

Identifikasi Bottleneck pada Hasil Ekstraksi Proses Bisnis ERP dengan Membandingkan Algoritma Alpha++ dan Heuristics Miner

Laeila Mardhatillah, Mahendrawathi Er, Renny P Kusumawardani

Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111

E-mail: mahendr_w@its-sby.edu

Abstrak— Saat ini banyak perusahaan menggunakan sistem informasi untuk menunjang proses bisnis. Namun kenyataannya hanya beberapa perusahaan yang melakukan evaluasi untuk proses bisnis pada sistem informasi tersebut. Evaluasi ini didapat dari data *Event log* yang merupakan hasil proses ekstraksi ERP. Cara melakukan evaluasi yaitu dengan *process mining*. *Process mining* berfungsi untuk menggali proses transaksi sehingga terbentuk suatu *workflow* proses bisnis yang actual. *Workflow* proses bisnis ini akan digambarkan dalam bentuk Petri Net, selanjutnya dari Petri Net inilah akan dilakukan analisis untuk mengidentifikasi adanya *Bottleneck*. *Bottleneck* merupakan peristiwa pada suatu transaksi yang memiliki waktu tunggu yang lebih lama dibandingkan transaksi lainnya dalam suatu proses bisnis.

Dengan adanya PROM Tools maka penggambaran mengenai proses bisnis ERP dapat terbentuk. Penggambaran model tersebut menggunakan dua algoritma, yaitu Algoritma Alpha ++ dan Heuristics Miner. Kedua algoritma ini digunakan untuk mencari perbedaan bottleneck yang terjadi. Dari penelitian didapatkan hasil bahwa algoritma sangat mempengaruhi letak *bottleneck*. Letak tersebut didasarkan pada perhitungan waktu token yang ada pada *place* (tempat antara dua transisi atau dua transaksi) saat terbentuk model. Heuristic Miner dapat memodelkan proses bisnis secara tepat karena memperhitungkan frekuensi. Sedangkan Algoritma Alpha ++ tidak dapat memperhitungkan frekuensi, sehingga cocok untuk proses bisnis yang tidak terlalu banyak pilihan prosedur. Selain itu semakin panjang proses bisnis akan semakin meningkatkan potensi terjadinya *bottleneck* karena perhitungan *bottleneck* membandingkan rata-rata waktu tunggu tiap *place* yang terbentuk.

Kata Kunci— Event log, Process mining, Petri Net, Bottleneck, Algoritma Alpha++, Heuristic Miner.

I. PENDAHULUAN

Saat ini banyak perusahaan menggunakan sistem informasi perusahaan atau *Enterprise Information Systems* untuk menunjang integrasi proses bisnisnya seperti ERP, CRM, BPM, WFM, dan PDM sistem. ERP merupakan sistem informasi yang terdiri dari berbagai macam modul. Modul ini terdiri dari berpuluh-puluh bahkan beratus-ratus proses bisnis yang saling terkait satu sama lain [1].

Untuk melakukan evaluasi terhadap proses bisnis ERP diperlukan pembentukan model bisnis. Namun model dari proses bisnis ERP susah diterjemahkan. Maka diperlukan

suatu metode untuk mempermudah pemodelan proses bisnis, yaitu dengan cara *process mining*.

Process mining berfungsi untuk menggali proses bisnis yang telah dieksekusi pada *event log* sehingga terbentuk suatu *workflow* proses bisnis yang actual. *Event log* sendiri merupakan pencatatan transaksi atau *audit trail* yang terjadi pada suatu sistem informasi, seperti ERP [2].

Namun untuk mendapatkan data berupa *event log* tidaklah semudah yang dibayangkan. Format dari *event log* memiliki berbagai ragam jenis. Akan tetapi format tertentu yang dapat dilakukan *process mining*. Selain itu tidak semua aplikasi sistem informasi memiliki *event log*. Selain dari sistem ERP, mendapatkan *event log* juga dapat dilakukan dengan cara melakukan query data base.

Hasil akhir *process mining* dari *event log* ERP tergambar dalam bentuk Petri Net. Agar pencatatan pada ERP dapat ditransformasi dalam bentuk Petri Net dibutuhkan perubahan model data. Model data *event log* yang ada pada ERP diconvert dalam bentuk *Mining XML*. Hal ini dilakukan untuk mengelompokkan data berdasarkan masing-masing proses bisnis yang berlangsung. Pengelompokan ini bertujuan untuk mempermudah penggambaran *workflow* [3].

Berbagai metode untuk *process mining* telah dikembangkan. Salah satu tools yang digunakan untuk melakukan *process mining* adalah PROM. Banyak algoritma yang dikembangkan pada tools tersebut. Beberapa diantaranya adalah algoritma Alpha++ dan Heuristics Miner.

Salah satu permasalahan yang mungkin muncul dalam proses bisnis yang panjang adalah terjadinya *bottleneck*. Saat satu proses membutuhkan waktu yang cukup lama agar dapat melanjutkan proses lainnya maka saat itulah dikatakan sebagai *bottleneck* [4].

Secara teori Petri Net dapat menggambarkan di mana akan terjadi *bottleneck*, sehingga dapat terlihat proses-proses dalam bisnis yang terdapat *bottleneck*. Untuk menggambarkan bagaimana *bottleneck* [5] terjadi dapat dilakukan dengan *PROM Tools*. Dari model yang telah terbentuk akan terdeteksi *case* yang mengalami *bottleneck* dari salah satu fitur yang disediakan dalam *PROM Tools*.

Sehingga target yang ingin dicapai adalah identifikasi *bottleneck* pada setiap model yang terbentuk dari penggambaran masing-masing Algoritma. Selanjutnya akan dianalisis mengapa *bottleneck* dapat terjadi pada suatu proses bisnis ERP.

II. KAJIAN PUSTAKA

A. Proses Bisnis

Proses bisnis adalah serangkaian atau sekumpulan aktifitas yang dirancang untuk menyelesaikan tujuan strategik sebuah organisasi [6]. Proses bisnis suatu perusahaan perlu dibuat suatu pemodelan untuk menggambarkan alur proses bisnis yang ada. Penggambaran proses bisnis dapat dilakukan dengan menggunakan diagram flow.

Penggambaran proses bisnis dapat dilakukan dengan cara flow map yaitu menggambarkan sendiri proses bisnis yang terjadi. Penggambaran ini dilakukan dengan cara mempelajari terlebih dahulu proses bisnis yang lalu menggambarkan alur proses melalui flow diagram.

Pada saat ini banyak perusahaan yang membutuhkan evaluasi pada proses bisnisnya sehingga dibutuhkan penggambaran proses bisnis yang actual. Untuk menggambarkan proses bisnis actual hanya dibutuhkan data *history* berupa *event log* [7].

B. Petri Net

Petri Net terdiri dari place, transisi dan panah yang menunjukkan hubungan antara place dan transisi. Place dapat berfungsi sebagai input atau output suatu transisi. Place sebagai input menyatakan keadaan yang harus dipenuhi agar transisi dapat terjadi. Setelah transisi terjadi maka keadaan akan berubah. Place yang menyatakan keadaan tersebut adalah output dari transisi.

Petri Net merupakan graf yang bipartite, sehingga tidak ada panah yang menghubungkan dua place atau dua transisi. Dalam notasi grafis, place dilambangkan dengan lingkaran, transisi dilambangkan dengan bujursangkar dan konektor dengan panah.

Petri Net dapat didefinisikan sebagai berikut[8] :

Petri Net merupakan pasangan 4-tuple (P,T,A,W) dengan :

- P adalah himpunan yang terdiri dari *palce* , $P = \{p_1, p_2, \dots, p_n\}$,
- T adalah himpunan yang terdiri dari transisi , $T = \{t_1, t_2, \dots, t_n\}$,
- A adalah himpunan arc, $A \subseteq (P \times T) \cup (T \times P)$
- W adalah fungsi bobot, $W: A \rightarrow \{1, 2, 3, \dots\}$.

C. Event log

Event log merupakan proses pencatatan histori berupa transaksi atau audit trail pada suatu tool sistem informasi. Setiap sistem informasi pasti memiliki *event log* sebagai bukti dari transaksi yang sedang berlangsung. Contohnya saja pencatatan *event log* yang ada pada ERP.

Event log berisi informasi tentang kegiatan berupa case atau taks tertentu. Case sendiri disebut dengan “*process instance*” merupakan suatu kejadian yang sedang berlangsung. Misalnya Order ke supplier (purchasing), order oleh pelanggan (Customer order) dan beberapa kajadian lainnya. Sedangkan taks adalah aktivitas didalam trace, bias berupa tahapan aktifitas. Jadi dalam *trace* bisa terdapat banyak *task* [9] .

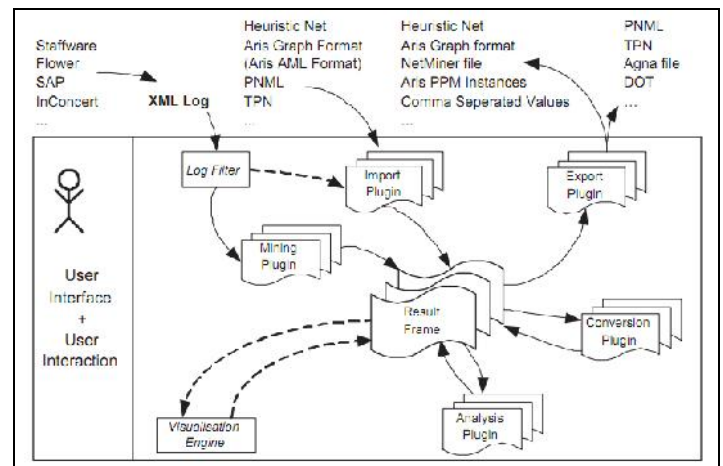
D. Algoritma Alpha++ dan Heuristics Miner

Untuk algoritma yang digunakan adalah algoritma Alpha++ dan Heuristics. Algoritma Alpha++ merupakan algoritma yang dapat mendeteksi adanya implicit dependensi [10]. Selain itu juga Alpha++ dipilih karena kelebihan yang dimiliki dibandingkan dengan algoritma Alpha yang lain. Dijelaskan pada jurnal Van der Alast bahwa algoritma Alpha di perluas menjadi Alpha+ dan di perluas lagi menjadi Alpha++.

Algoritma Heuristics memiliki kelebihan yang tidak dimiliki oleh Alpha++ seperti yang disebutkan dalam [11] bahwa algoritma ini dapat memperhitungkan frekuensi. Dan untuk ribuan log dapat menentukan proses yang dominan dan yang merupakan kebiasaan yang tidak umum dalam suatu proses. Dengan menggunakan perbedaan dua algoritma ini lah maka pendeteksian adanya bottleneck dilakukan.

E. PROM

Merupakan suatu tools yang digunakan untuk melakukan *process mining*. Adapun alur cara melakukan process mining tampak pada gambar 1



Gambar 1 Proses-proses yang dapat digambarkan pada PROM [12]

Pada bagian *result frame* adalah bentuk dari *workflow* proses bisnis yang *actual*. Selanjutnya *workflow* ini perlu dilakukan analisis lebih lanjut. Untuk menganalisisnya ada banyak cara yang dilakukan. Diantaranya adalah dengan analisis menggunakan *plugin* yang tersedia pada PROM, dibandingkan dengan proses bisnis yang ada pada perusahaan, dan dengan mengekspornya lalu menggunakan bantuan tools lain untuk menganalisa

F. ERP

ERP merupakan suatu sistem informasi perusahaan yang dirancang untuk mengkoordinasikan semua sumber daya, informasi dan aktifitas yang diperlukan untuk proses bisnis secara lengkap dan terintegrasi. ERP dirancang dengan sistem modular. Rancangan ini memungkinkan suatu perusahaan untuk memilih modul-modul yang ada pada ERP yang selanjutnya akan di sesuaikan dengan proses bisnis yang ada. Tiap modul umum terdiri dari berbagai macam transaksi [13] .

Pada dasarnya integrasi yang ada proses bisnis ERP menghubungkan transaksi yang ada pada modul-modul.

Sehingga dapat mendukung rencana bisnis yang ada pada suatu perusahaan

III. METODE PENELITIAN

A. Pengolahan Data

Setelah mendapatkan data maka hal selanjutnya yang dilakukan adalah menganalisa proses bisnis yang ada pada PT.XYZ.

Setelah melakukan pembelajaran tentang proses bisnis yang ada pada PT.XYZ. maka dilakukan mapping proses bisnis pada database. Proses mapping ini berguna dalam pemerolehan ekstraksi log

Data Preprocessing

Tahap ini merupakan tahap yang dilakukan sebelum data benar-benar dapat digunakan. Untuk tahap preprocessing dibagi menjadi dua yaitu tahap ekstraksi database dan tahap pembersihan data.

Ekstraksi database

ERP merupakan rangkaian proses bisnis yang panjang. Modul yang digunakan adalah modul Material Management untuk mengetahui alur data basenya dibutuhkan analisa terlebih dahulu dengan cara melihat hubungan foreign key. Tahap selanjutnya dilakukan query untuk pengambilan nilai dari kolom tabel. Untuk lebih jelasnya akan dijabarkan pada bab IV.

Pembersihan data

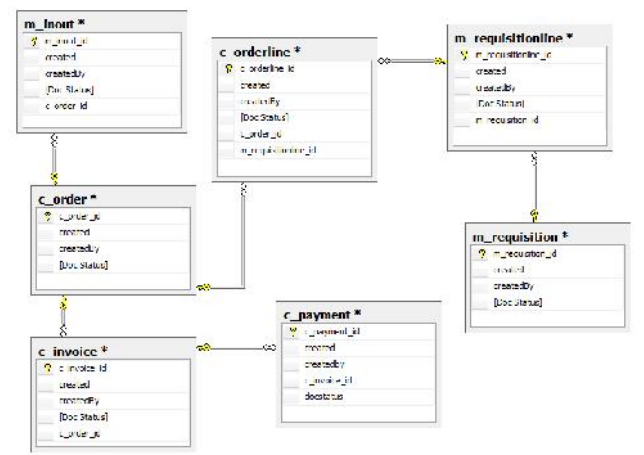
Pada tahap ini dilakukan pembersihan data atau pemilahan data. Dengan menganalisa data dan proses bisnis maka didapatkan bahwa aktivitas yang ada pada suatu proses bisnis adalah nama variable pada table. Oleh karena itu prose pemisahan tiap tahap proses untuk suatu request dilakukan secara manual dengan bantuan excel.

IV. IMPLEMENTASI

A. Ekstraksi database

Agar data dapat benar-benar digunakan untuk proses mining maka hal yang dilakukan adalah melakukan ekstraksi data base. Agar ekstrasi dapat dilakukan maka yang erlu dilakukan adalah mempelajari alur dari database. Adapun gambar alur database tersebut adalah sebagai berikut.

Pada gambar 2 dapat dilihat bahwa untuk mengetahui hubungan antara order dengan requisition dapat dilihat pada tabel order line. Berikut merupakan hubungan antar relasi yang terbentuk:



Gambar 2 Hubungan antar tabel untuk modul MM

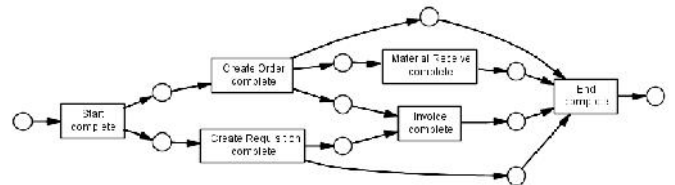
1. Pada tabel c_orderline, c_order_id memiliki banyak c_order_id dari table c_order.
2. Pada tabel c_orderline, m_requisitionline_id memiliki banyak m_requisitionline_id dari table m_requisitionline.
3. Pada tabel m_requisitionline, m_requisition_id memiliki banyak m_requisition_id dari table m_requisition.
4. Pada tabel c_invoice, c_order_id memiliki banyak c_order_id dari table c_order.
5. Pada tabel m_inout, c_order_id memiliki banyak c_order_id dari table c_order.
6. Pada tabel c_payment, c_invoice_id memiliki banyak c_invoice_id dari table c_invoice.

Dari hubungan antar table diatas maka dapat dilakukan query pada table yang ada diadempiere. Adapun query yang dilakukan terdapat pada halaman lampiran.

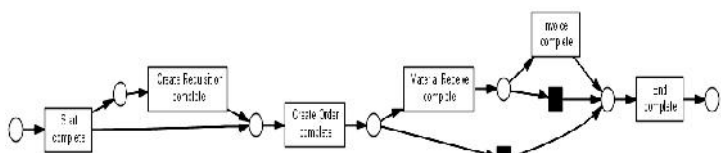
B. Model dengan Algorithma Alpha ++ dan Heuristic Miner

Setelah mendapatkan data yang terdapat pada Postgres melalui cara ekstrasi database dan proses pembersihan data, maka selanjutnya adalah membuat pemodelan berdasarkan data yang telah didapatkan.

Pemodelan dilakukan dengan menggunakan aplikasi PROM dan menggunakan algorithma alpha++ dan Heuristic Miner. Adapun model yang terbentuk ditunjukkan pada gambar 3 dan 4.



Gambar 3 Model yang terbentuk dengan Algorithma Alpha++

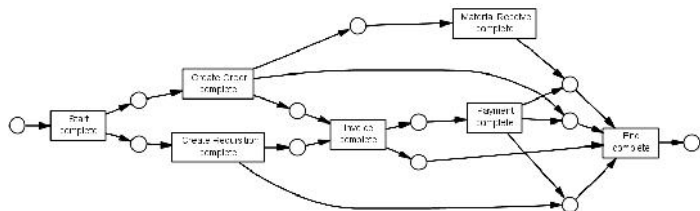


Gambar 4 Model yang terbentuk dengan Algoritma Heuristics Miner

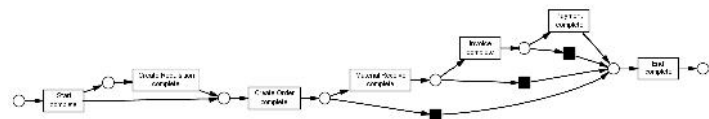
Gambar pada 3 dan 4 merupakan gambar untuk proses bisnis yang terjadi dari requisition hingga invoice. Alur yang terjadi dari gambar diatas adalah sebagai berikut

1. Requisition >> Order >> Material Receive >> Invoice
2. Order >> Material Receive >> Invoice

Untuk membuat model seperti pada gambar 3 dan 4 digunakan semua data yang ada pada Requisition hingga Invoice tanpa membuang data yang belum mencapai finish. Selain itu terdapat pemodelan lagi yang hingga sampai ada tahap payment. Adapun untuk model tersebut adalah sebagai berikut



Gambar 5 Model dengan Algoritma Alpha++ hingga Payment



Gambar 6 Model dengan Algoritma Heuristics Miner hingga Payment

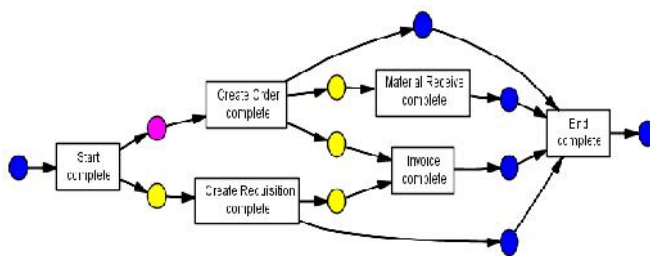
- Gambar 5 dan 6 merupakan model untuk kasus :
1. Requisition >> Order >> Material Receive >> Invoice >> Payment
 2. Order >> Material Receive >> Invoice >> Payment
- Pada kasus payment ini hanya terdapat 53 kasus.

V. ANALISIS

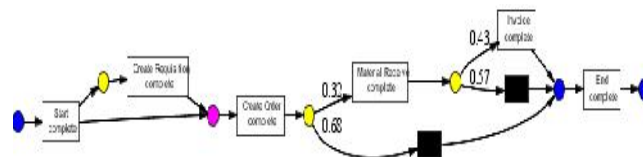
Selanjutnya adalah tahap identifikasi bottleneck. identifikasi ini berdasarkan dua kasus.

- Untuk kasus pertama alur yang terbentuk adalah
1. Requisition >> Order >> Material Receive >> Invoice
 2. Order >> Material Receive >> Invoice.
- Sedangkan untuk alur ke dua yang terbentuk adalah
1. Requisition >> Order >> Material Receive >> Invoice
 2. Order >> Material Receive >> Invoice

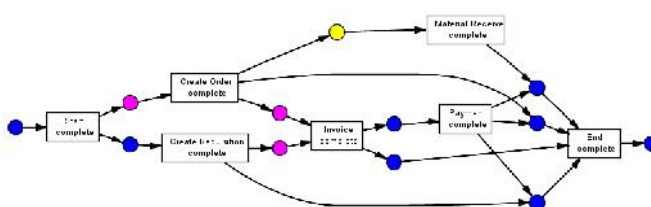
Berikut merupakan gambar dari model proses bisnis berdasarkan algoritma Alpha++ dan Heuristics Miner beserta place yang terjadi *bottleneck*.



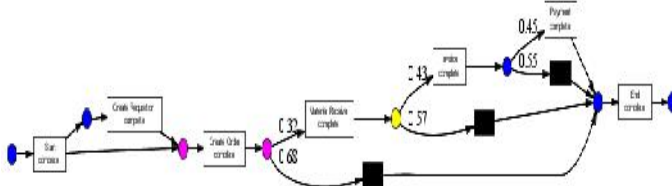
Gambar 7 Bottleneck pada kasus pertama dengan Algoritma Alpha++



Gambar 8 Bottleneck pada kasus pertama dengan Algoritma Heuristics Miner



Gambar 9 Bottleneck pada kasus kedua dengan Algoritma Alpha++



Gambar 10 Bottleneck pada kasus kedua dengan Algoritma Heuristics Miner

Keterangan untuk gambar 7 sampai 10 adalah sebagai berikut, warna biru menandakan waktu tunggu yang cepat, warna kuning menandakan waktu tunggu yang cukup cepat, dan warna magenta menunjukkan waktu tunggu yang tinggi.

Dari kedua kasus diatas akan terlihat bahwa bottleneck yang terjadi berbeda untuk setiap algoritma. Namun untuk kasus pertama pergintungan bottleneck terlihat sama karena pada place baik alpha++ dan Heuristic Miner yang mengalami bottleneck terdapat 815 token.

Token ini merupakan token yang berasal dari proses yang akan menuju transaksi order. Baik yang melalui requisition atau tanpa requisition. Bila proses tanpa melalui requisition maka waktu tunggu akan dihitung 0.

Perhitungan bottleneck hanya mengitung selisih waktu yang dibutuhkan antar transaksi yang dihubungkan oleh place. Lalu dibandingkan rata-rata waktu tunggu antara place. Pada kasus pertama ini jumlah token untuk algoritma Alpha++ dan Heuristics sama pada place yang mengalami bottleneck. Waktu rata-rata pun sama yaitu 29,19.

Performance information of the selected place:

Frequency: 851 visits
 Arrival rate: 0.29 visits per day

	Waiting time (days)	Synchronization time (day...	Sojourn time (days)
avg	29.2	0.0	29.2
min	0.0	0.0	0.0
max	2908.28	0.0	2908.28
stddev	236.5	0.0	236.5

Gambar 11 Total token yang ada pada kasus pertama

Waktu tunggu yang lama ini terdapat pada proses permintaan order yang melalui requisition. Yang menyebabkan lamanya proses adalah prosedur yang dijalankan saat request terlalu banyak menunggu daftar request. Selain itu pula waktu antara document request dibuat hingga komplit membutuhkan waktu yang cukup lama. Karena proses inilah maka bottleneck terjadi diantara proses requisition dengan order.

Pada kasus kedua ditambah satu transaksi dan hal ini dapat memberikan gambaran bentuk bottleneck yang berbeda pula. Sebagaimana digambarkan dalam 9 dan 10.

Pada kasus 9 yang diperhitungkan adalah jarak waktu antara invoice ke order, dan order tersebut harus melalui requisition karena proses sebelumnya berjalan bersamaan sehingga perhitungannya adalah dari order ke invoice dan requisition ke invoice. Dalam hal ini tentu saja perhitungan akan salah. Karena proses yang sebenarnya adalah sebelum masuk dalam transaksi invoice harus melalui material receive terlebih dahulu.

Sedangkan untuk gambar 10 terlihat bottleneck juga berada pada place diantara order dan material receive. Namun ada 6 item yang perhitungannya diambil dari requisition ke order. Namun tidak terlalu membuat perubahan rata-rata karena nilai rata-rata dari 6 item tersebut hanya 0,23.

Dengan melihat dua kasus tersebut maka akan terlihat bahwa penentuan bottleneck lebih tepat pada algorithm heuristic. Serta semakin panjang proses bisnis dapat menentukan perubahan perhitungan bottleneck. Hal ini disebabkan place yang dibandingkan waktu tungguanya semakin banyak pula. Sehingga kemungkinan untuk terjadi bottleneck semakin bertambah.

VI. KESIMPULAN

Berdasarkan pada hasil analisa untuk melakukan identifikasi bottleneck maka didapatkanlah hasil :

1. Pada tugas akhir telah dilakukan ekstraksi *event log* dengan cara melakukan query antar tabel. Tabel yang dipilih adalah tabel yang berhubungan dengan modul MM. Adapun tabel-tabel tersebut adalah sebagai berikut:
 - M_Requisition (No Dokumen)
 - M_Requisitionline (Daftar request berdasarkan No Dokumen)
 - C_Order (No Dokumen)
 - C_Orderline (Daftar order berdasarkan No Dokumen)
 - M_inout (Dokumen material receive)
 - C_Invoice (Dokumen invoice)
 - C_Payment (Dokumen payment)

Untuk semua table tersebut *variable* yang harus ada adalah Id *primary key*, tanggal dokumen dibuat,

pembuat dokumen , status dokumen , dan *foreign key* yang menggambarkan hubungan antar tabel.

2. Hasil uji coba pada data proses Material Management maka diketahui bahwa algoritma ikut berperan dalam ketepatan *bottleneck*. Semakin tepat suatu model menggambarkan proses bisnis yang terjadi maka semakin tepat pula perhitungan *bottleneck* untuk tiap *place*. Pada Algoritma Alpha++ tidak memperhitungkan frekuensi dalam memodelkan proses bisnis. Terlihat dari model yang membentuk AND karena ada sebagian kecil dari proses , yaitu requisition dengan order dan material receive dengan invoice yang berjalan hampir bersamaan. Sehingga algoritma ini cocok untuk proses bisnis yang tidak terlalu banyak pilihan prosedur. Sedangkan untuk Heuristics Miner memperhitungkan frekuensi sehingga dapat mendeteksi adanya noise pada data. Algoritma Heuristics menggambarkan transaksi yang paling dominan karena itu model yang terbentuk berupa garis lurus.
3. Dari hasil perhitungan dan keluaran PROM Tools untuk menentukan *bottleneck* dihitung berdasarkan *token* yang ada pada *place*. Dalam analisis diketahui bahwa algoritma ikut menentukan *bottleneck* yang terjadi. Hal ini berhubungan dengan tepat tidaknya algoritma dalam memodelkan . Dari model tersebut *place* yang terbentuk akan dihitung waktu tunggu tiap *token* yang ada. Selanjutnya dihitung waktu tunggu rata-rata tiap *place*. Model yang digambarkan dengan Algoritma Alpha++ untuk kasus requisition hingga payment, *bottleneck* terjadi pada place antara order ke invoice dan requisition ke invoice. Perhitungan waktu pada place tersebut menghitung waktu antara order ke invoice dan requisition ke invoice. Pada kenyataannya sebelum invoice terdapat material receive. Namun karena Algoritma Alpha++ menggambarkan model bentuk AND maka hubungan tersebut tidak tergambar. Letak *bottleneck* dalam hal ini tidak sesuai dengan prosedur proses bisnis yang berlaku pada PT.XYZ. Sedangkan model yang digambarkan dengan Heuristics letak *bottleneck* berada diantara order dan material receive.
4. Selain itu semakin banyak proses bisnis akan semakin meningkatkan potensi terjadinya *bottleneck* pada *place*. Hal ini disebabkan waktu tunggu rata-rata yang ada pada *place* semakin banyak yang dibandingkan.

DAFTAR PUSTAKA

[1] Hamilton, S. Maximizing Your ERP System : a Practical Guide for Manager. New York: McGraw-Hill (2003).
 [2] Weber, P. Research Progress Report 3 - Thesis Proposal Working Title of Thesis : Mechine Learning in Process Mining. School of Computer Science University of Brimingham (2010), p.3.
 [3] Eder, J., et al., A Generic Import Framework for Process Event-logs, in Business Process Management Workshops. Springer Berlin / Heidelberg (2006), p. 81-92.
 [4] Langley, G. J., Moen, R., Nolan, K. M., & Nolan, T. W. The Improvement Guide:A Practical Approach to Enhancing Organizational Performance. San Francisco: Jossey-Bass (2009).

- [5] Chowdhury, J. Application of Information Technology and Business Process Management (BPM) to enhance organizational process. Ittoolbox (2006).
- [6] Ritchi, H. Identifikasi Pengendalian Aplikasi Dalam Analisis Proses Bisnis. Pustaka UNPAD (2009).
- [7] Weber, P. Research Progress Report 3 - Thesis Proposal Working Title of Thesis : Mechine Learning in Process Mining. School of Computer Science University of Brimingham (2010). p.3.
- [8] Weber, P. Research Progress Report 3 - Thesis Proposal Working Title of Thesis : Mechine Learning in Process Mining. School of Computer Science University of Brimingham (2010) p.3.
- [9] Weber, P. A Framework for The Comparison of Process Mining Algorithms. School of Computer Science University of Brimingham (2009) p.1.
- [10] Weber, P., A Framework for The Comparison of Process Mining Algorithms. School of Computer Science University of Brimingham, (2009) p.1.
- [11] Weijters, A., van der Aalst, W., & Alves de Medeiros, A. Process Mining with the HeuristicsMiner. Eindhoven University of Technology (2007) 24.
- [12] Chowdhury, J. Application of Information Technology and Business Process Management (BPM) to enhance organizational process. Ittoolbox (2006).
- [13] Hamilton, S. Maximizing Your ERP System : a Practical Guide for Manager. New York: McGraw-Hill (2003).