

CLUSTERING BUSINESS PROCESS MODEL PETRI NET DENGAN COMPLETE LINKAGE

Fakhrunnisak Nur Aini

Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami No. 36 A Surakarta
fakhrunnisak@students.uns.ac.id

Sarngadi Palgunadi

Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami No. 36 A Surakarta
palgunadi@uns.ac.id

Rini Anggrainingsih

Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami No. 36 A Surakarta
rinianggra@gmail.com

ABSTRAK

Petri net merupakan salah satu notasi yang banyak digunakan sebagai *business process model*. Pemodelan *business process* dapat digunakan untuk membantu mempermudah pengembangan *software* dengan melakukan *clustering* pada model yang telah dibuat. Penelitian ini dilakukan untuk mengetahui bagaimana hasil *clustering business process* yang dimodelkan dengan *petri net* menggunakan *complete linkage*. Penelitian ini bertujuan untuk memodelkan *business process* dengan *petri net*, *clustering* berdasarkan strukturnya, dan validasi hasil *cluster* sehingga dapat dimanfaatkan oleh penelitian selanjutnya untuk melakukan implementasi modul yang dibutuhkan.

Dalam penelitian ini, 62 prosedur fakultas sub bagian keuangan dan kepegawaian Universitas Sebelas Maret digunakan sebagai *business process* yang dimodelkan dengan *petri net*. Setiap model divalidasi berdasarkan kriteria *structural analysis*, *soundness*, *liveness*, dan *boundedness*. *Clustering* dilakukan terhadap model yang valid dengan *complete linkage* menggunakan nilai *structural similarity* antar model. Pemotongan diterapkan pada dua titik untuk memperoleh jumlah *cluster* yang bervariasi. Hasil *clustering* dari setiap titik pemotongan divalidasi dengan *silhouette* untuk menentukan jumlah *cluster* yang paling optimal.

Hasil penelitian menunjukkan bahwa dari 62 prosedur, terdapat 50 model *petri net* valid yang digunakan untuk *clustering* dengan *complete linkage*. Pemotongan diterapkan pada titik 0.3 dan 0.4 yang menghasilkan 15 *cluster* dan 18 *cluster*. *Average Silhouette* pada titik 0.3 adalah 0.78, sedangkan pada titik 0.4 adalah 1. Dari nilai *average silhouette* dapat diketahui bahwa pemotongan pada titik 0.4 menghasilkan jumlah *cluster* yang paling optimal.

Kata Kunci: *Business process*, *Clustering*, *Complete linkage*, *Petri net*, *Structural Similarity*.

1. PENDAHULUAN

Petri net merupakan salah satu notasi *business process model* yang dibuat oleh Carl Adam Petri pada tahun 1962. *Petri net* mempunyai beberapa keunggulan yaitu proses dapat dideskripsikan secara grafis, memiliki pemodelan matematis yang kuat, dan memiliki banyak teknik analisis seperti *soundness*, *liveness*, *boundedness*, dan lain sebagainya. *Petri net* juga mudah dimengerti dan sangat memungkinkan untuk mengubah notasi *business process model* lain ke bentuk *petri net* [1]. *Petri net* banyak digunakan oleh peneliti untuk memodelkan *business process* (proses bisnis) [2].

Clustering terhadap proses bisnis dengan *petri net* sudah banyak dilakukan. *Clustering* proses bisnis bertujuan untuk mengelompokkan *business process model* yang mirip sehingga dapat membantu mempermudah pengembangan *software*. Hasil *clustering* dapat digunakan untuk menentukan jumlah modul yang akan dikerjakan pada saat pengembangan *software*. Pengembangan modul dilakukan dengan menganalisis pola proses bisnis yang berada dalam satu *cluster*, kemudian dapat dibuat modul secara umum [1].

Untuk mengembangkan modul diperlukan langkah pemodelan proses bisnis, *clustering* dan validasi *cluster*. Salah satu teknik *clustering* yang paling banyak digunakan adalah algoritma *hierarchichal* [3]. Kelebihan *hierarchichal* yaitu sangat fleksibel dalam penggunaan matriks *similarity* atau matriks *dissimilarity* [3]. Algoritma *hierarchichal* memiliki beberapa cara pengkombinasian *cluster* yang dinamakan *linkage* [3]. *Linkage* yang paling terkenal adalah *single linkage*, *complete linkage*, dan *average linkage*. Penentuan metode *linkage* yang sesuai dipengaruhi oleh sifat data [3]. Agar suatu proses bisnis yang akan dikombinasikan mempunyai nilai *similarity* dengan setiap individu yang sudah ada dalam *cluster*, maka *complete linkage* adalah yang paling sesuai. *Complete linkage* dapat mencegah pengkombinasian *process model* menjadi sebuah *cluster* tanpa adanya *similarity* dengan setiap data yang sudah ada dalam *cluster* [3].

Clustering proses bisnis dengan algoritma *hierarchichal* telah dilakukan oleh Jung [4]. Perhitungan nilai *structural similarity* antar *process model* dilakukan dengan *Cosine coefficient* menggunakan vektor *transition* dan *activity*. Penggunaan vektor ini kemungkinan tidak sesuai dengan *real world business process* [5]. Untuk menutupi kelemahan dalam penelitian tersebut, Bae [6] melakukan perhitungan *structural similarity* antara *process model* dengan menganalisis *dependency graph*. Perhitungan *similarity* dengan *dependency graph* telah dilakukan oleh Wang [7] dan Sarno [1]. Dalam penelitian tersebut *dependency graph* digantikan dengan matriks *incidency*. Perhitungan dengan pendekatan ini memiliki hasil yang bagus dalam menentukan jarak antara model satu dengan model yang lain [7].

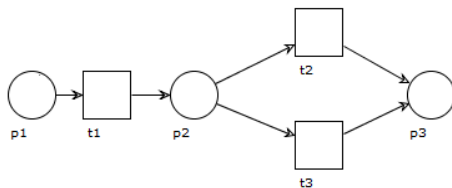
Pada penelitian ini perhitungan *structural similarity* dilakukan dengan matriks *incidency* sebagai *dependency graph*. Matriks *incidency* dihasilkan dari pemodelan bisnis proses dengan *petri net* menggunakan prosedur fakultas sub bagian keuangan dan kepegawaian Universitas Sebelas Maret. Matriks *incidency* digunakan untuk menentukan nilai *structural similarity* yang selanjutnya dikelompokkan menggunakan *complete linkage*. Penelitian ini berfokus pada pemodelan *petri net*, *clustering* berdasarkan strukturnya, dan

validasi hasil *cluster*. Penelitian ini merupakan langkah awal dari beberapa langkah yang dibutuhkan untuk mengembangkan modul pada pengembangan *software*.

2. LANDASAN TEORI

2.1 PETRI NET

Petri net dibuat oleh Carl Adam Petri pada tahun 1962 sebagai alat pemodelan dan analisis proses [8]. *Petri net* terdiri dari *place* dan *transition* yang dihubungkan oleh sebuah garis (*arc*). *Place* menggambarkan kondisi yang harus dipenuhi sebelum suatu tindakan dapat dilakukan. *Transition* menggambarkan suatu peristiwa atau tindakan. Gambar 1 merupakan contoh pemodelan *petri net*.



Gambar 1. Contoh *Petri net*

Petri net dapat direpresentasikan dengan matriks *incidency*. Matriks *incidency* terdiri dari matriks baris yang menyatakan *place* dan kolom menyatakan *transition*. Matriks bernilai 1 apabila *transition* menghasilkan output ke *output place*, dan bernilai -1 apabila *transition* mendapatkan input dari *input place*, selain itu nilai matriks adalah 0. M1 merupakan representasi pemodelan *petri net* pada Gambar 1 dengan matriks *incidency*.

$$M1 = \begin{bmatrix} -1 & 0 & 0 \\ 1 & -1 & -1 \\ 0 & 1 & 1 \end{bmatrix}$$

2.2 SIMILARITY BUSINESS PROCESS

Structural similarity antar *process model* dihitung untuk digunakan dalam proses *clustering*. Perhitungan nilai *structural similarity* menggunakan matriks *incidency* dari setiap proses model sebagai proses matriks. Jarak antara dua proses model (*d*) diperoleh dengan menghitung perbedaan matriks. Perhitungan nilai *similarity* dilakukan dengan formula berikut [1].

$$Sim = \frac{1}{(1 + d)} \tag{1}$$

dengan

sim = nilai *similarity*

d = jarak antar proses model

Langkah-langkah untuk menghitung nilai jarak [1]:

- Membentuk proses matriks menggunakan matriks *incidency*.
- Melakukan normalisasi proses matriks untuk menyamakan jumlah baris dan kolom dalam proses matriks, misalnya dalam model 1 terdapat satu node yang tidak ada dalam model lain, baris dan kolom dalam proses matriks ditambahkan dengan elemen 0.
- Menghitung matriks jarak (*dist*) dengan melakukan pengurangan terhadap proses matriks yang sudah dinormalisasi. Matriks jarak selanjutnya diabsolutkan

$$dist = |NM_1 - NM_2| \tag{2}$$

dimana

dist = matriks jarak

NM₁ = matriks hasil normalisasi dari matriks M1

NM₂ = matriks hasil normalisasi dari matriks M2

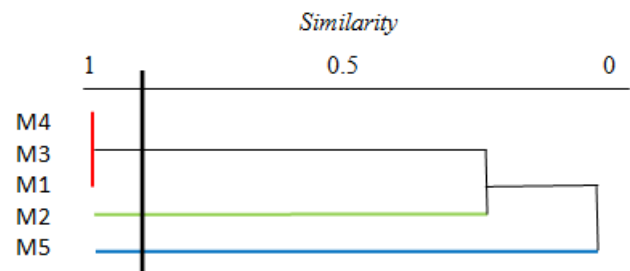
- Jarak (*d*) diperoleh dengan menambahkan setiap elemen dari matriks jarak (*dist*) yang sudah diabsolutkan.

Level *similarity* ditunjukkan dengan nilai $0 \leq sim \leq 1$ [9]. Jika *similarity* bernilai 0 maka tidak similar, dan jika bernilai 1 akan menunjukkan maksimal *similarity*.

2.3 COMPLETE LINKAGE CLUSTERING

Complete linkage merupakan salah satu metode dari algoritma *hierarchical*. Algoritma *hierarchical* dimulai dengan setiap data yang dianggap sebagai *entity* yang berbeda. Untuk langkah pertama, dua data yang paling similar akan dikombinasikan kedalam sebuah *cluster*. Langkah selanjutnya adalah apakah dua data yang lain akan dikombinasikan untuk membentuk *cluster* kedua atau data ketiga akan ditambahkan ke *cluster* yang sudah ada dengan menggunakan jarak maksimal dari minimal (*max min operation*).

Proses *clustering* dilakukan dengan nilai *structural similarity*. Hasil *clustering* dapat dilihat dengan dendrogram. Dendrogram akan menyajikan data pengkombinasian yang lengkap dari seluruh algoritma. Langkah pengkombinasian dapat dibaca dari bawah ke atas. Garis vertikal kiri yang bergabung merupakan langkah pertama pengkombinasian. Garis horizontal menunjukkan *similarity* dari model yang dikombinasikan. Dendrogram dapat dipotong pada titik tertentu untuk menentukan jumlah *cluster*. Gambar 2 merupakan gambar dendrogram hasil pemotongan pada titik 0.9 yang menghasilkan 3 *cluster*.



Gambar 2. Dendrogram

2.4 SILHOUETTE

Silhouette merupakan sebuah metode interpretasi dan validasi data *cluster* yang diperkenalkan oleh Rousseeuw pada tahun 1987. Dalam kasus *business process model* yang menggunakan nilai *similarity* sebagai jarak, *silhouette* merupakan metode validasi yang paling sesuai [9]. *Silhouette* akan mengevaluasi penempatan setiap objek dalam setiap *cluster* dengan membandingkan jarak rata-rata objek dalam satu *cluster* dan jarak antara objek dengan *cluster* yang berbeda. Untuk menghitung *silhouette* diperlukan hasil *clustering* dan *dissimilarity* antar titik. *Dissimilarity* dihitung dengan formula 1-sim. Kemudian nilai *silhouette* dapat dihitung dengan formula berikut [9]

$$s(i) = \frac{b[i] - a[i]}{\text{Max}(a[i], b[i])} \quad (3)$$

dimana
s(i) = *silhouette* pada titik ke-*i*
a(i) = rata-rata *dissimilarity* titik ke-*i* ke semua titik lain dalam satu *cluster*
b(i) = minimal *dissimilarity* titik ke-*i* ke titik lain dari *cluster* yang berbeda

Silhouette memiliki nilai $-1 \leq s \leq 1$. Jika dalam suatu *cluster* hanya memiliki 1 anggota, maka nilai *silhouette* adalah 0. Semakin mendekati nilai 1 semakin baik, maka suatu titik sudah berada pada *cluster* yang tepat. Ketika *silhouette* bernilai 0 maka suatu titik tidak jelas harus ditempatkan pada *cluster* yang mana. Ketika *s(i)* mendekati nilai -1 maka suatu titik berada pada *cluster* yang kurang tepat. *Average silhouette* dapat digunakan untuk mengevaluasi hasil *clustering* [9]. *Average silhouette* didapatkan dengan menghitung seluruh nilai *silhouette* dan tidak menghiraukan nilai 0.

3. METODOLOGI

Langkah-langkah yang dilakukan dalam penelitian ini adalah sebagai berikut:

1. Penentuan Prosedur
 Prosedur yang digunakan adalah prosedur yang berada pada dokumen mutu fakultas sub bagian keuangan dan kepegawaian Universitas Sebelas Maret. Dalam dokumen mutu, prosedur fakultas sub bagian keuangan dan kepegawaian terdiri dari 64 prosedur. Prosedur yang digunakan adalah prosedur nomor 1 sampai dengan nomor 64. Adapun untuk prosedur nomor 20 tidak digunakan karena tidak terdapat pada prosedur mutu, dan prosedur nomor 15 juga tidak digunakan karena adanya kesalahan dalam penulisan prosedur, sehingga jumlah seluruh prosedur yang digunakan adalah 62.
2. Pemodelan Prosedur dengan *Petri net*
 Prosedur dimodelkan dengan *petri net* menggunakan *software* Workflow Petri net Designer (WOPED) sehingga menghasilkan model yang sesuai. Pemodelan prosedur dilakukan dengan memodelkan notasi yang ada pada *flowchart* dengan notasi *petri net*. Pemodelan meliputi pemodelan notasi proses menjadi *transition*, data menjadi *place*, *decision* menjadi *OR-split*, dan proses paralel dimodelkan dengan *AND-split*.
3. Validasi Model *Petri net*
 Validasi dilakukan terhadap struktur model *petri net* yang meliputi *soundness*, *structural analysis*, *liveness*, dan *boundedness*. Validasi dilakukan dengan *software* Workflow Petri net Designer (WOPED). Hasil pemodelan *petri net* yang tidak valid tidak akan digunakan untuk proses selanjutnya.
4. Merepresentasikan Model *Petri net* ke Matriks *Incidency*
 Setelah pemodelan valid, *petri net* akan direpresentasikan dengan matriks *incidency*. Matriks *incidency* didapatkan dengan *software* Platform Independent Petri net Editor (PIPE).
5. Menghitung Matriks *Similarity* antar Matriks *Incidency*
 Perhitungan matriks *similarity* menggunakan matriks *incidency* sebagai proses matriks. Matriks *similarity* dihitung menggunakan persamaan 1 dan 2. Untuk melakukan perhitungan pada langkah 5, 6, dan 7

dikembangkan *software* menggunakan bahasa pemrograman Java.

6. *Clustering* Matriks *Similarity* dengan *Complete Linkage*
 Matriks *similarity* antar model akan dikelompokkan dengan *Complete linkage*. Hasil *clustering* dengan *complete linkage* dipotong pada dua titik untuk mendapatkan jumlah *cluster* yang bervariasi. Nilai yang digunakan untuk pemotongan dihitung menggunakan *student's t distribution*. Persamaan 4 merupakan rumus untuk menghitung *student's t distribution* [10].

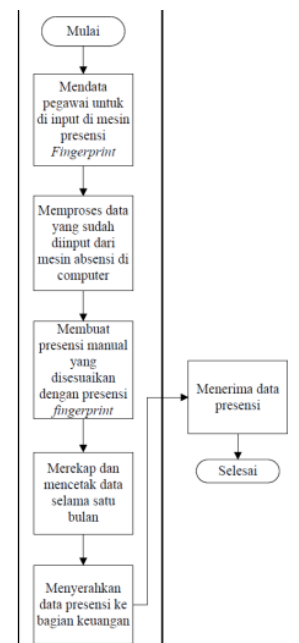
$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}} \quad (4)$$

dimana
t = nilai pada level *confidence* 97.5% yang bernilai 2.00 dan 99.5% yang bernilai 2.70.
r = estimasi titik pemotongan
n = jumlah data sampel

7. Validasi Hasil *Cluster* dengan *Silhouette*
 Hasil *cluster* akan divalidasi menggunakan *Silhouette*. *Silhouette* akan menentukan seberapa baik penempatan suatu objek dalam *cluster* dengan menghitung rata-rata *dissimilarity* setiap *cluster*. *Dissimilarity* dihitung menggunakan rumus 1-sim. Kemudian nilai *silhouette* akan dihitung menggunakan persamaan 3. *Average silhouette* dari setiap titik pemotongan juga akan dihitung untuk menentukan jumlah *cluster* yang paling optimal.

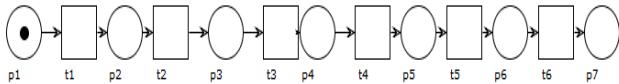
4. HASIL DAN PEMBAHASAN

Penelitian ini menggunakan 62 prosedur fakultas sub bagian keuangan dan kepegawaian Universitas Sebelas Maret. Gambar 3 merupakan contoh prosedur mutu pengelolaan absensi pegawai fakultas berbasis fingerprint dan absensi manual.



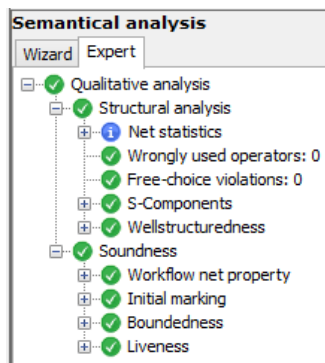
Gambar 3 prosedur mutu pengelolaan absensi pegawai fakultas berbasis fingerprint dan absensi manual [11]

Prosedur dimodelkan dengan *petri net* menggunakan *software* Workflow Petri net Designer (WOPED) sehingga menghasilkan model yang sesuai. Pemodelan prosedur dilakukan dengan memodelkan notasi yang ada pada *flowchart* dengan notasi *petri net*. Pemodelan meliputi pemodelan notasi proses menjadi *transition*, data menjadi *place*, *decision* menjadi OR-split, dan proses paralel dimodelkan dengan AND-split. Prosedur pada Gambar 3 selanjutnya dimodelkan dengan *petri net* sehingga menghasilkan 6 *transition* dan 7 *place*. Pemodelan prosedur pada Gambar 3 dapat digambarkan dengan model *petri net* seperti pada Gambar 4.



Gambar 4 Pemodelan prosedur dengan *petri net*

Setelah pemodelan selesai dibuat, proses validasi dilakukan dengan *software* WOPED. Gambar 5 merupakan hasil validasi pemodelan prosedur dengan *petri net* pada Gambar 4.



Gambar 5 Hasil validasi pemodelan

Analisis yang dilakukan terdiri dari *structural analysis*, *soundness*, *boundedness*, dan *liveness*. Icon hijau menandakan bahwa tidak terdapat error pada saat model divalidasi. Apabila terdapat error pada saat validasi maka icon berwarna merah akan muncul. Dari hasil validasi yang dilakukan terdapat 50 model yang valid dan 12 model yang tidak valid. Pemodelan yang tidak valid tidak akan digunakan untuk proses selanjutnya.

Dari 50 pemodelan yang valid, selanjutnya dibuat matriks *incidency* menggunakan *software* Platform Independent Petri net Editor (PIPE). Matriks *incidency* terdiri dari matriks baris yang menyatakan *place* dan kolom menyatakan *transition*. Jumlah baris dan kolom didalam matriks sama dengan jumlah *place* dan *transition*. M16 merupakan matriks *incidency* dari pemodelan pada Gambar 4.

$$M16 = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Setelah seluruh pemodelan sudah direpresentasikan dengan matriks *incidency*, maka matriks *similarity* antar matriks *incidency* akan dihitung. Nilai *structural similarity* kemudian digunakan untuk *clustering* dengan *complete linkage*. Hasil *clustering* secara keseluruhan dapat ditampilkan dengan dendrogram. Dendrogram akan dipotong pada dua titik untuk memperoleh jumlah *cluster*. Nilai yang digunakan untuk melakukan pemotongan dihitung menggunakan *student's t distribution* pada level *confidence* 97.5% dan 99.5%. Hasil perhitungan pada level *confidence* 97.5% menghasilkan nilai 0.3 sedangkan pada level *confidence* 99.5% menghasilkan nilai 0.4. Hasil *clustering* dengan pemotongan pada titik 0.3 menghasilkan 15 *cluster*, dan pemotongan pada titik 0.4 menghasilkan 18 *cluster*. Hasil *clustering* selanjutnya divalidasi dengan *silhouette*. Tabel 1 merupakan hasil *clustering* dan *silhouette* pada setiap titik pemotongan.

Tabel 1 Hasil *clustering* dan *silhouette*

Model	Pemotongan 0.3		Pemotongan 0.4	
	Cluster	Silhouette	Cluster	Silhouette
M10	1	1	1	1
M21	1	1	1	1
M6	2	0	11	0
M5	2	0.79	2	1
M13	2	0.79	2	1
M34	2	0.79	2	1
M43	2	0.79	2	1
M28	3	1	4	1
M38	3	1	4	1
M14	4	0.57	3	1
M16	4	0.57	3	1
M17	4	0.57	3	1
M27	4	0.57	3	1
M32	4	0.57	3	1
M39	4	0.57	3	1
M41	4	0.57	3	1
M46	4	0.57	3	1
M50	4	0.57	3	1
M19	4	0.36	5	1
M20	4	0.36	5	1
M29	4	0.36	5	1
M30	4	0.36	5	1
M36	4	0.36	5	1
M49	4	0.36	5	1
M11	5	0.21	6	1
M18	5	0.21	6	1
M35	5	0.21	6	1
M40	5	0.21	6	1
M8	5	0.71	7	1
M9	5	0.71	7	1
M12	5	0.71	7	1

Tabel 1 Hasil *clustering* dan *silhouette*

Model	Pemotongan 0.3		Pemotongan 0.4	
	Cluster	Silhouette	Cluster	Silhouette
M22	5	0.71	7	1
M23	5	0.71	7	1
M24	5	0.71	7	1
M25	5	0.71	7	1
M26	5	0.71	7	1
M31	5	0.71	7	1
M33	5	0.71	7	1
M48	5	0.71	7	1
M3	6	1	8	1
M4	6	1	8	1
M1	7	0	9	0
M2	8	0	10	0
M7	9	0	12	0
M15	10	0	13	0
M37	11	0	14	0
M42	12	0	15	0
M44	13	0	16	0
M45	14	0	17	0
M47	15	0	18	0
Average Silhouette	0.78		1	

Tabel 1 merupakan hasil perhitungan *silhouette* dari setiap titik pemotongan. Dari hasil perhitungan *silhouette*, juga didapatkan nilai *average silhouette* dari setiap titik pemotongan. Terdapat 2 titik pemotongan dengan *average silhouette* yang berbeda. Pemotongan pada titik 0.3 memiliki *average silhouette* 0.78, sedangkan pemotongan pada titik 0.4 memiliki *average silhouette* 1. Jumlah *cluster* yang paling optimal diperoleh pada titik pemotongan yang paling tinggi dan *average silhouette* yang mendekati 1. Dari setiap titik pemotongan sudah diperoleh nilai *average silhouette* yang mendekati 1, sehingga bisa dikatakan menghasilkan *cluster* yang optimal. Namun, pada penelitian ini, jumlah *cluster* yang paling optimal terdapat pada titik 0.4 dengan *average silhouette* tepat 1 sehingga menghasilkan *silhouette* yang paling maksimal.

5. KESIMPULAN DAN SARAN

Hasil penelitian menunjukkan dari 62 prosedur yang dimodelkan dengan *petri net*, terdapat 50 model yang valid dan 12 model yang tidak valid. Hasil *clustering* dengan pemotongan pada titik 0.3 menghasilkan 15 *cluster*, dan pada titik 0.4 menghasilkan 18 *cluster*. Jumlah *cluster* yang paling optimal dihasilkan oleh pemotongan pada titik 0.4 dengan jumlah *cluster* 18.

Untuk penelitian selanjutnya dapat melakukan perhitungan menggunakan *behavioral similarity* sehingga dapat menentukan karakteristik proses bisnis, dan prosedur yang tidak valid dapat disampaikan kepada pihak terkait.

6. DAFTAR PUSTAKA

[1] Sarno, R., Ginardi, H., Pamungkas, E. W., & Sunaryono, D. (2013). *Clustering of ERP Business Process Fragments. The International Conference on Computer,*

Control, Informatics and its Applications 2013 (IC3INA 2013).

- [2] Zhao, X., Wei, C., Lin, M., Feng, X., & Lan, W. (2010). Parallel Application Scheduling Model Based on *Petri net* with Changeable Structure. Dalam T. Aized, *Advances in Petri net Advances in Petri net* (hal. 153-174). Rijeka: Sciyo.
- [3] Drennan, R. D. (2009). *Statistics for Archaeologists: A Commonsense Approach* (2nd Edition ed.). US: Springer.
- [4] Jung, J.-Y., Bae, J., & Liu, L. (2009). Hierarchical Clustering of Business Process models. *International Journal of Innovative Computing, Information and Control*, 5 (12).
- [5] Qiao, M., Akkiraju, R., & Rembert, A. J. (2011). Towards Efficient Business Process Clustering and Retrieval: Combining Language Modeling and Structure Matching. Dalam S. Rinderle-Ma, F. Toumani, & a. K. Wolf, *BPM 2011* (hal. 199-214). Berlin Heidelberg : Springer-Verlag.
- [6] Bae, J., Liu, L., Caverle, J., & Rouse, W. B. (2010). Process Mining, Discovery, and Integration using Distance Measures. *International Journal of Web Services Research*.
- [7] Wang, J., Wong, R. K., Jianwei, D., Guo, Q., & Wen, L. (2012). Efficient Selection of Process Mining Algorithms. *IEEE Transactions on Services Computing*, 6 (4), 484-496
- [8] Aalst, W. v., & Hee, K. v. (2002). *Workflow Management: Models, Methods, and Systems*. London: The MIT Press.
- [9] Kaufman, L., & Rousseeuw, P. J. (2005). *Finding Groups in Data : An Introduction to Cluster Analysis*. Hoboken, New Jersey: John Wiley & Sons, Inc.
- [10] Spiegel, M. R., Schiller, J. J., & Srinivasan, R. A. (2009). *Schaum's Outline of Theory and Problems of Probability and Statistics* (3rd Edition ed.). New York: McGraw-Hill
- [11] Tim ISO. (2013). *Dokumen Mutu Fakultas Sub Keuangan dan Kepegawaian UNS*. Surakarta: Universitas Sebelas Maret.