Analisis *Business Process Model Similarity Checking* Menggunakan Teknik *Greedy Graph Matching*

Fadhilah Dwiyanti Basri
Telkom University, Indonesia
fadhilahdwiyanti@gmail.com

Abstrak –*Business combination* menghasilkan proses bisnis dalam jumlah besar sehingga seringkali terdapat proses bisnis yang sama dengan tujuan yang sama pula. Oleh sebab itu, dibutuhkan sistem yang dapat mengecek kesamaan proses bisnis sehingga nantinya dapat membantu *user* dalam mengambil keputusan jika ingin melakukan *redesign* atau penggabungan proses bisnis.

Kata kunci : *business process similarity checking, greedy graph matching, syntactic similarity, node insertions/ deletions, edge insertions/deletions, node substitutions*

I. PENDAHULUAN

Untuk menjadi perusahaan yang besar dibutuhkan strategi yang tepat seperti penggabungan usaha *(business combination). Business combination* menghasilkan proses bisnis dalam jumlah besar sehingga seringkali terdapat proses bisnis yang sama dengan tujuan yang sama pula. Oleh sebab itu, dibutuhkan sistem yang dapat mengecek kesamaan proses bisnis sehingga nantinya dapat membantu *user* dalam mengambil keputusan jika ingin melakukan *redesign* atau penggabungan proses bisnis. *Business process similarity checking* merupakan sistem yang dapat mengecek kesamaan dari dua proses bisnis sehingga menghasilkan nilai *similarity.* Dalam mengecek kesamaan, digunakan metode *greedy graph matching.* Metode ini mencari pasangan *node* dengan nilai *matching score* paling optimal [11][13]. Dalam penelitian ini, *matching score* ialah nilai *syntactic similarity* yang paling tinggi. Setiap iterasi, *greedy* memilih pasangan *node* dengan nilai *syntactic similarity* paling tinggi, kemudian pasangan *node* tersebut dihapus agar tidak dipilih lagi pada iterasi selanjutnya. Hal ini terus berulang sampai tidak ada lagi pasangan *node* yang dianggap dapat meningkatkan nilai *similarity (graph edit distance similarity)* [11][13].

II. BAHAN DAN METODE

Tujuan yang ingin dicapai dari tugas akhir ini adalah mengimplementasikan sistem yang mampu mengecek kesamaan dari proses bisnis. Teknik *matching* yang digunakan yaitu metode *greedy graph matching*. Teknik ini melakukan *mapping* ke semua pasangan *node* yang memiliki nilai *matching score* tertinggi. Jika pasangan *node* tersebut dipilih oleh *greedy*, selanjutnya *node* dihapus agar tidak dipilih lagi pada iterasi selanjutnya. *Greedy* terus melakukan iterasi sampai tidak ada lagi pasangan *node* yang dapat meningkatkan nilai *similarity* [11][13].

Data *input* yang digunakan berasal dari prosedur perizinan Badan Pelayanan

Perizinan Terpadu (BPPT) kota Bandung, yaitu tanda daftar perusahaan (TDP) dan izin pemancangan tiang pancang jembatan penyeberangan orang (JPO) [1][2].

Selanjutnya, prosedur perizinan tersebut dimodelkan di BPMN Visio sehingga menjadi proses bisnis yang terdiri dari *activity, gateway* dan *event.* Setelah dimodelkan di BPMN Visio, proses bisnis TDP dan JPO selanjutnya di*export* secara manual ke Microsoft Excel (.xls).

Setelah memodelkan prosedur perizinan dan melakukan *export* dari BPMN ke Microsoft Excel maka selanjutnya adalah menggambarkan graf dari proses bisnis. *Activity, event* dan *gateway* digambarkan sebagai *node,* sedangkan *connecting object* menjadi *edge* [12].

Setelah digambarkan graf proses bisnisnya, maka selanjutnya dibuat konektivitasnya sesuai dengan graf yang telah digambarkan sebelumnya. Konektivitas dibuat di file yang sama dengan hasil *export.*

2.1. *Indexing*

Indexing dilakukan agar data lebih mudah ditemukan. Pada penelitian ini, *indexing* dilakukan pada semua *node.* Misal, *node* "tanda tangan" diberi nomor indeks 13. Maka semua *node* dengan nilai "tanda tangan" diberi indeks 13. Jika terdapat *node* baru seperti "lengkap", maka diberikan nomor indeks baru, yaitu 6. Sehingga semua *node* dengan nilai "lengkap" diberikan indeks 6, begitu seterusnya sampai semua *node* memiliki indeks. Hal ini perlu dilakukan untuk proses setelah *indexing,* yaitu *preprocessing.*

2.2. Preprocessing

Setelah melakukan *indexing,* proses selanjutnya adalah *preprocessing.* Pada proses ini, sistem membuat konektivitas namun sesuai dengan nomor indeksnya. Misal, *node* dengan nilai *"start"* diberi indeks 1, *node* dengan nilai "mencari informasi" diberi indeks 2, *node* dengan nilai "memberikan informasi administrasi dan teknis, menyerahkan formulir" diberi indeks 3, maka hasil *preprocessing*nya adalah 1 menuju 2, 2 menuju 3.

Hasil *indexing* sangat berguna saat *preprocessing* karena dengan adanya nomor indeks maka semua *node* yang menuju "tanda tangan" berarti *node* tersebut menuju nomor indeks 13. Dengan begitu dapat dihindari ambiguitas pada sistem seperti *node* "mencari informasi" yg dituju oleh *node* "start" beda dengan "mencari informasi" yang menuju "memberikan informasi administrasi dan teknis, menyerahkan formulir", padahal seharusnya *node* "mencari informasi" tersebut dianggap sama.

2.3. Syntactic Similarity

Dalam melakukan penghitungan ini, dilakukan proses *mapping* 1 ke n, dimana *business process* 1 menelusuri semua *node* yang terdapat pada *business process* 2. *Syntactic similarity* ini dihitung berdasarkan persamaan (2.1). Untuk penghitungan *string edit distance* sendiri, digunakan metode levenshtein *distance,* dimana metode ini mencari nilai minimum operasi yang dikeluarkan untuk menghitung *string edit distance.*

Setelah ditemukan hasil *string edit distance,* selanjutnya dibagi

dengan *maxlength. Maxlength* atau pada sistem ditulis *max* merupakan jumlah *string* terpanjang antara kedua *node.* Misal, antara *node* "lengkap" dengan total *string* 7 dan *node* "tanda tangan" dengan total *string* 12, maka yang menjadi *maxlength* yaitu 12. Nilai *syntactic similarity* menjadi faktor penentu saat dilakukan *greedy graph matching* karena pasangan *node* yang dipilih *greedy* dilihat berdasar nilai *syntactic similarity*nya.

2.4. Greedy Graph Matching

Setelah mendapatkan hasil *syntactic similarity* pada setiap pasangan *node* selanjutnya dilakukan *greedy graph matching. Greedy* memilih pasangan *node* yang dianggap dapat meningkatkan nilai *similarity* karenanya *greedy* memilih pasangan *node* dengan nilai *syntactic similarity* tinggi. Dalam penelitian ini, nilai *similarity* ditunjukkan dengan nilai *graph edit distance similarity.*

Setelah pasangan *node* dipilih maka *node* dihapus agar pada iterasi selanjutnya tidak dipilih lagi.

Greedy mencari nilai *syntactic similarity* per *node* berdasarkan proses bisnis yang memiliki total *node* terpanjang. Misal, JPO memiliki total *node* 19 dan TDP memiliki total *node* 14, sehingga yang terpanjang ialah JPO. Selanjutnya, *node* pertama di JPO yaitu *"start",* namun karena *node* tersebut tidak memiliki nilai (λ) sehingga tidak dimasukkan saat menghitung *syntactic similarity* maka tidak dilibatkan saat melakukan *greedy graph matching.* Selanjutnya, *node* kedua dari JPO yaitu "mencari informasi", maka *greedy* melihat semua nilai *syntactic similarity* dari pasangan *node* "mencari informasi" dan *node* dari proses bisnis TDP. Pasangan *node* dengan nilai *syntactic similarity* tertinggi dipilih oleh *greedy.* Kemudian *node* yang telah dipilih dari proses bisnis TDP tersebut dihapus agar tidak dipilih lagi saat iterasi selanjutnya.

Setelah itu, proses bisnis ketiga dari JPO yaitu "memberikan informasi administrasi dan teknis, menyerahkan formulir", selanjutnya *greedy* melihat semua nilai *syantactic similarity* dari pasangan *node* "memberikan informasi administrasi dan teknis, menyerahkan formulir" dan *node* dari proses bisnis TDP. Pasangan *node* dengan nilai *syntactic similarity* tertinggi dipilih oleh *greedy.* Kemudian, *node* yang telah dipilih dari proses bisnis TDP dihapus lagi. Begitu seterusnya, sampai dianggap tidak ada lagi pasangan *node* yang dapat meningkatkan nilai *similarity.*

III. HASIL

Uji coba yang dilakukan menggunakan dua proses bisnis yaitu tanda daftar perusahaan (TDP) dan izin pemancangan tiang pancang jembatan penyebarangan orang (JPO). Penghitungan *graph edit distance similarity* dilakukan dengan membandingkan JPO dan TDP.

Tabel 3.1 Hasil penghitungan JPO dan TDP

Penghitungan	Hasil
node insertions/deletions (sn)	7
rata-rata *node insertions/deletions (snv)*	0,24138
edge insertions/deletions (se)	11
rata-rata *edge insertions/deletions (sev)*	0,31429
rata-rata *node substitutions (sbv)*	0,88923
graph edit distance similarity	0,51837

Tabel 3.2 Hasil penghitungan TDP dan JPO

Penghitungan	Hasil
node insertions/deletions (sn)	7
rata-rata *node insertions/deletions (snv)*	0,24138
edge insertions/deletions (se)	11
rata-rata *edge insertions/deletions (sev)*	0,31429
rata-rata *node substitutions (sbv)*	0,88923
graph edit distance similarity	0,51837

IV. PEMBAHASAN

Dari hasil pengujian, didapatkan perbandingan nilai *similarity* yang terjadi diantara proses bisnis seperti yang ditunjukkan pada tabel 4.5:

Tabel 4.1 Hasil penghitungan TDP dan JPO

Perbandingan		*Graph Edit Distance Similarity*
Proses bisnis 1	Proses bisnis 2	
TDP	TDP	1
TDP	JPO	0,51837
JPO	JPO	1
JPO	TDP	0,51837

Tabel perbandingan 4.1 didapatkan menggunakan metode *greedy graph matching* dimana metode tersebut memilih pasangan *node* dengan nilai *syntactic similarity* tertinggi agar dapat meningkatkan nilai *graph edit distance similarity*. Dalam prosesnya sistem ini melibatkan penghitungan *syntactic similarity*, *node insertions/deletions (sn),* rata-rata *node insertions/deletions (snv), edge insertions/deletions (se),* rata-rata *edge insertions/deletions (sev)* dan rata-rata *node substitutions (sbv).*

Menurut penulis, *greedy graph matching* tidak efektif digunakan untuk menghitung *graph edit distance similarity* karena dalam prosesnya, setiap iterasi *greedy* hanya melihat nilai *syntactic similarity* tertinggi dalam *node* yang sama, selanjutnya pasangan *node* yang telah dipilih kemudian dihapus. Padahal mungkin saja diiterasi selanjutnya *node* yang telah dihapus tersebut memiliki pasangan dengan nilai *syntactic similarity* yang lebih tinggi [13].

Berdasarkan tabel 4.1 juga ditunjukkan bahwa penetapan proses bisnis sebagai proses bisnis 1 atau sebagai proses bisnis 2 tidak berpengaruh terhadap hasil pengecekan *graph edit distance similarity*. Dapat dilihat proses bisnis 1, yaitu JPO dengan proses bisnis 2, yaitu TDP menghasilkan nilai *graph edit distance similarity* 0,51837, begitu pula

jika kedua proses bisnis tersebut dibalik maka menghasilkan nilai *graph edit distance similarity* yang sama yaitu 0,51837.

Nilai *similarity* proses bisnis JPO dan TDP bisa saja lebih tinggi jika menggunakan metode lain. Seperti yang sudah dikatakan sebelumnya, *greedy graph matching* tidak efektif pada sistem ini. Secara kasat mata, 11 label pada proses bisnis JPO dan TDP memiliki nilai *syntactic similarity* 1. Namun, pada sistem ini, sesuai dengan proses yang dilakukan oleh *greedy*, cuman 5 label saja dengan nilai *syntactic similarity* 1 yang dipasangkan . Itulah bentuk tidak efektifnya *greedy* pada sistem ini.

V. KESIMPULAN

Berdasarkan hasil pengujian dan analisis yang telah dipaparkan sebelumnya, maka dapat ditarik kesimpulan sebagai berikut:

1. Penghitungan *similarity* menggunakan *greedy graph matching* dimana pada setiap iterasi, *greedy* mencari pasangan *node* dengan nilai *matching score* paling optimal (nilai *syntactic similarity* yang paling tinggi) kemudian pasangan *node* tersebut dihapus agar tidak dipilih lagi pada iterasi selanjutnya. Hal ini terus berulang sampai tidak ada lagi pasangan *node* yang dianggap dapat meningkatkan nilai *similarity*. Kemudian, dihitung rata-rata *node substitutions (sbv), node insertions/deletions (sn), edge insertions/ deletions (se)* untuk mendapatkan *graph edit distance similarity* (nilai *similarity*).
2. *Greedy graph matching* tidak efektif digunakan untuk menghitung *graph edit distance similarity* karena dalam prosesnya, setiap iterasi *greedy* hanya melihat nilai *syntactic similarity* tertinggi pada *node* yang sama, selanjutnya pasangan *node* yang telah dipilih kemudian dihapus. Padahal mungkin saja diiterasi selanjutnya *node* yang telah dihapus tersebut memiliki pasangan dengan nilai *syntactic similarity* yang lebih tinggi.
3. Pengecekan *similarity* menggunakan *greedy graph matching* menghasilkan persentase *similarity* yang dimana jika kedua proses bisnis tersebut dibalik sebagai proses bisnis 1 maupun sebagai proses bisnis 2, tetap sama.

VI. REFERENSI

[1] BPPT Kota Bandung, "Prosedur Izin Pemancangan Tiang pancang Jembatan Penyeberangan Orang (JPO)/Reklame," [Online]. Available: http://app.boss.or.id/#. [Diakses 13 Februari 2015].

[2] BPPT Kota Bandung, "Prosedur Izin Tanda Daftar Perusahaan," [Online]. Available: http://app.boss.or.id/#. [Diakses 13 Februari 2015].

[3] C. Scharf, Acquisitions Merges Sales B/T, New Jersey: Prentice Hall, 1985.

[4] F. Andhika, "Penerapan String Suggestion dengan Algoritma Levenshtein Distance dan Alternatif Algoritma Lain dalam Aplikasi," 6 Desember 2010. [Online]. Available: http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2010-2011/Makalah2010/MakalahStima2010-100.pdf. [Diakses 17 Maret 2015].

[5] F. Jones dan D. Rama, Accounting Information Systems: A Business Process Approach, South-Western College, 2002.

[6] Lucid Software Inc., "BPMN Symbols Explained," 2015. [Online]. Available: https://www.lucidchart.com/pages/bpmn-symbols-explained. [Diakses 10 April 2015].

[7] Lucid Software Inc., "Connecting Objects," 2015. [Online]. Available: https://www.lucidchart.com/pages/bpmn/connecting-objects. [Diakses 10 April 2015].

[8] Lucid Software Inc., "Event," 2015. [Online]. Available: https://www.lucidchart.com/pages/bpmn/events. [Diakses 10 April 2015].

[9] Lucid Software Inc., "Gateway Types," 2015. [Online]. Available: https://www.lucidchart.com/pages/bpmn/gateways. [Diakses 10 April 2015].

[10] M. Weske, Business Process Management Concepts, Languages, Architectures, Springer-Verlag Berlin Heidelberg, 2012.

[11] R. Dijkman, M. Dumas dan L. Garcia-Banuelos, "Graph Matching Algorithms for Business Process Model Similarity Search," dalam *Business Process Management*, Germany, Springer Berlin Heidelberg, 2009, pp. 48-63.

[12] R. Dijkman, M. Dumas, B. Dongen, R. Kaarik dan J. Mendling, "Similarity of Business Process Models: Metrics and Evaluation," *Information Systems*, vol. 36(2), no. 10.1016/j.is.2010.09.006, pp. 498-516, 2011.

[13] R. Dijkman, M. Dumas, L. Garcia-Banuelos dan R. Kaarik, "Aligning Business Process Models," dalam *Enterprise Distributed Object Computing Conference, 2009. EDOC '09. IEEE International*, Auckland, 2009.

[14] T. Jewett, "SQL Technique: Views and Indexes," [Online]. Available: http://www.tomjewett.com/dbdesign/dbdesign.php?page=views.php. [Diakses 10 April 2015].

[15] Techopedia TM., "Data Preprocessing," [Online]. Available: http://www.techopedia.com/definition/14650/data-preprocessing. [Diakses 10 April 2015].