

Analisis Kemiripan Model Proses Bisnis menggunakan Algoritma Heuristik

Analysis of Similarity Business Process Model using Heuristic Algorithm

Septia Dusitella Sembiring¹, Dana Kusumo Ph.D.², Angelina Prima K ST.,MT³

^{1,3}Teknik Informatika, ²Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

¹septiasembiring7@gmail.com, ²dskusumo.itt@gmail.com, ³angelina.st3@gmail.com

Abstrak

Setiap organisasi maupun perusahaan memiliki satu atau beberapa proses bisnis untuk mendukung analisis, desain ulang dan implementasi dari sebuah aktivitas. Permasalahan yang terjadi adalah ketika satu atau beberapa proses bisnis memiliki beberapa kesamaan sehingga harus mengidentifikasi efektifitas maupun efisiensi pada model proses bisnis yang berbeda. Pada penelitian ini penulis akan menganalisis kemiripan model proses bisnis dengan metode struktural menggunakan algoritma heuristik pada studi kasus pelaksanaan Tugas Akhir (TA) Universitas Telkom. Hasil dari penelitian ini yaitu penentuan tingkat kemiripan dilakukan dengan cara menghitung *string edit distance*, *string edit similairty*, pemetaan dengan algoritma heuristik, *graph edit distance* dan *graph edit similarity* secara berurutan. Juga tingkat kemiripan antara dua proses bisnis dan perbandingan cara kerja pada algoritma heuristik dibandingkan dengan menggunakan A*.

Kata kunci : proses bisnis, kemiripan struktural, algoritma heuristik, A*

Abstract

*Each company or organization has one or several business process to support analysis, redesign and implementation of an activity. The problem that occurs is when one or more business process has some similarity and have to identify the effectiveness and efficiency in the different model business process. This final project will doing an analysis of the similarity of the business process model using structural similarity with heuristic algorithm in case of execution final project in Telkom University. Output from this research is level of similarity by calculating string edit distance (sed), string edit similarity (ses), mapping with heuristic algorithm, graph edit distance (ged) and graph edit similarity (ges) in sequence. Similarity between two process business and ways of working on heuristic algorithm compared using A**

Keywords: *business process, structural similarity, heuristic algorithm, A**

1. Pendahuluan

Proses bisnis banyak diterapkan di dalam organisasi maupun perusahaan untuk mendukung analisis, desain ulang maupun implementasi aktivitas. Pada dasarnya, proses bisnis adalah sekumpulan koleksi dari aktivitas yang berhubungan dan terstruktur ataupun tugas untuk mencapai sebuah tujuan [1]. Proses bisnis pelaksanaan TA pada Fakultas Informatika dengan Fakultas lain memiliki perbedaan tetapi beberapa juga memiliki beberapa kesamaan. Permasalahan proses bisnis ini mengacu pada kemiripan proses bisnis untuk mengidentifikasi model proses bisnis mana yang paling menyerupai dari proses model bisnis yang diberikan. Sebagai organisasi yang telah mencapai level tinggi dari *Business Process Management (BPM)*, sebuah organisasi cenderung mengakumulasi cukup banyak model proses bisnis – dilaporkan dalam ratusan atau ribuan kasus dari perusahaan multinasional [1].

Permasalahan pencarian tingkat kemiripan proses bisnis dapat ditinjau dari 3 sisi yaitu *text/label similarity*, *structural similarity* dan *behavioral similarity*. Penyelesaian untuk pencarian tingkat kemiripan telah banyak dilakukan dan diidentifikasi dengan metode *text similarity (label similarity)*, *structural similarity* dan *behavioural similarity* dalam studi kasus yang berbeda dan pemakaian algoritma maupun tools yang berbeda pula.

Dalam metode *structural similarity* pencarian tingkat kemiripan dapat diselesaikan dengan 4 algoritma heuristik yaitu algoritma greedy, algoritma exhaustive dengan pruning, algoritma heuristik maupun algoritma A-

star [1]. Tugas akhir ini menganalisis kemiripan model proses bisnis menggunakan algoritma heuristik dengan metode *structural similarity* pada proses bisnis pelaksanaan Tugas Akhir Fakultas Informatika, Fakultas Teknik Industri dan Fakultas Teknik Elektro

2. Proses Bisnis

Proses bisnis adalah sekumpulan koleksi dari aktivitas yang berhubungan dan terstruktur ataupun tugas untuk mencapai sebuah tujuan [1]. Karakteristik dari proses bisnis dapat dilihat seperti dibawah ini [2] :

- a. *Definability* : Harus dengan jelas mendefinisikan batasan – batasan masukan (*input*) dan keluaran (*output*)
- b. *Order* : Harus terdiri dari aktivitas yang diurutkan berdasarkan posisi di waktu dan spasial
- c. *Customer* : harus ada yang menjadi penerima dari hasil proses yaitu *customer*
- d. *Value – adding* : transformasi yang terjadi di dalam proses harus memberikan nilai tambah pada penerima
- e. *Embeddedness* : Sebuah proses tidak akan ada dengan sendirinya, harus tertanam di sebuah struktur organisasi
- f. *Cross – functionality*: proses umumnya, walaupun tidak mencakup beberapa fungsi.

Proses bisnis dapat dimodelkan kedalam beberapa notasi yang tersedia seperti seperti *Event – driven Process Chain (EPC)*, *UML Activity Diagram* dan *Business Process Modelling Notation (BPMN)* [1]

2.1 Kemiripan Proses Bisnis

Kemiripan suatu proses bisnis adalah adanya kemiripan ataupun kesamaan dari proses bisnis suatu organisasi atau perusahaan. Menurut Michael Becker dan Ralf Laure [3] pengukuran kemiripan proses bisnis disarankan untuk tujuan yang berbeda – beda misalnya pengukuran kepatuhan antara referensi dan model aktual, pencarian model yang mirip didalam *repository*. Kemiripan model proses bisnis dapat dibagi menjadi 3 metode yaitu :

- a. *Label/text similarity* : perbandingan kemiripan berdasarkan label yang muncul pada proses bisnis.
- b. *Structural similarity* : pengukuran kemiripan dilakukan dengan mengibaratkan proses bisnis seperti graf
- c. *Behavioural similarity* : pengukuran kemiripan dengan melihat perilaku model proses bisnis.

2.2 Label/text similarity

Pengukuran kemiripan yang pertama dinamakan *label/text similarity*. Berdasarkan perbandingan dari label yang muncul di model proses bisnis (label *task*, label *events*, dll) dapat dicari menggunakan kemiripan *syntactic*, *semantic* atau *contextual* [1]. Dalam melakukan label *matching similarity* dapat digunakan metode *semantic*, *syntactic* ataupun *contextual*. Ada dua hal yang dilakukan untuk mencari *label/text similarity* yaitu :

2.2.1 String edit distance (sed)

Edit distance adalah cara untuk melihat seberapa mirip dan tidak miripnya 2 buah string yang dibandingkan dengan cara menghitung minimum jumlah operasi untuk mengubah satu string ke string yang lain.

Operasi dalam string edit distance ini ada 3 yaitu operasi *insert*, hapus dan substitusi. Pada *string edit distance* disediakan algoritma yaitu *Levenshtein distance* dan tools yang bernama *Levenshtein calculator*. Sebagai contoh ses antara “*Verify invoice*” dan “*Verification invoice*” adalah 7 yaitu substitusi “y” dan “i” dan penambahan “cation”.

2.2.2 String Edit Similarity (ses)

Ses menunjukkan seberapa besar kemiripan dari 2 buah string yang dibandingkan setelah dilakukan pencarian *sed*.

Ada 3 cara untuk menghitung kemiripan antara elemen dari beberapa proses model yang berbeda [4] yaitu :

- a. *Syntactic similarity*, hanya memperhitungkan sintaks dari label/string
- b. *Semantic similarity*, melihat dari semantic (makna) dari sebuah kata pada label/string
- c. *Contextual similarity*, tidak hanya mempertimbangkan elemen dari label itu sendiri, tetapi juga konteks dimana elemen terjadi

$$syn(n_1, n_2) = 1 - \frac{ed(l_1(n_1), l_2(n_2))}{\max(|l_1(n_1)|, |l_2(n_2)|)}$$

2.3 Structural Similarity

Pengukuran kemiripan yang kedua adalah metode secara struktural. Pada metode ini model proses bisnis diibaratkan berbentuk seperti graf sebagai basis untuk pengukur kemiripan [5]. Digunakan konsep *ged* dan *ges* untuk mengevaluasi kemiripan antara 2 buah model proses bisnis

2.2.1 Graph Edit Distance (*ged*)

Ged adalah jumlah minimum nilai dari operasi yang diperlukan untuk mengubah atau mentransformasikan satu graf ke graf yang lainnya.

Operasi – operasi yang dilakukan *ges* adalah sebagai berikut [1]:

- 1) Node *substitutions* : node yang berada pada satu graf disubstitusi dengan node dalam graf lain jika dan hanya jika mereka cocok
- 2) Node *insertion/deletion* : node diinputkan kedalam atau dihapus keluar dari graf
- 3) Edge *insertion/deletion* : edge yang diinputkan kedalam atau dihapus keluar dari graf

Diasumsikan bahwa nilai untuk mensubstitusi sebuah node adalah satu minus dari kemiripan node. Kemiripan node dilihat dari label atau tipe dari node. *ged* dapat dicari dengan rumus berikut :

$$|skipn| + |skipe| + 2 \cdot \sum_{(n_1, n_2) \in M} (1 - Sim(n_1, n_2))$$

Dimana :

Skipn : satu set dari *inserted* ataupun *deleted* node

Skipe : satu set dari *inserted* ataupun *deleted* edge

2.2.2 Graph – Edit Similarity (*ges*)

Ges adalah kemungkinan kemiripan paling maksimal diantara kedua graf [1]. Untuk mengkomputasikan *ges* pada dua buah proses graf, sebelumnya harus dilakukan pencarian pemetaan yang akan mengembalikan kemiripan yang mungkin dan paling maksimal [1]. Pencarian pemetaan ini akan mengakibatkan kompleksitas faktorial atau yang biasa disebut NP-completeness. Pada penelitian ini akan digunakan algoritma heuristik untuk melakukan pemetaan.

a. Algoritma Heuristik

Pada penelitian ini algoritma yang dipakai adalah algoritma heuristik. Algoritma ini bertujuan untuk mencari yang paling optimal diantara semua solusi yang mungkin, yaitu mencari nilai yang paling minimum ataupun maksimum [6]. Pada studi kasus algoritma ini akan mencari nilai *similarity* paling tinggi dan melakukan pemangkasan untuk node yang telah dipetakan.

Setelah pemetaan dilakukan maka pencarian *ges* dilakukan dengan menggunakan rumus sebagai berikut [4]:

$$gedsim(B1, B2) = 1 - avg(fskipe, fskipn, fsubn)$$

Dimana :

$$fskipn = \frac{|skipn|}{|N1|+|N2|} \quad ; \quad fskipe = \frac{|skipe|}{|E1|+|E2|} \quad ; \quad fsubn = \frac{2 \cdot \sum_{(n,m) \in M} 1 - Sim(n,m)}{|N1|+|N2|-|skipn|}$$

3. Pembahasan

3.1.1 String edit distance dan string edit similarity

Pada pengujian akan dilakukan perbandingan antara program yang telah dibuat dengan tools yang telah tersedia yaitu *Levensthein Calculator* untuk menyatakan kevalidan program yang telah dibuat. Berdasarkan penelitian didapatkan bahwa pengujian kevalidan yang didapat dari pengujian dari *sed* dan *ses* program dibandingkan dengan program *Levensthein Calculator* yang sudah tersedia adalah valid karena menunjukkan nilai yang sama pada keduanya. Hubungan antara *sed* dan *ses* adalah berbanding terbalik, semakin kecil *sed* maka *ses* akan semakin besar dan sebaliknya

3.1.2 Graph edit distance dan graph edit similarity

Pengujian ini dilakukan untuk mengetahui seberapa besar usaha untuk mengubah satu proses bisnis ke proses bisnis lainnya dan mengetahui seberapa besar kemiripan antara proses bisnis Fakultas Teknik Informatika, Industri dan Elektro yang ada di Universitas Telkom. Untuk mendapatkan *ges* dan *ged* dilakukan perhitungan seperti rumus yang telah dijelaskan diatas. Setelah perhitungan dilakukan maka hasil yang didapatkan adalah sebagai berikut :

TABLE 1 : HASIL GED DAN GES PADA SETIAP PROSES BISNIS TA

<i>Query</i>	<i>Variant</i>	<i>ged</i>	<i>ges</i>
Teknik Industri	Teknik Informatika	57.19	0.63
Teknik Informatika	Teknik Industri	57.19	0.63
Teknik Industri	Teknik Elektro	51.33	0.70
Teknik Elektro	Teknik Industri	51.33	0.70
Teknik Elektro	Teknik Informatika	59.80	0.61
Teknik Informatika	Teknik Elektro	59.80	0.61
Teknik Industri	Teknik Industri	0	1
Teknik Elektro	Teknik Elektro	0	1
Teknik Informatika	Teknik Informatika	0	1

Dari tabel diatas diperoleh bahwa nilai *ged* dan *ges* adalah berbanding terbalik. Semakin besar nilai *ged* maka nilai *ges* akan semakin kecil yang berarti semakin besar usaha untuk merubah satu graf ke graf lainnya maka semakin kecil kemiripan dari 2 graf dan sebaliknya. Diperoleh juga bahwa semakin kecil *ged* maka *ges* akan semakin besar dan sebaliknya

3.1.3 Cara kerja algoritma heuristik dan A*

Pengujian ini dilakukan untuk mengetahui bagaimana cara kerja antara algoritma heuristik dan A* dan penyebab Perbedaan *ged* dan *ges* yang didapatkan oleh kedua algoritma

TABEL 1 HASIL PENGUJIAN PERBANDINGAN ALGORITMA HEURISTIK DAN ALGORITMA A*

<i>Query</i>	<i>Variant</i>	Algoritma Heuristik		Algoritma A*	
		<i>ged</i>	<i>ges</i>	<i>ged</i>	<i>ges</i>
TA Teknik Industri	TA Teknik Informatika	57.19	0.63	84.36	0.42
TA Teknik Informatika	TA Teknik Industri	57.19	0.63	84.36	0.42
TA Teknik Industri	TA Teknik Elektro	51.33	0.70	76.52	0.54
TA Teknik Elektro	TA Teknik Industri	51.33	0.70	76.52	0.54
TA Teknik Elektro	TA Teknik Informatika	59.80	0.61	90.97	0.33
TA Teknik Informatika	TA Teknik Elektro	59.80	0.61	90.97	0.33
TA Teknik Informatika	TA Teknik Informatika	0	1	0	1
TA Teknik Elektro	TA Teknik Elektro	0	1	0	1
TA Teknik Industri	TA Teknik Industri	0	1	0	1

Berikut merupakan tabel analisis hasil pengujian antara cara kerja algoritma heuristik dan algoritma A*

TABEL 2 PERBANDINGAN CARA KERJA ALGORITMA HEURISTIK DAN A*

Algoritma Heuristik	Algoritma A*
<i>Ses</i> yang diambil pada setiap pemetaan adalah yang paling maksimal	<i>Ses</i> yang diambil pada setiap pemetaan adalah yang memenuhi <i>cutoffvalue</i> tertentu
Jika 2 proses bisnis dibandingkan maka yang memiliki jumlah node lebih kecil semua node akan dipetakan	Node yang tidak memenuhi syarat akan menjadi skipn (node yang diinsert)
Melakukan sistem pemangkasan pada saat pemetaan	Melakukan sistem pemangkasan pada saat pemetaan
Hasil <i>ges</i> akan lebih besar daripada A* karena usaha untuk memetakan semua node. Sehingga nilai <i>skipn</i> akan lebih kecil.	Hasil <i>ges</i> lebih kecil dari heuristik karena pengaruh <i>skipn</i> dan <i>skipe</i> yang besar membuat <i>ged</i> semakin besar

4. Kesimpulan

Berdasarkan hasil percobaan dan analisis dari penelitian ini adalah penentuan tingkat kemiripan menggunakan metode structural dapat dilakukan dengan cara pencarian *sed*, *ses*, pemetaan dengan algoritma heuristik, *ged* dan *ges* secara berurutan.

Pada tabel 5 terlihat bahwa hubungan *sed* dan *ses* adalah berbanding terbalik. Semakin kecil usaha untuk mengubah satu string ke string yang lain maka nilai kemiripannya akan semakin besar dan sebaliknya.

Bahwa nilai pemetaan algoritma heuristik dengan *ged* berbanding lurus. Semakin kecil nilai pemetaan maka nilai *ged* akan semakin kecil dan sebaliknya. Terlihat juga bahwa hubungan nilai *ged* dan *ges* berbanding terbalik. Semakin kecil usaha untuk mengubah satu graf ke graf yang lain maka nilai kemiripannya semakin besar dan sebaliknya.

Pada perbandingan algoritma heuristik dan A* nilai *ges* pada A* terlihat selalu lebih kecil dari heuristik diakibatkan oleh pengaruh nilai *ged* pada A* yang besar karena cara kerja yang berbeda



References

- [1] M. D. L. G.-B. Remco Dijkman, "Graph Matching Algorithm for Business Process Model Similarity," 2009.
- [2] "Business Process," Wikipedia.org, [Online]. Available: en.wikipedia.org/wiki/Business_process. [Accessed 25 February 2015].
- [3] M. Becker and R. Laure, "A Comparative Survey of Business Process," Department of Business Information Systems, University of Leipzig, Germany.
- [4] R. Dijkman, M. Dumas, B. v. Dongen, R. Kaarik and J. Mendling, "Similarity Business Process Models : Metrics and Evaluation".
- [5] J. L, L. Z and Q. F, "An Improved Structure-based Approach to Measure," Beihang University (BUAA), China.
- [6] K. Natallia, "An introduction to heuristic algorithms," Department of Informatics and Telecommunications University of Trento, Italy, Italy.

