

PENGUNAAN *WORKFLOW REPOSITORY* DAN KOMBINASI METODE TEMU KEMBALI BERPRIORITAS UNTUK MENINGKATKAN PERFORMA CONFIGURABLE-PROCESS DI ERP: STUDI KASUS ERP2011/13

Rigga Widar Atmagi¹⁾, Riyanarto Sarno²⁾

Teknik Informatika, Fakultas Teknologi Informasi¹⁾²⁾
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia

Email: rigga.atmagi11@mhs.if.its.ac.id¹⁾

ABSTRAK

Kekuatan ERP terletak dari banyaknya kebutuhan *workflow* yang disediakan. Perusahaan yang menggunakan perangkat lunak ERP menginginkan investasi yang telah dikeluarkan mampu untuk mendukung bisnis yang terus bertransformasi. Mempertimbangkan hal ini, maka kami mengusulkan sebuah metode yang disebut tempat penyimpanan *workflow* (*workflow repository*). *Workflow* yang disimpan dapat berbentuk *workflow* model dan dapat digunakan secara pribadi atau dibagikan kepada publik. Hal ini tergantung pada pengembang *workflow*. *Workflow* yang disimpan dipetakan menggunakan atribut yang mirip dengan OWL-s dan atribut kualitas *workflow*. Pengguna akhir mendapatkan *workflow* tersebut kembali dengan mempertimbangkan aspek dari perhitungan kemiripan OWL-s, kualitas, konteks dan struktur. Untuk mendapatkan *workflow* kembali, *workflow repository* menggunakan teknik prioritasasi. Teknik prioritasasi memiliki akurasi yang baik dan cocok digunakan karena memberikan garansi hasil yang bebas dari *workflow* yang tidak dapat dieksekusi dari langkah pertama. Dengan metode ini, perusahaan yang telah berinvestasi pada perangkat lunak ERP dapat menggunakan akses *workflow* yang tersedia atau menjalankan *workflow* bersama dengan perusahaan lain. Percobaan untuk penelitian ini menggunakan ERP2011/13 dengan data tes skenario *workflow* model yang telah dikondisikan. Hasil yang didapatkan adalah tingkat temu kembali (*precision*= rata-rata < 0.5, *recall*=1) dan kemudahan akses dibanding mengelola *workflow* secara konvensional. Kesimpulan, didapatkan bahwasannya metode prioritasasi *workflow* memiliki presisi yang lebih tinggi.

Kata kunci: *Workflow, Workflow Repository, Similarity*

ABSTRACT

*ERP strength lies in how many process variations that can be offered to support user requirement/agility. Several companies, which used ERP software, expect it can support their business, which continue to transform. Considering that reason, we developed a method for utilizing an entity as workflow ERP storing. This entity has known as workflow repository. Our repository store workflow as workflow model. Each stored workflow should be shared or private depend developer preferences. Workflow model will be mapped using OWL-s similar approach and workflow quality attribute. End user should find their wanted workflow by our formula, which consider calculation from OWL-s similarity, quality, context, and workflow structure. For retrieval process, we used prioritization method. Prioritization has a better accuracy and is more appropriate to use. Prioritization guarantee workflow that is resulted from its process is free of un-executable workflow and tackles various variations among workflow from beginning. Other instances ERP that is installed in different company could access workflow privately or shared running workflow among them. This experiment conduct by utilizing ERP2011/13 and using conditioned workflow model scenario. Result is retrieval ratio (*precision*=average<0.5 and *recall*=1) and ease of access than conventional one.*

Keywords: *Workflow, Workflow Repository, Similarity*

1. Pendahuluan

Pada saat ini, era perangkat lunak dengan skala *enterprise* terdapat minimal 3 fakta sebagai berikut:

1. Perangkat lunak ERP merupakan investasi bagi perusahaan. Banyak perusahaan menggunakan ERP untuk mengelola *workflow* internal perusahaan.
2. Setiap perusahaan memiliki kebutuhan yang berbeda. Sedangkan, perangkat lunak ERP yang dijual pada umumnya memberikan modul *workflow* yang bersifat umum.
3. Kustomisasi *workflow* dilakukan oleh perusahaan itu sendiri. Pada perusahaan yang melakukan transformasi dalam jangka pendek, memiliki kemungkinan yang tinggi terhadap perubahan proses. Perubahan *workflow* yang sering berubah-ubah akan menjadi beban biaya dan waktu.

Workflow merupakan hal yang paling menarik pada perkembangan teknologi SOA. Dengan menggunakan *workflow*, pengguna dapat melakukan intervensi/perubahan alur kerja sistem secara komposisi/orkestrasi. Akan tetapi kuantitas *workflow* dalam perangkat lunak yang digunakan perusahaan adalah besar. Setiap modul mungkin memiliki *workflow* yang mirip/variasi dari *workflow* yang lain. Untuk terus mendukung pemeliharaan dan perbaikan secara berkelanjutan disarankan untuk menggunakan *workflow repository*.

Workflow repository mempunyai peran yang lebih dari sekedar penyimpanan proses. *Workflow repository* akan bertindak sebagai *Shared Processing Center (SPC)* dimana seluruh *workflow* yang menjalankan kegiatan perusahaan melalui perangkat lunak ERP terkandung didalamnya. Penggunaan *workflow repository* mengedepankan konsep *reuse*. *Workflow repository* memiliki beberapa fitur sebagai berikut:

1. Penyimpanan *workflow* model menggunakan metadata. Metadata *workflow* dibangun menggunakan

konsep ontologi yang kemudian dipetakan ke dalam *database*.

2. Pengelompokan *workflow* yang memiliki kesamaan. Konsep pengelompokan *workflow* menggunakan *hierarchal clustering*.
3. Rekomendasi *workflow* dengan konsep kemiripan dan variasi. Konsep kemiripan dengan membandingkan *workflow*, dapat mempertimbangkan 3 paradigma, yaitu kemiripan sintaksis, semantik dan kontekstual [1]. Konsep variasi dapat menggunakan *Tree Longest Common Subsequences (T-LCS)* dan *micro-macro workflow variability design*.

Kebutuhan *workflow repository* pada umumnya mengikuti kebutuhan temu kembali *workflow*. Semakin kompleks prosedur temu kembalinya, maka semakin banyak data yang harus disimpan di metadata *workflow* tersebut. *Workflow repository* yang hanya menyimpan label *node* tidak menjamin hasil yang akurat walaupun mudah untuk diimplementasikan. Karena jumlah variasi dan kebutuhan mempercepat temu kembali, *workflow* dikelompokan sesuai dengan kesamaan strukturnya [2] [3].

Perhitungan kemiripan *workflow* telah dievaluasi oleh penelitian sebelumnya. Beberapa penelitian mengedepankan kemiripan struktur (*structure similarity*), label (*syntactic similarity*), sifat (*behavioural similarity*) atau konteks (*contextual similarity*) [1] [4] [5]. Metode kombinasi dengan membobotkan masing-masing hasil kemiripan metode masih menghasilkan kesalahan dalam eksekusi *workflow* yang direkomendasikan.

Pendekatan kemiripan deskripsi tidak selalu menjamin kebenarannya. *Workflow, service* ataupun *method* memiliki karakteristik yang unik. Kemiripan deskripsi yang diterapkan di pencarian kembali berbasis teks sulit untuk diterapkan di lingkungan SOA. *Workflow, service, ataupun method* hanya dapat berjalan dengan baik jika masukan dan prakondisinya tepat. Dengan kata lain, rekomendasi kemiripan deskripsi menuntut

hasil yang tinggi (lebih dari 0.9).

Pendekatan kemiripan struktur sebagai contoh pada metode *graph edit distance* [6] [5], akan meleset jika *workflow* yang dibandingkan menerapkan enkapsulasi yang sangat ketat. Nilai kemiripan yang akan dihasilkan bisa saja sangat rendah walaupun sebenarnya *workflow* tersebut bertujuan mempunyai *input* yang sama.

Oleh karena itu, keempat metode tersebut tidak dapat dihitung secara terpisah dan diberi bobot pada perhitungan akhirnya. Makalah ini mengusulkan kombinasi metode temu kembali *workflow* dengan memberikan prioritas langkah pada setiap tahapannya. Ada 4 metode yang dikombinasikan, yaitu kemiripan deskripsi, konteks, struktur dan kualitas.

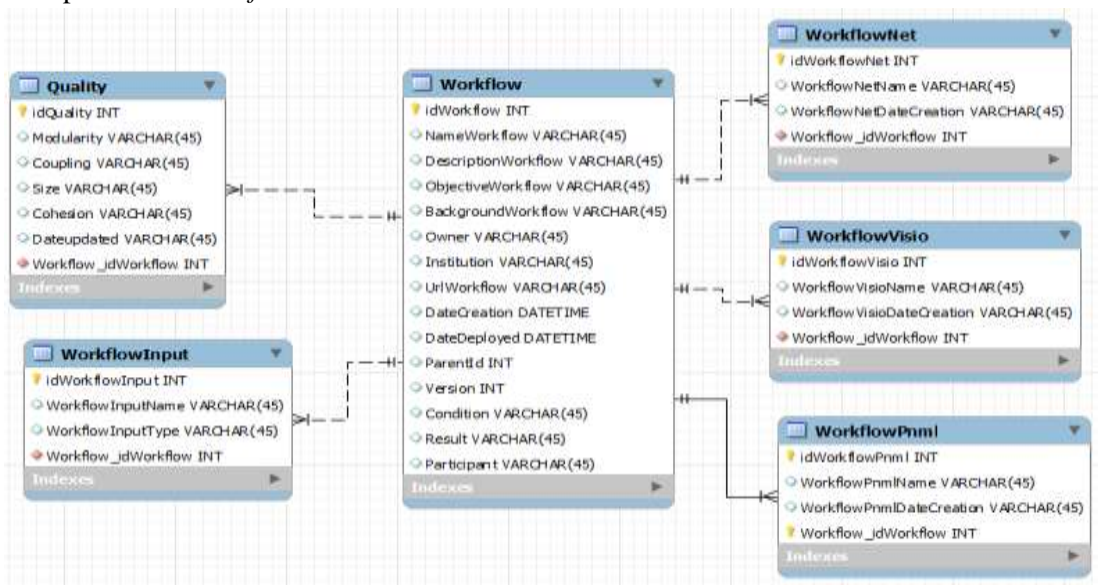
2. Penyimpanan Metadata *Workflow*

Langkah pertama adalah melakukan inventarisasi *workflow* pada ERP2011/13. Atribut *workflow* akan disimpan dalam metadata yang ditunjukkan pada Gambar 1 *Workflow* memiliki *precondition*, *mid-process*, dan *post condition*. *Mid-process* diturunkan berdasarkan atribut OWL-s, yaitu *profile*, model dan *grounding* serta tambahan atribut lain. Selain OWL-s, terdapat atribut *workflow* model dan atribut

kualitas *workflow*. Langkah kedua adalah memetakan onologi kedalam skema *database*, *workflow repository* menggunakan *database* terpusat dengan *framework* MSSQL Server 2008 R2.

- ***Workflow has_precondition condition***
- ***Workflow has_result result***
- ***Workflow has_participant participant***
- ***Workflow has_profile, model, grounding, fork_from and version***
Profile has name, description, objective, background, owner, institution, and url_location
Model has input_type and output_type.
Grounding has input_name and output_name.
- ***Workflow has_attached workflow_model_pnml, workflow_model_visio and workflow_model_net***
Workflow_model_pnml, workflow_model_visio and workflow_model_net has name, date_creation
- ***Workflow has_valued_of workflow_quality***
Workflow_quality has_indicator_of size, coupling, cohesion, modularity, date_updated

Gambar 1. Bentuk ontologi sebuah metadata *workflo*

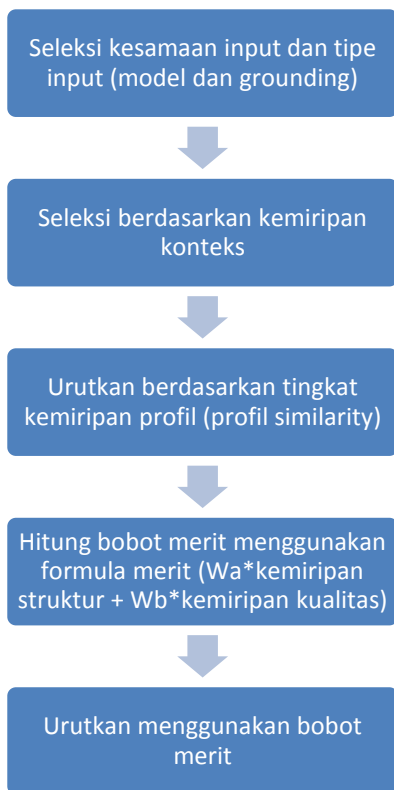


Gambar 2. Hasil pemetaan ontologi metadata *workflow* dalam bentuk skema *database*

3. Metode Kombinasi Dan Prioritas

Metode kemiripan merupakan metode yang digunakan pada permasalahan temu kembali, khususnya pada pemroses teks. Metode kemiripan yang diusulkan adalah kemiripan struktur, OWL-s, kualitas dan konteks.

Prioritas adalah salah satu solusi untuk meningkatkan kemungkinan dimana *workflow* hasil rekomendasi dapat selalu digunakan dan tepat guna. Kemiripan *workflow* dan kemiripan teks memiliki sifat yang berbeda. *Workflow* hanya dapat dieksekusi jika masukannya tepat. *Workflow* yang dieksekusi dikatakan tepat guna jika menghasilkan *output* yang diinginkan. Proses rekomendasi mengikuti diagram alir yang ditunjukkan oleh Gambar 3.



Gambar 3. Diagram alir proses untuk menghasilkan rekomendasi *workflow* yang tepat eksekusi dan tepat guna

A. Menghilangkan *Workflow* Yang Tidak Bisa Dieksekusi

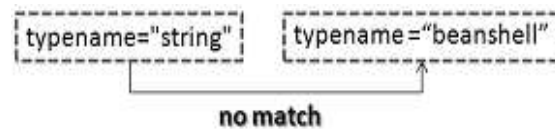
Hipotesis pertama dijawab pada tahap pertama, himpunan *workflow* akan diseleksi berdasarkan atribut model dan *grounding*. Atribut model dan *grounding* terdiri dari

nama dan tipe *input output* sebuah *workflow/service*. Hasil tahap pertama tidak diperingkat.

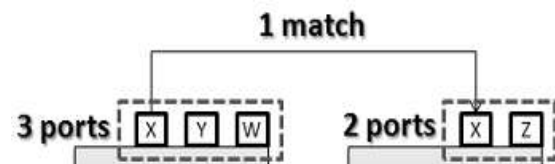
Jika nama dan tipe *input-output workflow* sama maka *workflow* tersebut dikatakan “match”. Jika tidak sama, sebagai justifikasi akan dihitung nilai proporsionalnya (*Proportional factor*) α .

4. Nilai kemiripan tipe *input-output* berkisar dari 0 atau 1, digambarkan pada Gambar 4.
5. Kemiripan antar jumlah *input* yang dimiliki *workflow* seperti contoh pada Gambar 5 dihitung menggunakan Persamaan (1), nilai proporsional (*proportional factor*). IP_A adalah jumlah parameter (disebut *port*). IP_A berarti jumlah *port* yang dimiliki oleh A. *Matches* adalah jumlah *port* yang bertipe sama dari A dan A’.

$$\frac{(IP_A + IP_{A'}) - (2 \times matches)}{IP_A + IP_{A'}} \quad (1)$$



Gambar 4. Contoh perbandingan tipe *input-output workflow*



Gambar 5. Contoh perbandingan jumlah *input workflow*

B. Mendapatkan *Workflow* Yang Dapat Dieksekusi Dan Memiliki Konteks Yang Sama

Hipotesis kedua dijawab dengan menghitung kemiripan konteks *workflow* tersebut. Jika kemiripan konteks lebih rendah dari *threshold* maka *workflow* akan dikeluarkan dari himpunan rekomendasi. Hasil tahap kedua tidak diperingkat.

Workflow yang disimpan memiliki *property* `Workflow_model_pnml`. Dengan

menggunakan tool BeehiveZ, dapat dihasilkan hasil kemiripan kontekstual [7].

C. Mempermudah Pengguna Menemukan Workflow Yang Tepat Guna Dari Sekumpulan Workflow Yang Tepat Eksekusi

Hipotesis ketiga dijawab pada tahapan ketiga akan memanfaatkan atribut *profile*. Atribut *profile* adalah atribut yang menerangkan deskripsi *workflow* tersebut sebagai contoh adalah nama, alamat, dan deskripsi *workflow*. Perhitungan dilakukan secara sintaksis dan hasil pencarian diperingkat. Kemiripan tertinggi adalah peringkat pertama.

Atribut *profile* memiliki banyak item didalamnya, seperti yang tertulis pada Persamaan (2). Hasil kemiripan atribut *profile* (β) adalah hasil rata-rata dari seluruh jumlah total kemiripan masing-masing item. Kemiripan *profile* dihitung menggunakan komputasi WordNet [8].

$$\beta = \frac{(\sum_{k=0}^n Item_k)}{n} \quad (2)$$

D. Memberikan Hasil Yang Berkualitas Dari Workflow Yang Tepat Guna Dan Tepat Eksekusi

Hipotesis terakhir akan dijawab dengan perhitungan kemiripan struktur dan kualitas secara komputasional. Hasil kemiripan struktur dan kualitas akan diberikan bobot yang berjumlah 1. Hasil akhirnya adalah peringkat *workflow* berdasarkan ketepatan guna, ketepatan eksekusi dan ketercapaian ekspektasi kualitas yang diharapkan.

Kemiripan struktur dan kualitas dihitung menggunakan *jaccard strcuture similarity* menggunakan tool BeehiveZ [7]. Hasil kemiripan struktur dan kualitas diberi bobot dan dijumlahkan. Nilai ini adalah atribut akan melengkapi ranking dari tahap ke-3. Ranking pertama adalah *entry* dengan nilai tahap ke-3 dan tahap ke-4 tertinggi. Ranking terendah adalah *entry* dengan nilai tahap ke-3 dan tahap ke-4 terendah.

4. Hasil Uji Coba

Untuk menghitung *precision* dan *recall*, digunakan himpunan *workflow* pada aplikasi ERP2011/13 (lihat Tabel 1).

Perhitungan presisi sebagaimana tertera pada Persamaan (3) adalah hasil bagi dari *true* positif (hasil yang direkomendasikan benar) dengan jumlah dari *true* positif dan *false* positif (hasil yang tidak direkomendasikan tetapi dinyatakan benar oleh sistem). Sedangkan *recall* pada Persamaan (4) adalah hasil bagi dari *true* positif dengan penjumlahan *true* positif dan *false* negatif (himpunan bukan hasil benar).

Tabel 1 Dataset *workflow*

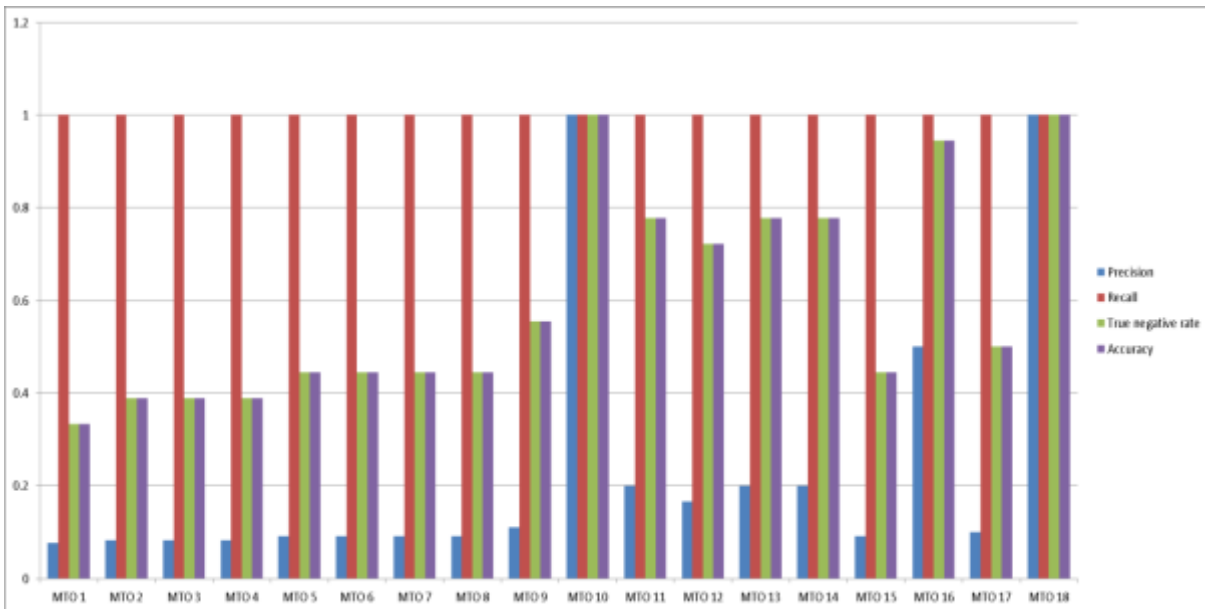
Kode	Workflow	Format
MTO 1	WF_AssignItem	XAMLX
MTO 2	WF_calculateEOQ_ROP_SS	XAMLX
MTO 3	WF_CloseProduction	XAMLX
MTO 4	WF_CompareWarehouseItemBalance	XAMLX
MTO 5	WF_CreatingDB	XAMLX
MTO 6	WF_FinalizeSalesOrder	XAMLX
MTO 7	WF_GetActualItemBalance	XAMLX
MTO 8	WF_GetItemBalance	XAMLX
MTO 9	WF_GetPlanItemBallance	XAMLX
MTO 10	WF_GetWinner	XAMLX
MTO 11	WF_InitializeBid	XAMLX
MTO 12	WF_InsertCustomerPayment	XAMLX
MTO 13	WF_InsertFAAdditional	XAMLX
MTO 14	WF_InsertPurchaseInvoice	XAMLX
MTO 15	WF_InsertReceivingBatch	XAMLX
MTO 16	WF_InsertReceivingFropPurchOr	XAMLX
MTO 17	WF_InsertSalesInvoice	XAMLX
MTO 18	WF_InsertSalesOrderDetail	XAMLX
MTO 19	WF_InsertSalesReturnDetail	XAMLX
MTO 20	WF_IssuedItem	XAMLX
MTO 21	WF_NumberRackDetail	XAMLX
MTO 22	WF_PostJournal_CB	XAMLX
MTO 23	WF_PostJournal_COGS	XAMLX
MTO 24	WF_PostJournal_FG	XAMLX
MTO 25	WF_PostJournal_IV	XAMLX
MTO 26	WF_PostJournal_MF_FG	XAMLX
MTO 27	WF_PostJournal_WIPa	XAMLX
MTO 28	WF_processSalesReturn	XAMLX
MTO 29	WF_ScheduleMachine	XAMLX
MTO 30	WF_SetProductionPre-executionInitials	XAMLX
MTO 31	WF_UpdateInItemBalance	XAMLX
MTO 32	WF_UpdateInItemBalanceByPurchReq	XAMLX
MTO 33	COGM	XAMLX
MTO 34	COGS	XAMLX
MTO 35	gettingActualItemBalance	XAMLX
MTO 36	gettingItemBalance	XAMLX
MTO 37	gettingPlanItemBallance	XAMLX

$$Precision = TP / (TP + FP) \tag{3}$$

$$Recall = TP / (TP + FN) \tag{4}$$

Workflow tersebut merepresentasikan workflow yang didukung oleh system ERP. Masing-masing workflow akan ekstrak atributnya dan simpan ke dalam workflow repository sesuai dengan metadata yang dibangun. Masing-masing workflow yang disimpan, disebut sebagai entry, akan dijadikan sebagai query search. Dengan kata lain workflow akan dibandingkan dengan workflow lainnya. Kutipan hasil di tampilkan di Tabel 32.

Merujuk pada Gambar 6 nilai recall mencapai 1. Nilai ini mencerminkan metode yang diusulkan memberikan keluaran hasil yang paling relevan. Sekali lagi ditekankan, proses temu kembali workflow memiliki karakteristik unik dibandingkan dengan temu kembali workflow. Workflow yang direkomendasikan/True Positif, lihat Tabel 23, adalah workflow yang tepat eksekusi (dapat dieksekusi sesuai dengan input dan output yang dipersyaratkan) dan tepat guna (keluarannya sesuai dengan apa yang diinginkan).



Gambar 6. Hasil ROC uji coba

Tabel 2. Kutipan hasil pencarian menggunakan dataset

	MTO 1	MTO 2	MTO 3	MTO 4	MTO 5	MTO 6	MTO 7
MTO 1	0.75	0.65	0.65	0.65	0.63	0.65	0.63
MTO 2	0.65	0.75	0.68	0.68	0.66	0.66	0.69
MTO 3	0.65	0.68	0.75	0.68	0.66	0.66	0.66
MTO 4	0.65	0.68	0.68	0.75	0.69	0.66	0.66
MTO 5	0.63	0.66	0.66	0.69	0.75	0.68	0.68
MTO 6	0.65	0.66	0.66	0.66	0.68	0.75	0.68
MTO 7	0.63	0.69	0.66	0.66	0.68	0.68	0.75
MTO 8	0.63	0.66	0.69	0.66	0.68	0.68	0.68
MTO 9	0.53	0.56	0.57	0.56	0.57	0.58	0.58
MTO 10	0.40	0.40	0.39	0.40	0.40	0.40	0.40
MTO 11	0.56	0.49	0.49	0.49	0.48	0.50	0.48
MTO 12	0.51	0.49	0.49	0.54	0.50	0.48	0.48

Tabel 3. Kutipan hasil perhitungan presisi, recall, aaccuracy (akurasi)

	MTO 1	MTO 2	MTO 3	MTO 4	MTO 5	MTO 6
Precision	0.076923077	0.083333	0.083333	0.083333	0.090909	0.090909
Recall	1	1	1	1	1	1
True negative rate	0.333333333	0.388889	0.388889	0.388889	0.444444	0.444444
Accuracy	0.333333333	0.388889	0.388889	0.388889	0.444444	0.444444

5. Kesimpulan

Pencarian kembali dari *workflow repository* merupakan hal yang penting bagi pengguna. Rekomendasi *workflow* memberikan informasi kepada pengguna untuk melakukan *reuse*. Dengan melakukan *reuse*, penggunaan *workflow* akan lebih efisien dan efektif.

Pada penelitian ini diusulkan cara memberikan rekomendasi *workflow* secara komputasional. Penelitian ini menggunakan kombinasi 4 metode kemiripan, yaitu struktur, OWL-s, kualitas dan konteks dengan mengedepankan prioritas pada setiap langkahnya. Dari hasil ujicoba, dapat disimpulkan bahwa metode yang diusulkan dapat menghilangkan kesalahan pencarian *workflow* yang tidak bisa eksekusi dan menghasilkan *recall* yang mencapai 1. Sehingga metode yang diusulkan dapat digunakan dengan baik.

Daftar Pustaka

- [1] R. Dijkman, M. Dumas and B. v. Dongen [2010], "Similarity of Business Process Models: metrics and Evaluation," *Information Systems*, vol. 36, no. 2011.
- [2] C. J. Kim, H. S. Chung and S. E. Cho [2008], "Micro And Macro *Workflow* Variability Design Techniques Of Component," *Information and Software Technology*, vol. 50, no. 4.
- [3] Y. J. Jung, S. J. Bae and L. Liu [2009], "Hierarchical Clustering Of Business Process Models," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 12.
- [4] V. Silva, F. Chirigati, K. Maia, E. Ogasawara, D. d. Oliveira, V. Braganholo, L. Murta and M. Mattoso [2011], "Similarity-based *workflow* clustering," *Journal of Computational Interdisciplinary Sciences*.
- [5] B. v. Dongen, R. Dijkman and J. Mendling [2008], "Measuring Similarity between Business Process Models," *Proceedings of the 20th international conference on Advanced Information Systems Engineering*.
- [6] A. Wombacher and C. Li [2010], "Alternative approaches for *workflow* similarity," *IEEE International Conference on Services Computing*.
- [7] T. Jin, J. Wang and L. Wen [2011], "Efficiently Querying Business Process Models with BeehiveZ.," *In proceeding of: Proceedings of the Demo Track of the Nineth Conference on Business Process Management*.
- [8] L. Chen, G. Yang, D. Wang and Y. Zhang [2010], "WordNet-powered Web Services Discovery Using Kernel-Based Similarity Matching Mechanism," *Service Oriented System Engineering (SOSE)*.

