, Jul. 2020, doi: 10.47233/JTEKSIS.V2I2.120.
- [11] P. Anggraini, D. P. Mulya, and S. Sularno, "Perancangan Aplikasi Customer Relationship Management Berbasis Wap Pada The Aliga Hotel," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 2, no. 2, pp. 161-186, Jul. 2020, doi: 10.47233/JTEKSIS.V2I2.128.
- [12] R. Fauzan, D. Siahaan, S. Rochimah, and E. Triandini, "Class diagram similarity measurement: A different approach," in *2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE)*, 2018, pp. 215-219.
- [13] E. Triandini, R. Fauzan, D. O. Siahaan, and S. Rochimah, "Sequence diagram similarity measurement: A different approach," in *2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 2019, pp. 348-351.
- [14] R. Fauzan, D. Siahaan, S. Rochimah, and E. Triandini, "Use case diagram similarity measurement: A new approach," in *2019 12th International Conference on Information & Communication Technology and System (ICTS)*, 2019, pp. 3-7.
- [15] A. Adamu and W. M. N. W. Zainon, "Matching and retrieval of state machine diagrams from software repositories using Cuckoo Search Algorithm," in *2017 8th International Conference on Information Technology (ICIT)*, 2017, pp. 187-192.
- [16] H. Munawaroh, D. O. Siahaan, R. Fauzan, and E. Triandini, "Structural similarity measurement using Graph Edit Distance-greedy on statechart diagrams," in *2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS)*, 2020, pp. 1-6. doi: 10.1109/ICORIS50180.2020.9320764.
- [17] R. Fauzan, D. Siahaan, S. Rochimah, and E. Triandini, "Activity diagram similarity measurement: A different approach," in *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2018, pp. 601-605. doi: 10.1109/ISRITI.2018.8864284.

- [18] J.-B. Gao, B.-W. Zhang, and X.-H. Chen, "A WordNet-based semantic similarity measurement combining edge-counting and information content theory," *Eng Appl Artif Intell*, vol. 39, pp. 80–88, 2015.
- [19] M. Sevenster, J. Bozeman, A. Cowhy, and W. Trost, "A natural language processing pipeline for pairing measurements uniquely across free-text CT reports," *J Biomed Inform*, vol. 53, pp. 36–48, 2015.
- [20] Z. Yuan, L. Yan, and Z. Ma, "Structural similarity measure between UML class diagrams based on UCG," *Requir Eng*, 2019, doi: <https://doi.org/10.1007/s00766-019-00317-w>.
- [21] A. Adamu and W. M. N. W. Zainoon, "A framework for enhancing the retrieval of UML diagrams," in *International Conference on Software Reuse*, 2016, pp. 384–390.
- [22] A. Adamu and W. M. N. W. Zainon, "Multiview Similarity Assessment Technique of UML Diagrams," *Procedia Comput Sci*, vol. 124, pp. 311–318, 2017.