

TUGAS AKHIR - KS09 1336

**ANALISIS PENGARUH VOLUME DAN VARIASI ARTIKEL
TERHADAP *LEAD TIME* PENYELESAIAN PENGEPAKAN
DI *PRODUCTION DISTRIBUTION CENTER* PT. XYZ
DENGAN MENGGUNAKAN ALGORITMA *DUPLICATE
GENETIC***

Suviani Ningrum
NRP 5210100149

Dosen Pembimbing
Mahendrawathi ER, S.T., M.Sc., Ph.D.

JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2014

FINAL PROJECT - KS09 1336

**THE IMPACT OF ARTICLES VOLUME AND VARIATION
ON PACKING LEAD TIME IN PRODUCTION
DISTRIBUTION CENTER AT PT. XYZ USING
DUPLICATE GENETIC ALGORITHM**

Suviani Ningrum
NRP 5210100149

Supervisor
Mahendrawathi ER, S.T., M.Sc., Ph.D.

DEPARTMENT OF INFORMATION SYSTEM
Faculty of Information Technology
Institute of Technology Sepuluh Nopember
Surabaya 2014

**ANALISIS PENGARUH VOLUME DAN VARIASI ARTIKEL
TERHADAP *LEAD TIME* PENYELESAIAN PENGEPAKAN
DI *PRODUCTION DISTRIBUTION CENTER* PT.XYZ
DENGAN MENGGUNAKAN ALGORITMA *DUPLICATE
GENETIC***

Nama Mahasiswa : Suviani Ningrum
NRP : 5210 100 149
Jurusan : Sistem Informasi FTIF-ITS
Dosen Pembimbing I : Mahendrawathi ER, S.T, M.Sc, Ph.D.
Dosen Pembimbing II : -

Abstrak

PT. XYZ merupakan perusahaan manufaktur sepatu yang proses bisnis utamanya adalah memproduksi shoe dan upper shoe. Sepatu-sepatu yang telah jadi akan ditempatkan dalam gudang PDC (*Production Distribution Center*) sebelum didistribusikan. Pada PDC, sepatu-sepatu tersebut dipack sesuai dengan waktu yang dijadwalkan. Proses pengepakan dilakukan dalam sistem picking wave, yang mana dalam satu picking wave terdapat banyak artikel. Dalam prosesnya, artikel yang sama akan dikerjakan terlebih dahulu untuk semua M-Group setelah itu dilanjutkan dengan artikel berikutnya. Adanya system ini memungkinkan terdapatnya pengaruh volume dan variasi artikel dalam satu wave terhadap lead time penyelesaian pengepakan. Untuk itu, *PT. XYZ* perlu memodelkan proses bisnis pengepakan yang sesungguhnya dijalankan. Dari pemodelan tersebut akan dilakukan analisis bottleneck dan variasi artikel terhadap lead time penyelesaian pengepakan.

Dalam Tugas Akhir ini, pemodelan proses bisnis dilakukan dengan teknik penggalan proses dari data catatan kejadian modul warehouse management SAP *PT.XYZ*. Data kemudian diolah untuk dihasilkan model proses menggunakan Algoritma Duplicate Genetic. Penggunaan algoritma ini karena algoritma ini robust terhadap gangguan sebab melakukan penggalan model menurut frekuensi perilaku yang terjadi secara tepat. Keluaran dari penggalan proses

adalah petri net yang menunjukkan urutan dan relasi antar aktivitas pada proses bisnis yang diamati. Petri net ini kemudian akan diukur menggunakan tiga dimensi, yaitu dimensi fitness, dimensi presisi, dan dimensi struktur.

Dari hasil pemodelan yang telah dilakukan, kemudian dilanjutkan dengan analisis bottleneck dan analisis variasi artikel terhadap lead time. Dari analisis ini diketahui bahwa ternyata terdapat bottleneck pada salah satu aktivitas yang berdampak pada menumpuknya sepatu di gudang serta terdapat pengaruh variasi artikel terhadap lead time penyelesaian pengepakan. Dengan pengetahuan yang telah diperoleh dari analisis, selanjutnya diberikan sejumlah rekomendasi yang kiranya mampu mengatasi permasalahan yang terjadi di PT. XYZ.

Kata Kunci: SAP, ERP, Warehouse Management, Penggalan Proses, ProM Tools, Algoritma Duplicate Genetic.

THE IMPACT OF ARTICLES VOLUME AND VARIATION ON PACKING LEAD TIME IN PRODUCTION DISTRIBUTION CENTER AT PT. XYZ USING DUPLICATE GENETIC ALGORITHM

Student name : Suviani Ningrum
SIDN : 5210 100 149
Department : Sistem Informasi FTIF-ITS
Supervisor I : Mahendrawathi ER, S.T, M.Sc, Ph.D
Supervisor II : -

Abstract

PT. XYZ is a manufacturing company that produces shoe and upper shoes. Finished shoes is stored in the PDC (Production Distribution Center) warehouse before distributed to corporate warehouse and retailers. At PDC, these shoes will be packed according to the scheduled time. The process of packing is done by using picking wave system. One picking wave can include many articles. In this process, the same articles will be processed for all M-Group followed by next article. This system means that the packing lead time is influenced by volume and variety of articles in a wave. Thus, PT. XYZ needs to model the business process of packing at PDC department.

In this final project, business process modeling is done by process mining and extracting event log from the warehouse management module of PT. XYZ's SAP ERP System. Then the data is processed using Duplicate Genetics Algorithm to produce the process model. The output of process mining is a petri net, which shows the sequence and relationship between activities in the business processes. Petri net is measured using three dimensions, i.e. fitness, precision, and structure.

The business process model is then analyzed in terms of: 1) bottleneck and 2) impact of article variation on lead time. Based on the analysis it was found that there is a bottleneck in one of the activities that have an impact on the shoes stacked in the warehouse.

There appears to be a big influence of article's variations to the packing lead time. Several recommendations to overcome the problems that occur in the PT. XYZ are proposed based on the findings.

Keywords: SAP, ERP, Warehouse Management, Process Mining, ProM Tools, Algorithm *Duplicate Genetic*.

**ANALISIS PENGARUH VOLUME DAN VARIASI ARTIKEL
TERHADAP LEAD TIME PENYELESAIAN PENGEPAKAN
DI PRODUCTION DISTRIBUTION CENTER PT. XYZ
DENGAN MENGGUNAKAN ALGORITMA DUPLICATE
GENETIC**

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

Suviani Ningrum
NRP. 5210 100 149

Surabaya, Juli 2014

Ketua Jurusan Sistem Informasi

Dr. Eng. FEBRILIYAN SAMOHA, S.Kom., M.Kom.
NIP.19730219 199802 1 001

Halaman ini sengaja dikosongkan

**ANALISIS PENGARUH VOLUME DAN VARIASI ARTIKEL
TERHADAP LEAD TIME PENYELESAIAN PENGEPAKAN
DI PRODUCTION DISTRIBUTION CENTER PT. XYZ
DENGAN MENGGUNAKAN ALGORITMA DUPLICATE
GENETIC**

TUGAS AKHIR

**Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember**

Oleh :
**Suviani Ningrum
NRP. 5210 100 149**

**Disetujui Tim Penguji : Tanggal Ujian : 17 Juli 2014
Periode Wisuda : September 2014**

Mahendrawathi ER, S.T., M.Sc., Ph.D. (Pembimbing I)

Wiwik Anggraeni, S.Si., M.Kom.

(Pengujii I)

Irmasari Hafidz, S.Kom., M.Sc.

(Pengujii II)

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur dipanjatkan kepada Allah SWT yang senantiasa memberikan limpahan rahmat kepada Penulis selama menyelesaikan Tugas Akhir ini, yang berjudul **Analisis Pengaruh Volume dan Variasi Artikel terhadap Lead Time Penyelesaian Pengepakan di Production Distribution Center PT. XYZ dengan Menggunakan Algoritma Duplicate Genetic** dengan baik. Dengan selesainya pengerjaan Tugas Akhir ini maka selesai pula masa perjuangan penulis selama 8 semester di kampus Sistem Informasi ITS tercinta ini.

Tak lupa ucapan terima kasih penulis yang sebesar-besarnya kepada mereka yang senantiasa memberikan dukungan dan motivasi yang tiada hentinya:

- Bapak, mamak, adik-adik, dan segenap keluarga besar penulis yang tiada henti memberikan doa, kepercayaan, dan dukungan sehingga penulis dapat menyelesaikan pendidikan khususnya Tugas Akhir ini.
- Ibu Mahendrawati ER, S.T, M.Sc, Ph.D selaku pembimbing dan dosen wali yang telah banyak membantu, memberikan bimbingan, motivasi, waktu, dan nasehat kepada penulis. Tak ada kata yang mampu menggambarkan kebanggaan penulis karena telah mengenal ibu.
- Ibu Wiwik Anggraeni, S.Si., M.Kom dan Irmasari Hafidz, S.Kom, M.Sc., selaku dosen penguji yang telah banyak membantu melalui kritik, saran, dan motivasi kepada penulis. Sungguh merupakan kebanggaan bagi penulis karena telah diberikan kesempatan untuk mendapatkan pembelajaran dari ibu berdua.
- Bapak Akhmad Mukhlason selaku dosen wali di awal perkuliahan. Terima kasih atas bimbingan dan motivasinya.
- Bapak dan Ibu Dosen Pengajar beserta staf dan karyawan di Jurusan Sistem Informasi yang telah banyak memberikan ilmu

dan telah menjadi bagian dari kehidupan penulis selama 4 tahun ini.

- Kepada Pak Wahyudin, Pak Jarotd, Mbak Ila, Pak Mark, Bu Lena, Pak Guntur, Pak Atmari, dan seluruh keluarga besar Departemen Production Distribution Center PT. XYZ, terima kasih atas waktu, tempat, dan ilmu yang diberikan. Semoga kerjasama dan kebersamaan yang terjalin selama proses magang tetap berjalan dengan baik.
- Seluruh keluargaku, saudaraku FOXIS, teman-teman Laboratorium Sistem Pendukung Keputusan dan Inteligensia Bisnis (SPK-IB); Amalia, Nizar, Amel, Muhammad, dan semuanya yang tidak bisa saya sebutkan satu-satu, terima kasih atas persaudaraan, kebersamaan, dan perjuangan selama ini.
- Teman-teman terdekat penulis; Syum, Opi, Ikar, Dita, Fani, Dinda, Ichan, Devi, Djay, Fachri, dan Adis yang selalu mendukung dan memberi motivasi kepada penulis. Semoga persaudaraan ini tidak berhenti sampai disini. Teman seperjuangan penggalian proses; Ikar, Dita, dan Abdan. Terima kasih atas bantuan, dukungan, dan semangat yang telah diberikan.
- Segenap warga Sistem Informasi; Genesis, 8ios, Ae9is, Basilisk, dan Solaris yang juga telah menjadi bagian dari kehidupan penulis selama 4 tahun ini.
- Berbagai pihak yang belum sempat penulis sebutkan satu-persatu.

Penulis sadar bahwa apa yang dikerjakan ini masih jauh dari kata sempurna. Akhir kata, semoga buku tugas akhir ini dapat memberikan informasi dan pengetahuan yang bermanfaat, meskipun hanya satu dua patah kata.

Surabaya, Juni 2014

Penulis

DAFTAR ISI

Abstrak	vii
Abstract	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan permasalahan.....	4
1.3. Batasan Permasalahan	4
1.4. Tujuan	5
1.5. Manfaat	5
1.6. Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA	7
2.1. Proses Bisnis	7
2.2. Volume dan Variasi.....	8
2.3. Penggalan Proses	8
2.4. Catatan Kejadian (<i>Event Log</i>)	11
2.5. Petri Net	13
2.6. ProM Framework	14
2.7. Ekstraksi Data	16
2.8. Algoritma Penggalan Proses	19
2.8.1. Algoritma <i>Duplicate Genetic</i>	19
2.8.2. Algoritma Genetika.....	30
2.8.3. Perbedaan Algoritma <i>Duplicate Genetic</i> dan Algoritma Genetika.....	31
2.9. Pengukuran Performa Model.....	32
2.9.1. <i>Fitness</i>	32
2.9.2. Presisi	33
2.9.3. Struktural	34
2.10. Enterprise Resource Planning (ERP).....	35
2.11. Perangkat Lunak <i>SAP</i>	36

2.11.1. Modul <i>Warehouse Management</i>	36
2.12. Penelitian Sebelumnya	37
BAB III METODE PENGKERJAAN TUGAS AKHIR	41
3.1. Persiapan	41
3.1.1. Studi Literatur	41
3.1.2. Pengumpulan Data	41
3.1.3. Standardisasi Catatan Kejadian	42
3.2. Penggalan Proses	42
3.2.1. Pembuatan Model Proses	42
3.3. Pengukuran Model	42
3.4. Analisis Hasil	43
3.5. Pembuatan buku Tugas Akhir	43
BAB IV PEMODELAN PROSES BISNIS	45
4.1. Studi Kasus	45
4.1.1. Divisi <i>Production Distribution Center</i> (PDC).....	45
4.2. Persiapan	51
4.2.1. Pengumpulan Data	51
4.2.2. Standardisasi Catatan Kejadian	61
4.3. Penggalan Proses	65
4.3.1. Masukan.....	65
4.3.2. Proses.....	65
4.3.3. Keluaran.....	80
4.4. Pengukuran Model	80
4.4.1. Masukan Pengukuran Model.....	80
4.4.2. Proses Pengukuran Model.....	81
4.4.3. Keluaran Pengukuran Model.....	88
BAB V ANALISIS MODEL	91
5.1. Analisis <i>Bottleneck</i>	91
5.2. Analisis Waktu Tunggu.....	98
5.3. Analisis Variasi terhadap Lead Time	104
BAB VI KESIMPULAN DAN SARAN	115
6.1. Kesimpulan	115
6.2. Saran	116
DAFTAR PUSTAKA	117
RIWAYAT PENULIS	119

LAMPIRAN A PERBANDINGAN MODEL PARAMETER DEFAULT DAN PARAMETER PERCOBAAN	A-1
LAMPIRAN B DATA HASIL EKSTRAKSI	B-1
LAMPIRAN C DATA CATATAN KEJADIAN	C-1

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 2.1 Contoh catatan kejadian.....	12
Tabel 2.2 Sumber informasi untuk menentukan aktivitas	17
Tabel 4.1 Pembagian artikel tiap shift karyawan dalam 1 wave	49
Tabel 4.2 Tabel Dasar modul <i>Warehouse</i>	54
Tabel 4.3 Pemetaan aktivitas dengan atribut pada tabel WM	55
Tabel 4.4 Daftar atribut yang diekstrak	56
Tabel 4.5 Daftar atribut yang pasti dipilih untuk diekstrak.....	57
Tabel 4.6 Alur skenario yang mungkin terjadi	57
Tabel 4.7 Potongan Catatan Kejadian.....	61
Tabel 4.8 Potongan parameter percobaan	68
Tabel 4.9 Rincian skenario	81
Tabel 4.10 Skenario pada catatan kejadian.....	82
Tabel 4.11 <i>Log replay</i> : Start.Good receipt.....	83
Tabel 4.12 <i>Log replay</i> : High Rack	84
Tabel 4.13 <i>Log replay</i> : Start.Picking.....	84
Tabel 4.14 <i>Log replay</i> : Start.Packing	84
Tabel 4.15 <i>Log replay</i> : Start.Loading.....	85
Tabel 4.16 <i>Log replay</i> : End.....	85
Tabel 4.17 Rekapitulasi jumlah token skenario 1	86
Tabel 4.18 Ringkasan Hasil Pengukuran Performa Model	88
Tabel 5.1 Informasi <i>bottleneck</i>	93
Tabel 5.2 Perbandingan <i>bottleneck</i> pada empat wave.....	95
Tabel 5.3 Perbandingan dotted chart dua wave	101
Tabel A.1 Perbandingan model parameter <i>default</i> dan parameter percobaan	A-1
Tabel B.1 Hasil ekstraksi <i>file</i> 21145.....	B-1
Tabel B.2 Hasil ekstraksi <i>file</i> 21207.....	B-2
Tabel B.3 Hasil ekstraksi <i>file</i> 21110.....	B-3
Tabel B.4 Hasil ekstraksi <i>file</i> 21180.....	B-4
Tabel C.1 Catatan kejadian file EventLog.....	C-1

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 1.1	Prosedur packing di PDC.....	3
Gambar 2.1	3 Posisi dalam penggalian proses.....	9
Gambar 2.2	3 Tipe dasar penggalian proses berdasarkan masukan & keluaran.....	10
Gambar 2.3	Contoh catatan kejadian dalam format MXML.....	13
Gambar 2.4	Model Petri Net	14
Gambar 2.5	Gambaran umum ProM <i>framework</i>	15
Gambar 2.6	Prosedur utama ekstraksi data.....	17
Gambar 2.7	Contoh model yang mengandung duplicate task.....	20
Gambar 2.8	Model penggalian yang terlalu spesifik	22
Gambar 2.9	Langkah utama DGA	24
Gambar 2.10	Contoh causal matrix	25
Gambar 2.11	Individu yang dihasilkan causal matrix.....	26
Gambar 2.12	Individu yang berupa model proses	26
Gambar 2.13	Model proses dengan presisi yang berbeda	33
Gambar 2.14	Struktur <i>Warehouse</i>	37
Gambar 3.1	Metodologi pengerjaan TA	44
Gambar 4.1	Proses bisnis PDC secara umum	46
Gambar 4.2	Bagan pengelompokkan sepatu dalam 1 picking wave	49
Gambar 4.3	<i>Transfer Orders: List of Resident Documents</i>	58
Gambar 4.4	Tampilan data <i>file</i> 21145	59
Gambar 4.5	<i>Formula</i> menggabungkan kolom	60
Gambar 4.6	Hasil penggabungan kolom.....	60
Gambar 4.7	Tampilan data di Disco	62
Gambar 4.8	Ekspor hasil konversi dengan Disco	63
Gambar 4.9	Proses menyimpan hasil konversi	63
Gambar 4.10	Potongan hasil konversi catatan kejadian berformat *.mxml	64
Gambar 4.11	Potongan hasil konversi catatan kejadian sebuah aktivitas	64
Gambar 4.12	Tampilan ProM membaca catatan kejadian	65
Gambar 4.13	Pemilihan Algoritma <i>Duplicate Genetic</i> pada ProM..	66

Gambar 4.14 Tampilan mengatur parameter pada ProM	66
Gambar 4.15 Grafik parameter population size	70
Gambar 4.16 Grafik parameter initial population type	71
Gambar 4.17 Grafik parameter maximum number generation	72
Gambar 4.18 Grafik parameter seed	73
Gambar 4.19 Grafik parameter power value.....	74
Gambar 4.20 Grafik parameter elitism rate	75
Gambar 4.21 Grafik parameter <i>fitness</i> type	75
Gambar 4.22 Grafik parameter selection method type	76
Gambar 4.23 Grafik parameter crossover type	77
Gambar 4.24 Grafik parameter crossover rate	78
Gambar 4.25 Grafik parameter mutation rate	79
Gambar 4.26 Model proses yang dihasilkan oleh parameter default	80
Gambar 4.27 Skenario yang terbentuk pada Disco	81
Gambar 4.28 Nilai presisi model pada ProM.....	87
Gambar 5.1 Melakukan performance analysis with petri net.....	91
Gambar 5.2 Mengubah time measured	92
Gambar 5.3 Hasil <i>bottleneck</i> model proses bisnis	92
Gambar 5.4 Melakukan dotted chart analysis	98
Gambar 5.5 Tampilan dotted chart analysis	99
Gambar 5.6 Pengaturan dotted chart analysis.....	99
Gambar 5.7 Pengaturan warna komponen	100
Gambar 5.8 Dotted chart artikel 70145258772.....	103
Gambar 5.9 Perbandingan rata-rata lead time.....	105
Gambar 5.10 Grafik volume terhadap lead time	106
Gambar 5.11 Perbandingan lead time per artikel.....	107
Gambar 5.12 Grafik volume terhadap lead time	108
Gambar 5.13 Grafik volume artikel terhadap lead time dalam 1 wave	109
Gambar 5.14 Perbandingan artikel dan volume sama.....	110
Gambar 5.15 Perbandingan artikel beda dan volume sama	111
Gambar 5.16 Perbandingan artikel sama dan volume beda	112

BAB I

PENDAHULUAN

Pada bagian pendahuluan akan dibahas mengenai latar belakang, perumusan masalah, batasan, tujuan, dan manfaat tugas akhir ini.

1.1. Latar Belakang

Enterprise Resource Planning (ERP) adalah suatu sistem informasi perusahaan yang memungkinkan perusahaan mengotomatiskan dan mengintegrasikan berbagai proses bisnis utamanya (Hall, 2008). Informasi yang ada pada *ERP* akan disimpan dalam *database* tunggal dan bersifat *real-time* serta dapat diambil saat dibutuhkan oleh berbagai fungsi. Salah satu informasi *ERP* yaitu yang berbentuk catatan kejadian (*event log*). Catatan kejadian merupakan catatan historis seluruh pengguna *ERP* perusahaan mengenai proses bisnis apa saja yang dilakukan oleh pengguna secara *real-time*. Informasi dalam catatan kejadian dapat dianalisis untuk kemudian dimodelkan proses bisnis aktual perusahaan.

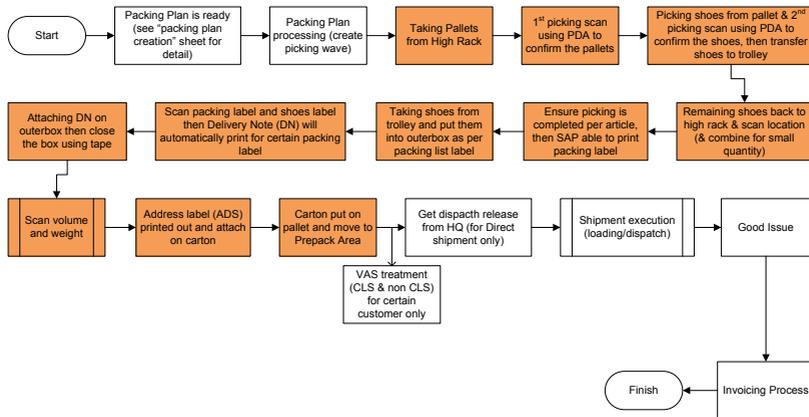
Salah satu teknik memodelkan catatan kejadian *ERP* adalah Penggalian Proses (*Process Mining*). Penggalian proses adalah suatu disiplin ilmu penelitian baru yang terletak diantara komputerisasi intelegensia, *data mining*, analisis, dan pemodelan proses. Ide dari penggalian proses adalah untuk menemukan, memantau, dan meningkatkan proses sebenarnya dengan mengekstraksi pengetahuan dari catatan kejadian yang ada dalam sistem (Aalst, 2011). Catatan kejadian berisi data mengenai aktivitas yang telah dilakukan oleh perusahaan, waktu pelaksanaan, dan eksekutor. Catatan kejadian bisa didapatkan dengan mengekstraksi *database* sistem informasi yang ada.

Dalam perkembangannya terdapat beberapa algoritma yang digunakan dalam penggalian proses, diantaranya algoritma *alpha*, algoritma *heuristic* miner, dan algoritma genetika. Berdasarkan

algoritma-algoritma tersebut, diketahui bahwa algoritma genetika mampu mengatasi kelemahan algoritma *alpha* dan algoritma *heuristic miner*. Kelebihan algoritma genetika adalah mampu mendeteksi adanya *short-loop*, kejadian tersembunyi (*hidden activity*), *robust* terhadap *log* yang mengandung *noise*, *imcompleteness*, serta mampu mendeteksi adanya pilihan tidak bebas (*non-free choice*). Namun algoritma ini kurang stabil dengan *AND split/join* terutama jika terjadi model percabangan paralel yang panjang dan tak mampu mendeteksi adanya kejadian duplikat (*duplicate task*), serta membutuhkan waktu yang lama dalam memproses data. Kelemahan algoritma genetika mampu ditutupi oleh algoritma *duplicate genetic*. Algoritma ini merupakan perluasan dari Algoritma Genetika. Kelebihan algoritma ini adalah waktu yang dikonsumsi untuk memproses data lebih cepat dan mampu mendeteksi kejadian duplikat (*duplicate task*) yang tak dapat ditangani oleh algoritma genetika. Namun algoritma ini memerlukan iterasi yang lebih banyak untuk mendapatkan solusi model yang lengkap dan presisi jika dibandingkan dengan algoritma genetika (Medeiros A. K., Genetic Process Mining, 2006). Berdasarkan kelebihan inilah DGA diterapkan dalam studi kasus tugas akhir ini.

PT. XYZ merupakan salah satu perusahaan yang sudah menerapkan *ERP* yaitu *SAP* dalam menjalankan proses bisnisnya. PT. XYZ merupakan perusahaan manufaktur yang proses bisnis utamanya adalah memproduksi *shoe* dan *upper shoe*. Sepatu-sepatu yang telah jadi akan ditempatkan dalam gudang PDC (*Production Distribution Center*) terlebih dahulu sebelum didistribusikan. Sepatu-sepatu ini merupakan sepatu siap pakai dan lengkap dengan *innerbox*-nya atau kardus pembungkus. Sebelum sepatu didistribusikan, sepatu di-*packing* terlebih dahulu. *Packing* merupakan proses pengemasan sepatu ke dalam *outerbox*. *Packing* dilakukan sesuai dengan waktu yang dijadwalkan.

Proses *packing* dilakukan dengan sistem *picking wave*¹. Dalam satu *picking wave* terdapat banyak artikel². Saat proses *packing*, artikel yang sama diselesaikan terlebih dahulu untuk semua *M-Group*³ setelah itu baru menyelesaikan artikel yang lain.



Gambar 1.1 Prosedur packing di PDC
Sumber: PT.XYZ

Hal ini menyebabkan beberapa persoalan (letak persoalan digambarkan pada Gambar 1.1 di bagian bagan berwarna oranye yang mengacu pada Gambar 4.1 bagian PDC-Storage 200 dan PDC-Storage 916), yaitu:

¹ *Picking wave* adalah sistem atau alur pengambilan sepatu dari *high rack* hingga sepatu selesai di-pack.

² Artikel: Jenis sepatu yang dihasilkan oleh PT. XYZ. Terdapat 3 kelompok jenis sepatu yang dihasilkan, yaitu pria, wanita, dan anak-anak. Kelompok pria terdapat 518 jenis sepatu, kelompok wanita terdapat 680 jenis sepatu, dan kelompok anak-anak terdapat 274 jenis sepatu.

³ *M-Group* adalah kumpulan dari satu atau beberapa artikel yang memiliki kriteria yang sama, seperti metode transportasi (MoT) yang sama, tanggal tenggat waktu pengiriman yang sama, rute pengiriman yang sama, pelanggan yang sama, dll.

1. *Picking Wave* yang memiliki *volume* dan variasi artikel lebih sedikit, prosesnya lebih mudah dan mungkin lebih cepat dibandingkan *picking wave* yang memiliki *volume* dan variasi yang lebih banyak.
2. Hal ini mengakibatkan adanya indikasi pengaruh terhadap *lead time* (waktu tunggu) penyelesaian pengepakan.

Dengan permasalahan yang terjadi, diperlukan pemodelan proses bisnis yang sesungguhnya dijalankan di bagian gudang PDC. Dari pemodelan proses bisnis akan dilakukan analisis dengan menggunakan Algoritma *Duplicate Genetic* untuk mengetahui *lead time* penyelesaian pengepakan serta pengaruh *volume* dan variasi artikel sepatu terhadap *lead time* penyelesaian pengepakan di PDC PT. XYZ. Analisis tersebut nantinya dapat digunakan untuk mengidentifikasi peluang peningkatan proses.

1.2. Rumusan permasalahan

Berdasarkan latar belakang yang sudah dipaparkan diatas, maka perumusan masalah dalam tugas akhir ini adalah:

- a. Bagaimana membentuk catatan kejadian *SAP ERP* dari proses *packing* PDC di PT.XYZ?
- b. Bagaimana model proses *packing* PDC yang dihasilkan oleh Algoritma *Duplicate Genetic* berdasarkan catatan kejadian *SAP ERP*?
- c. Apa pengaruh *volume* dan variasi artikel sepatu terhadap *lead time* penyelesaian pengepakan di PDC?
- d. Rekomendasi apa yang dapat diberikan kepada PT. XYZ untuk meningkatkan proses yang terjadi?

1.3. Batasan Permasalahan

Batasan permasalahan dari tugas akhir ini adalah sebagai berikut:

- a. Sumber data berupa catatan kejadian dari modul *Warehouse* di PT. XYZ yang terkait dengan proses *packing* PDC.
- b. Sumber data catatan kejadian berasal dari aplikasi *SAP* dengan melakukan proses ekstraksi langsung dari basis data *SAP*.

- c. Dimensi pengukuran performa model yang digunakan adalah dimensi *fitness*, presisi, dan struktural.

1.4. Tujuan

Tujuan dari tugas akhir ini adalah memodelkan dan menganalisis proses bisnis *packing* PDC PT.XYZ dalam rangka identifikasi peluang peningkatan proses.

1.5. Manfaat

Manfaat yang diberikan tugas akhir ini adalah:

- a. Membantu untuk mengetahui cara mengekstraksi catatan kejadian *SAP ERP*.
- b. Membantu mengetahui cara membentuk model proses dengan menggunakan Algoritma *Duplicate Genetic*.
- c. Membantu mengetahui pengaruh *volume* dan variasi artikel sepatu terhadap *lead time* penyelesaian pengepakan di PDC.
- d. Membantu mengevaluasi proses bisnis *packing* PDC.

1.6. Sistematika Penulisan

Sistematika penulisan buku tugas akhir ini dibagi dalam bab sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini dibahas mengenai latar belakang, rumusan masalah, tujuan, manfaat, dan sistematika penulisan Buku Tugas Akhir ini.

BAB II TINJAUAN PUSTAKA

Pada bab ini dibahas mengenai tinjauan pustaka yang mendukung dalam pengerjaan buku Tugas Akhir ini. Teori-teori tersebut meliputi: proses bisnis, volume dan variasi produk, penggalian proses, catatan kejadian, Petri Net, ProM, Algoritma *Duplicate Genetic*, pengukuran performa, *Enterprise Resource Planning*, dan *SAP*.

BAB III METODE Pengerjaan Tugas Akhir

Pada bab ini dibahas mengenai metode pengerjaan tugas akhir ini. Terdapat empat fase dalam pengerjaan tugas akhir ini, yaitu

fase persiapan, penggalian proses, pengukuran model, dan analisis hasil. Selain keempat fase tersebut, terdapat satu tambahan aktivitas pengerjaan tugas akhir lainnya, yaitu tahap pembuatan buku Tugas Akhir.

BAB IV PEMODELAN PROSES BISNIS

Pada bab ini dijelaskan mengenai perancangan pengumpulan data, standardisasi catatan, implementasi catatan kejadian ke dalam penggalian proses menggunakan algoritma *duplicate genetic*, dan pengukuran performa model yang dihasilkan dari penggalian proses.

BAB V ANALISIS HASIL MODEL PROSES

Pada bab ini akan dilakukan analisis hasil pemodelan. Terdapat beberapa analisis dilakukan yaitu analisis *bottleneck*, analisis waktu tunggu, dan analisis variasi artikel terhadap lead time.

BAB V ANALISIS HASIL MODEL PROSES

Pada bab ini dijelaskan kesimpulan yang didapatkan dari seluruh proses pengerjaan tugas akhir beserta saran untuk proses pengembangan selanjutnya.

BAB II

TINJAUAN PUSTAKA

Pada bagian ini akan dibahas mengenai tinjauan pustaka dan teori-teori yang mendukung dalam pengerjaan tugas akhir. Teori-teori tersebut antara lain; teori proses bisnis, penggalian proses, catatan kejadian, Petri Net, algoritma Heuristic miner, ekstraksi data, evaluasi, perencanaan sumber daya perusahaan, dan SAP.

2.1. Proses Bisnis

Proses bisnis merupakan bagian dari bisnis. Proses bisnis adalah abstraksi yang menampilkan hubungan kerjasama antara sumber daya dan proses perubahan sumber daya dalam bisnis. Proses bisnis sendiri lebih ditekankan pada bagaimana aktivitas-aktivitas dikerjakan, bukan pada hasil yang dikeluarkan. Proses bisnis memiliki tujuan, masukan, dan keluaran yang dihasilkan. Adapun masukan dalam proses bisnis berupa sumber daya yang digunakan untuk proses, seperti material mentah yang digunakan dalam proses manufaktur. Sedangkan keluaran dalam proses bisnis berupa ketercapaian tujuan dan hasil pokok dari proses, seperti produk jadi dalam proses manufaktur (Erikson & Penker, 2000).

Teori lain tentang proses bisnis dikemukakan oleh Davenport (1992) yang menyatakan bahwa proses bisnis adalah aktivitas spesifik berurutan, yang memiliki waktu, tempat, sebuah awal, sebuah penyelesaian, masukan, dan keluaran yang jelas. Michael Porter (1998) menjelaskan bahwa setiap perusahaan terbentuk dari sejumlah proses bisnis yang mana setiap proses bisnis tersebut menghasilkan nilai bagi konsumen. Porter juga menjelaskan bahwa konsumen tidak hanya konsumen yang berasal dari luar perusahaan tetapi juga karyawan atau pihak eksekutif di dalamnya.

Berdasarkan beberapa teori yang telah dijabarkan sebelumnya, dapat disimpulkan bahwa proses bisnis adalah sekumpulan aktivitas yang berurutan, memiliki masukan, dan keluaran yang dihasilkan sesuai dengan tujuan dari proses bisnis tersebut.

2.2. Volume dan Variasi

Permintaan terhadap variasi produk yang dihasilkan oleh perusahaan manufaktur semakin hari semakin bertambah. Menurut Pine (1993) terdapat tiga faktor yang mendorong permintaan untuk meningkatkan variasi, yaitu:

- Meningkatkan heterogenitas dalam target pasar perusahaan
- Memperluas distribusi pendapatan dalam pasar
- Memperlambat pertumbuhan dalam pasar

Faktor-faktor tersebut memberikan tekanan pada perusahaan untuk meningkatkan variasi produk.

Menurut Martin & Ishii (1997), semakin banyak variasi maka semakin banyak pula tahapan proses pembuatan variasi produk. Adapun area yang terkena dampak dari meningkatnya variasi, meliputi:

- Logistik
- Kualitas
- Perubahan kapasitas dikarenakan pengaturan system
- Inventori bahan mentah
- Proses dalam inventori
- Inventori barang jadi
- dll

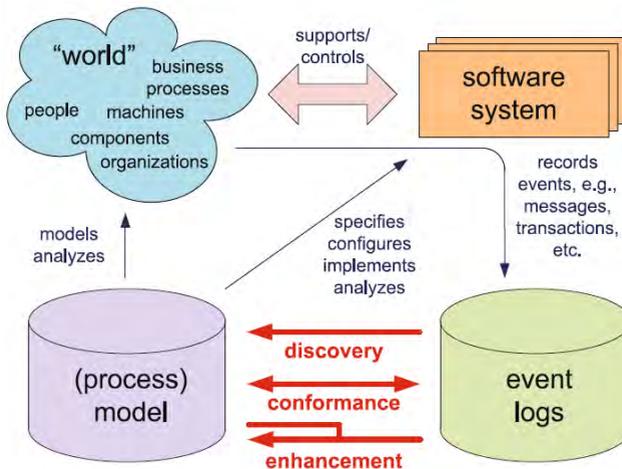
Banyaknya tahapan proses yang dilakukan untuk pembuatan variasi tersebut dapat diminimalkan dengan mengurangi jumlah tahapan proses. Dengan meminimalkan jumlah tahapan proses, maka juga dapat meminimalkan titik penyimpanan inventori, kompleksitas logistik, dll.

2.3. Penggalian Proses

Penggalian proses (*Process mining*) merupakan disiplin penelitian yang terletak diantara komputersasi intelegensia, *data mining*, analisis, dan pemodelan proses. Teknik penggalian proses menggunakan *event log* (rekaman tindakan aktual) sebagai masukannya (Aalst, Weijters, & Maruste, 2004). Penggalian proses memberikan wawasan mengenai dependensi kontrol aliran, penggunaan data, pemanfaatan sumberdaya, dan berbagai kinerja

yang berhubungan dengan statistik (Utami, 2013). Terdapat dua alasan penggalian proses sangat bermanfaat; pertama, dapat digunakan sebagai perangkat untuk mengetahui bagaimana orang atau sebuah prosedur bekerja. Kedua, penggalian proses dapat digunakan untuk membandingkan proses aktual dengan proses bisnis yang telah didefinisikan sebelumnya (Andreswari, 2013).

Penggalian proses dapat diterapkan pada sistem yang luas. Sistem ini bisa berupa sistem informasi, seperti ERP, atau sistem informasi lainnya selain ERP. Adapun sistem ini mampu menghasilkan *event logs*, yaitu rekaman tindakan aktual. Penggalian proses memiliki tiga tipe (Aalst & et.al, 2012), yaitu:



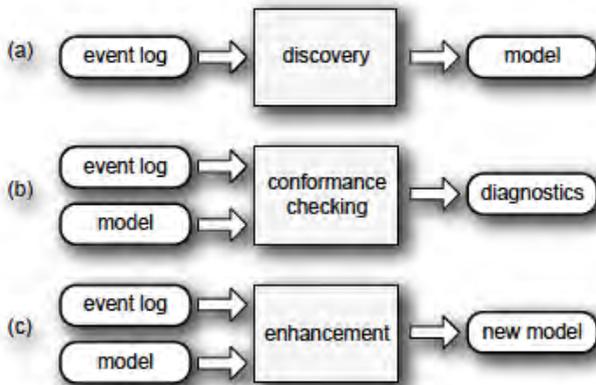
Gambar 2.1 3 Posisi dalam penggalian proses
Sumber: (Aalst & et.al, 2012)

1. Penemuan (*discovery*)
 Penemuan merupakan tipe penggalian proses yang menggunakan *event log* untuk membentuk model proses tanpa menambahkan informasi diluar dari apa yang dihasilkan.
2. Kesesuaian (*conformance*)

Kesesuaian merupakan tipe penggalian proses yang membandingkan model proses yang dibentuk berdasarkan *event log* dengan model proses yang ada. Kesesuaian juga dapat digunakan untuk memeriksa jika kenyataan, seperti yang tercatat dalam *event log*, sesuai dengan model yang telah didefinisikan.

3. Peningkatan (*enhancement*)

Peningkatan merupakan tipe penggalian proses yang memberikan perbaikan terhadap model proses yang dibentuk berdasarkan rekaman tindakan aktual (*event logs*).



Gambar 2.2 3 Tipe dasar penggalian proses berdasarkan masukan & keluaran
 Sumber: (Aalst & et.al, 2012)

Gambar 2.2 mendeskripsikan tiga tipe penggalian proses. Tipe pertama, yaitu *discovery*. *Discovery* menggunakan rekaman tindakan aktual untuk menghasilkan model. Tipe kedua, yaitu *conformance*. *Conformance* menggunakan rekaman tindakan actual dan model sebagai masukan untuk menghasilkan diagnosa informasi perbedaan dan kesamaan antara model dengan *log*. Tipe ketiga, yaitu *enhancement*. *Enhancement* menggunakan rekaman tindakan aktual dan model sebagai masukan untuk menghasilkan model baru. Informasi penggalian proses dapat dilihat dari empat perspektif (Aalst W. M., 2011), yaitu:

1. Perspektif aliran-kontrol
Perspektif ini berfokus pada aliran kontrol dan bertujuan untuk menemukan karakter terbaik dari semua jalur aktivitas, seperti urutan aktivitas-aktivitas.
2. Perspektif organisasional
Perspektif ini berfokus pada informasi mengenai sumber daya tersembunyi yang ada dalam *log*, seperti siapa aktor (pengguna, sistem, peran, dan departemen) yang terlibat dan bagaimana hubungannya. Adapun perspektif ini bertujuan untuk mengetahui struktur organisasi dengan mengklasifikasikan aktor-aktor dalam kaitannya dengan peran dan unit organisasionalnya dan jaringan sosial yang menunjukkan hubungan antar aktor.
3. Perspektif kasus
Perspektif ini berfokus pada properti kasus, seperti karakter data yang melekat pada proses yang sedang diamati. Karakter data yang sama kemudian dikumpulkan dan dimasukkan ke dalam sebuah kasus yang sama.
4. Perspektif waktu
Perspektif ini berfokus pada waktu dan jumlah kemunculan atas kejadian-kejadian. Misalnya menemukan *bottleneck*, mengukur tingkatan layanan, dan memprediksi sisa waktu proses dalam menjalankan suatu kasus.

2.4. Catatan Kejadian (*Event Log*)

Catatan kejadian merupakan kumpulan catatan aktivitas pengguna terhadap sistem atau aplikasi sistem informasi. Sebuah catatan kejadian mampu memberikan informasi mengenai sumber daya yang digunakan dalam melakukan aktivitas, misalnya pekerja yang mengeksekusi suatu tugas.

Menurut Aalst, dkk (2003) mengungkapkan bahwa kejadian yang mampu dicatat adalah kejadian yang mengacu pada aktivitas (*activity*), kejadian yang mengacu pada kasus (*case*), kejadian yang memiliki subjek yang juga mengacu sebagai pemicu (*originator*), dan kejadian yang memiliki catatan waktu (*timestamp*). Terdapat

beberapa atribut yang termasuk dalam catatan kejadian. Atribut-atribut tersebut sebagai berikut:

- Kasus : Rangkaian aktivitas dalam catatan (*log*)
- Id kasus : Tanda pengenal untuk setiap kasus
- Aktivitas : Hal yang dilakukan dalam sebuah kasus
- Keterangan waktu : Properti yang menunjukkan waktu dieksekusinya aktivitas
- Eksekutor : Properti yang menunjukkan pelaku aktivitas

Tabel 2.1 menunjukkan contoh catatan kejadian yang terdiri atas atribut id kasus, aktivitas, eksekutor, dan keterangan waktu eksekusi aktivitas.

Tabel 2.1 Contoh catatan kejadian
Sumber: (Arsad, 2013)

No. Kasus	Aktivitas	Eksekutor	Keterangan Waktu
kasus 1	activity A	John	9-3-2004:15.01
kasus 2	activity A	John	9-3-2004:15.12
kasus 3	activity A	Sue	9-3-2004:16.03
kasus 3	activity B	Carol	9-3-2004:16.07
kasus 1	activity B	Mike	9-3-2004:18.25
kasus 1	activity C	John	10-3-2004:09.23
kasus 2	activity C	Mike	10-3-2004:10.34
kasus 4	activity A	Sue	10-3-2004:10.35
kasus 2	activity B	John	10-3-2004:12.34
kasus 2	activity D	Pete	10-3-2004:12.50
kasus 5	activity A	Sue	10-3-2004:13.05
kasus 4	activity C	Carol	11-3-2004:10.12
kasus 1	activity D	Pete	11-3-2004:10.14
kasus 3	activity C	Sue	11-3-2004:10.44

No. Kasus	Aktivitas	Eksekutor	Keterangan Waktu
kasus 3	activity D	Pete	11-3-2004:11.03
kasus 4	activity B	Sue	11-3-2004:11.18
kasus 5	activity E	Clare	11-3-2004:12.22
kasus 5	activity D	Clare	11-3-2004:14.34
kasus 4	activity D	Pete	11-3-2004:15.56

Catatan kejadian agar dapat diproses menggunakan aplikasi ProM harus diubah dahulu ke dalam format MXML (*Mining XML*). Gambar 2.3 adalah contoh catatan kejadian yang telah diubah ke dalam format MXML.

```

<AuditTrailEntry>
  <WorkflowModelElement/> Task A </Wf.M.E.>
  <EventType/> complete </EventType/>
  <TimeStamp/> 2005-10-26T12:37:33... </TimeStamp/>
  <Originator/> John Doe </Originator/>
  <Data/>
    <Attribute name="x"/> 1 </Attribute/>
    <Attribute name="y"/> whatever </Attribute/>
  </Data/>
</AuditTrailEntry/>

```

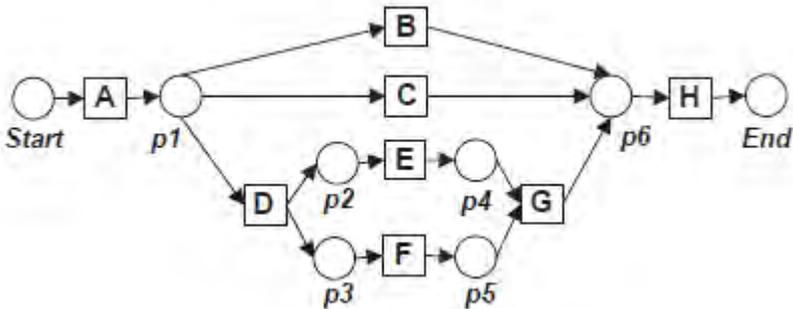
Gambar 2.3 Contoh catatan kejadian dalam format MXML
Sumber: (Arsad, 2013)

2.5. Petri Net

Petri Net adalah *tool* yang digunakan untuk mendeskripsikan dan mempelajari sistem pengelolaan informasi (Murata, 1989). Petri Nets merupakan *tool* pemodelan yang berbasis grafis dan matematis. Sebagai *graphical tool*, Petri Nets dapat digunakan sebagai bantuan komunikasi visual untuk *flow chart*, diagram balok, dan jaringan. Sebagai *mathematical tool*, Petri Net memungkinkan

untuk mengatur *equations*, *algebraic equations*, dan model matematis lainnya.

Petri Nets adalah struktur dinamis yang terdiri atas satu set transisi (*transitions*), dan *places*. *Transitions* merepresentasikan aksi dan disimbolkan dengan kotak. *Place* merepresentasikan kondisi yang harus dipenuhi sebelum aksi dilakukan dan disimbolkan dengan lingkaran. Kedua simbol ini dihubungkan oleh *arcs* atau anak panah yang menunjukkan hubungan dari *place* ke *transition* atau sebaliknya. *Place* dapat memiliki atau tidak memiliki *token* (titik hitam). *Transition* bisa dijalankan apabila setiap *place* yang menjadi masukan mengandung sebuah *token* (Medeiros, Weijters, & Aalst, Using Genetic Algorithms to Mine Process Models: Representation, Operators and Results, 2005). Model dari Petri Net ditunjukkan pada Gambar 2.4.

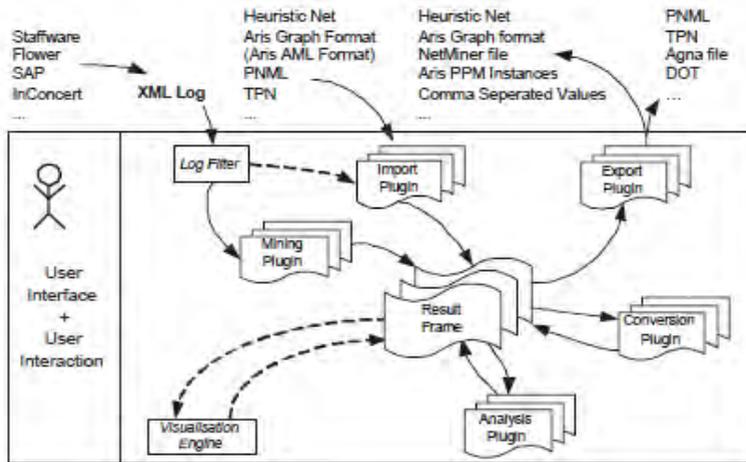


Gambar 2.4 Model Petri Net
 Sumber: (Medeiros, Weijters, & Aalst, 2005)

2.6. ProM Framework

ProM merupakan *open-source tools* yang secara khusus dibuat untuk mendukung pengembangan penggalian proses (Verbeek, Dongen, Mendling, & Aalst, 2006). *Framework* ini bersifat *pluggable*, yang artinya pengguna bebas menambah ataupun mengurangi *plug-in* yang ada dalam ProM *framework* sesuai dengan kebutuhan tanpa perlu takut terjadi perubahan pada *framework*. *Plug-*

in yang ditambahkan pada ProM pada dasarnya merupakan implementasi algoritma yang biasanya digunakan dalam lingkup penggalian proses dan sesuai dengan *framework*.



Gambar 2.5 Gambaran umum ProM *framework*

Sumber: (Dongen, Medeiros, Verbeek, Weijters, & Aalst, 2005)

Penjelasan detail mengenai gambaran umum ProM dapat dilihat pada Gambar 2.5. Berdasarkan Gambar 2.5, dapat diketahui bahwa ProM mampu membaca catatan dengan format XML melalui komponen penyaring catatan (*Log filter*). Selain itu, ProM juga mampu menghasilkan berbagai macam model, salah satunya Petri Net yang divisualisasikan melalui *visualization engine*. Fungsi ini dilakukan oleh komponen *mining plugin*. *Import plugin* sebagaimana namanya merupakan bagian yang menangani instalasi *plug-in* algoritma baru. ProM juga memiliki *analysis plugin* yang merupakan komponen yang berfungsi dalam hal perhitungan terhadap model, seperti perhitungan *fitness*, dan sebagainya. *Export plugin* merupakan komponen yang memiliki fungsi “simpan menjadi”, misalnya Petri Net (simpan ke dalam format PNML). *Conversion*

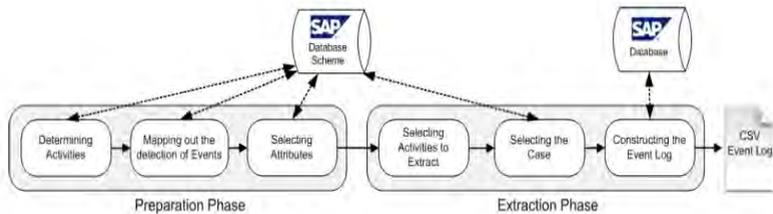
plugin memiliki fungsi mengonversi format data yang berbeda, misalnya dari EPC ke Petri Net.

2.7. Ekstraksi Data

Ekstraksi data merupakan langkah yang penting dalam penggalian proses. Langkah ini berupa ekstraksi data catatan kejadian dari SAP. Struktur dan konten suatu catatan kejadian menentukan gambaran proses dan hasil dari penggalian proses yang dapat ditampilkan. Terdapat prosedur dan aspek penting yang perlu diperhatikan dalam pengekstraksian data, diantaranya yaitu (Piessens, 2011):

- Penentuan Proyek (*Project Decision*)
Penentuan proyek berfungsi untuk memastikan bahwa catatan kejadian yang akan diekstrak sudah mencakup gambaran yang tepat mengenai proses sehingga tak perlu dilakukan pengulangan pengekstraksian sebelum struktur catatan kejadian memenuhi ekspektasi. Pada langkah ini dilakukan penentuan cakupan, tujuan, dan focus dari proses ekstraksi data.
- Prosedur
Langkah ini berisi mengenai lima hal penting yang perlu diketahui ketika membuat catatan kejadian untuk sebuah proses bisnis, yaitu:
 - a. Aktivitas (*activities*) yang terdapat dalam proses bisnis
 - b. Mengenali jumlah kemunculan (*occurrence*) setiap aktivitas
 - c. Atribut (*attribute*) yang termasuk dalam tiap aktivitas
 - d. Kasus (*case*) yang menentukan cakupan proses bisnis
 - e. Format keluaran (*output format*) dari hasil catatan kejadian

Gambar 2.6 merupakan diagram aliran prosedur utama dalam proses ekstraksi data.



Gambar 2.6 Prosedur utama ekstraksi data

Sumber: (Piessens, 2011)

- Fase Persiapan

Fase persiapan memiliki beberapa tahap, yaitu:

- Menentukan aktivitas

Menentukan aktivitas merupakan tahap awal dalam fase persiapan. Pada tahap ini ditentukan aktivitas-aktivitas yang sesuai dengan proses. Pemilihan aktivitas dapat dibagi menjadi dua tahapan, yaitu pertama menentukan semua aktivitas yang dapat muncul dalam sebuah proses. Kedua yaitu menentukan aktivitas hanya dengan melihat *subset* aktivitas dari seluruh rangkaian aktivitas. Penentuan aktivitas yang kedua ini dapat dilakukan saat fase ekstraksi. Sumber informasi dalam menentukan aktivitas dapat dilihat pada Tabel 2.2.

Tabel 2.2 Sumber informasi untuk menentukan aktivitas

Sumber: (Piessens, 2011)

Standard	Corporate Environment
SAP Best Practices	Process Executor
SAP Easy Access Menu	SAP Consultant
Online Material	
Change Tables	

- Mengelompokkan kejadian-kejadian

Langkah ini bertujuan untuk mencari tahu aktivitas-aktivitas yang berhubungan dengan proses, berdasarkan tabel yang ada dan cara

mengeksekusi tabel tersebut. Terdapat beberapa sumber dari SAP yang dapat digunakan dalam memetakan kejadian-kejadian, yaitu:

1. *Literature Review*
2. Memantau perubahan tabel
3. Informasi secara *online*
4. Melihat hubungan tiap tabel
5. Melakukan *SQL trace* untuk melacak

- Memilih atribut

Memilih atribut merupakan langkah yang dilakukan dalam menentukan atribut-atribut yang akan digunakan untuk memodelkan proses. Umumnya, suatu catatan kejadian terdiri atas informasi mengenai *case identifier*, *activity name*, *executor*, dan *timestamp*. Informasi tersebut cukup untuk membentuk model proses. Akan tetapi, penambahan atribut yang mampu memberikan informasi lebih juga diperbolehkan.

- Fase Ekstraksi

Pada fase ini dilakukan untuk menentukan *outline* dari proses dan mengumpulkan seluruh informasi proses guna mengekstraksi catatan kejadian. Terdapat beberapa aktivitas dalam fase ekstraksi, yaitu:

- Memilih aktivitas yang akan diekstrak

Dalam fase ekstraksi, dari sekelompok atribut dipilih beberapa aktivitas yang dipertimbangkan untuk diekstraksi. Hal ini dilakukan untuk memastikan atribut yang dipilih telah memberikan gambaran yang jelas terkait dengan proses.

- Menentukan skenario

Langkah ini dilakukan dengan membuat kombinasi beberapa aktivitas yang mungkin terjadi.

- Mengonstruksi catatan kejadian

Langkah ini merupakan langkah terakhir dalam mengekstraksi catatan kejadian. Pada langkah ini

dibangun konstruksi catatan kejadian dengan meng-*query database* SAP. Catatan kejadian dapat diekstrak melalui prosedur berikut:

Contohnya terdapat sekumpulan aktivitas A

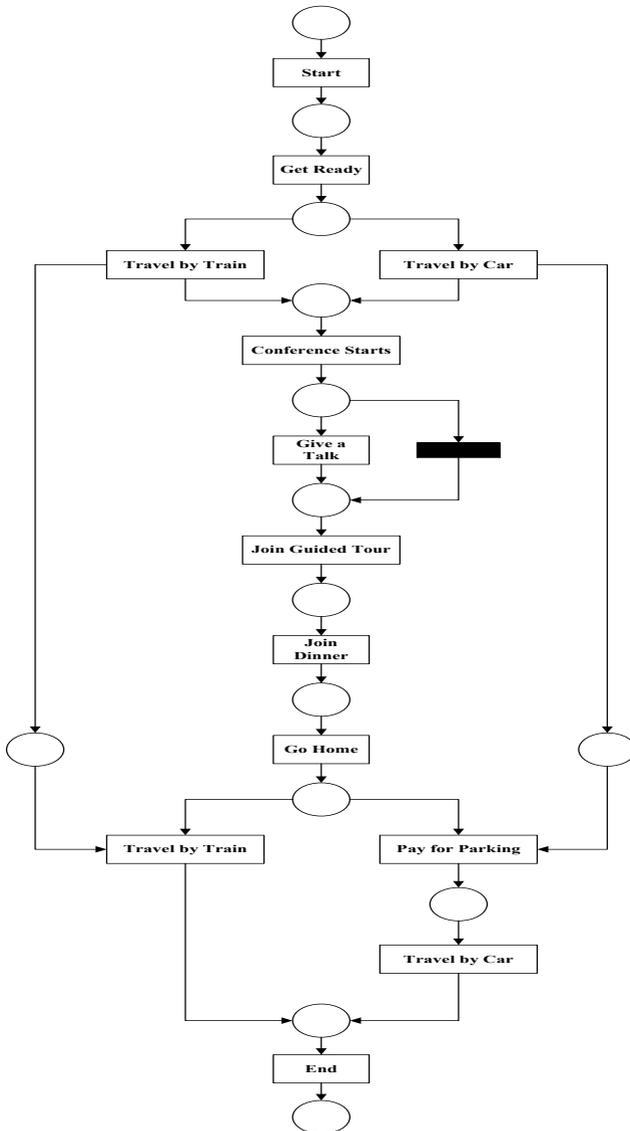
1. Menentukan skenario A
2. Untuk setiap aktivitas $a \in A$
3. Lalu ambil setiap aktivitas a dan tempatkan hasilnya pada R
4. Untuk setiap rekaman $r \in R$
5. Ekstrak atribut relevan att dari r
6. Simpan att ke dalam catatan kejadian

2.8. Algoritma Penggalan Proses

Berikut ini dijelaskan beberapa algoritma yang dapat digunakan dalam penggalan proses, yaitu Algoritma *Duplicate Genetic* dan Algoritma Genetika.

2.8.1. Algoritma *Duplicate Genetic*

Algoritma *Duplicate Genetic* merupakan perluasan dari Algoritma Genetika yang mampu menemukan adanya duplikasi *task* (Medeiros A. K., 2006). Gambaran algoritma ini ditunjukkan sebagai *causal matrices*, yang memiliki kelebihan dalam mendukung pemodelan semua aliran kontrol umum di model proses. Kelebihan algoritma *duplicate genetic* adalah mampu mendeteksi adanya *short-loop*, kejadian tersembunyi (*hidden activity*), *robust* terhadap *log* yang mengandung *noise*, *incompleteness*, serta mampu mendeteksi adanya pilihan tidak bebas (*non-free choice*) karena memanfaatkan penggalan model menurut frekuensi perilaku yang terjadi secara tepat. Selain itu, algoritma ini juga memiliki kemampuan dalam mendeteksi kejadian duplikat (*duplicate task*). Namun, dibalik kelebihan tersebut algoritma duplikat juga memiliki kelemahan yaitu algoritma ini memerlukan iterasi yang lebih banyak untuk mendapatkan solusi model yang lengkap dan presisi.



Gambar 2.7 Contoh model yang mengandung duplicate task
 Sumber: (Medeiros A. K., 2006)

2.9.1.1. Blok Utama Algoritma *Duplicate Genetic* (DGA)

Algoritma *Duplicate Genetic* memiliki tiga blok utama: gambaran internal individu, pengukuran *fitness*, dan operator genetika (Medeiros A. K., 2006).

1. Gambaran Internal Individu

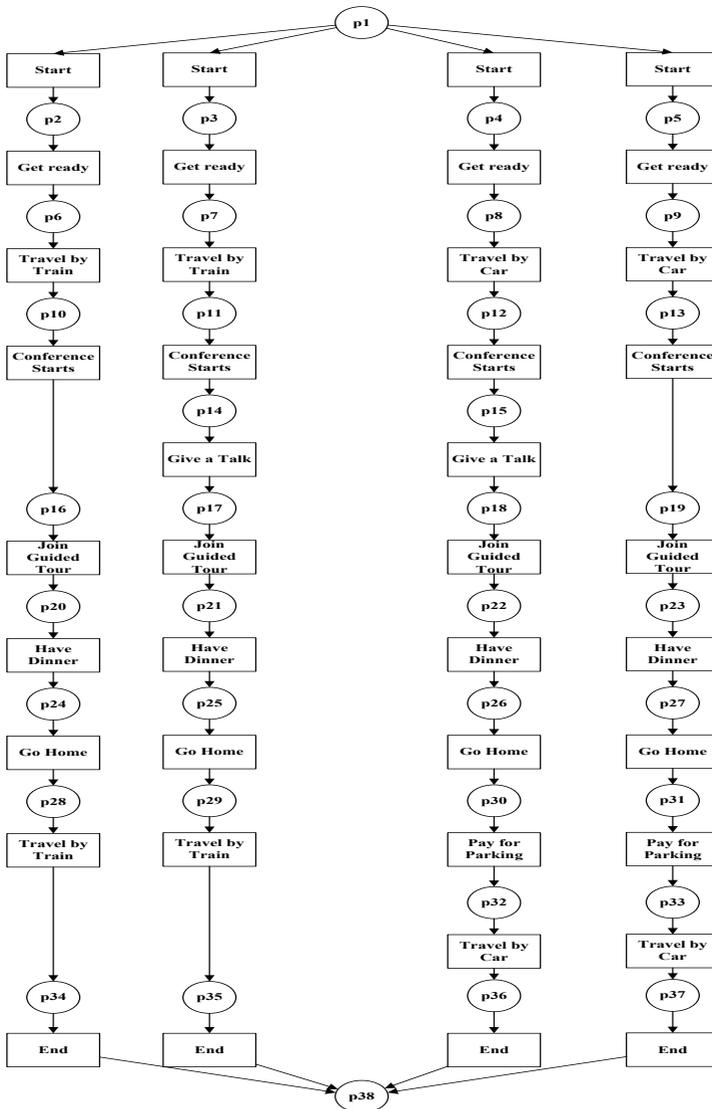
Kebutuhan utama dalam mendefinisikan gambaran internal adalah mampu menampilkan hubungan ketergantungan antara *tasks* yang ada dalam *log*. Dengan kata lain, model harus secara jelas menampilkan *task* apa saja yang mampu mengeksekusi *task* yang lain. DGA memerlukan gambaran internal yang mendukung duplikat. Gambaran internal yang mendukung duplikat terjadi ketika satu atau lebih aktivitas individu (model proses) dapat memiliki label atau nama aktivitas yang sama. Gambaran internal individu dapat pula disebut sebagai kausal matriks.

2. Pengukuran *Fitness*

Pengukuran *fitness* DGA berdasarkan pada dua pengembangan dari GA (*Genetic Algorithm*). Perpanjangan pertama adalah penambahan *folding requirement* pada pengukuran *fitness*. Kemudian pengembangan kedua adalah adaptasi dari *continuous semantics parser* pada GA untuk menangani duplikat selama perulangan *log* yang dilakukan oleh *fitness*.

a. Pengembangan Pengukuran *Fitness*

Penambahan *folding requirement* pada *fitness* DGA berfungsi untuk menghindari solusi yang terlalu spesifik. Solusi yang terlalu spesifik mengakibatkan pengukuran *fitness* memerlukan lebih banyak metrik untuk mempenalti spesialisasi yang berlebih saat berada di ruang pencarian.



Gambar 2.8 Model penggalan yang terlalu spesifik
 Sumber: (Medeiros A. K., 2006)

b. Pengembangan *Continuous Semantics Parser*

DGA menggunakan *continuous semantics parser* saat menghitung *fitness* individu. *Continuous semantics parser* mengandung arti bahwa proses *parsing* berlangsung bahkan ketika aktivitas yang di-*parsed* tidak aktif. Dengan kata lain, aktivitas yang seharusnya di-*parsed* selalu dieksekusi. Proses *parsing* harus menyelesaikan skenario, seperti skenario semua duplikat tidak aktif dan skenario beberapa duplikat aktif. Skenario pertama diselesaikan dengan secara acak mengeksekusi satu dari beberapa duplikat. Sedangkan skenario kedua diselesaikan berdasarkan elemen keluaran duplikat yang aktif. Duplikat yang dieksekusi untuk skenario kedua ini adalah duplikat yang muncul di posisi pertama mulai dari posisi *trace* saat ini berada hingga akhir dari *trace*. Namun jika ternyata terdapat beberapa duplikat yang memenuhi kondisi ini, maka pengekseskusion dilakukan terhadap satu diantara beberapa duplikat tersebut.

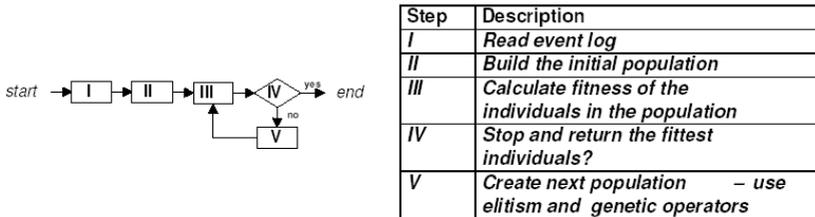
3. Operator Genetika

Operator genetika yang digunakan dalam algoritma duplikat adalah *crossover* dan *mutation*. *Crossover* bekerja dengan menukar hubungan sebab-akibat antara masukan/keluaran dari duplikat di dua individu. Pada DGA, *crossover* bekerja dengan memilih *label* yang terdapat dalam rentang fungsi *Label* setiap individu. Berdasarkan *label* ini, satu dari duplikat-duplikat yang terhubung dengan *label* ini dipilih dalam satu *parent*, dan duplikat lainnya yang juga berhubungan dengan *label* yang sama dipilih di *parent* yang lain. Untuk prosedur berikutnya sama dengan *crossover* pada GA. Operator *mutation* bekerja dengan menyisipkan, mengganti atau menghapus hubungan sebab-akibat dari masukan/keluaran pada aktivitas individu. Pada DGA sama dengan GA sebab

mutation bekerja secara independen dalam aktivitas yang duplikat ataupun tidak.

2.9.1.2. Pembentukan Model pada Algoritma *Duplicate Genetic*

Terdapat beberapa langkah dalam membentuk model pada algoritma *duplicate genetic* seperti yang ditunjukkan pada Gambar 2.9.



Gambar 2.9 Langkah utama DGA
Sumber: ProM 5.2 – Plugin reference

Penjelasan mengenai langkah-langkah tersebut adalah sebagai berikut (Medeiros A. K., 2006) :

1. Membaca catatan kejadian

Langkah pertama yang dilakukan adalah membaca catatan kejadian. Penjelasan mengenai catatan kejadian dapat dilihat pada tinjauan pustaka sub-bab 2.4.

2. Membangun populasi awal menggunakan *heuristics* untuk mengatur jumlah maksimum duplikat

Langkah kedua adalah membangun populasi dengan menggunakan *heuristic* atau hubungan ketergantungan antar aktivitas. Hal ini bertujuan untuk mengetahui seberapa kuat aktivitas *a* terhubung pada *label t* yang menyebabkan aktivitas lainnya *a'* terhubung dengan *label t*. Hubungan ketergantungan antar aktivitas dapat dihitung dengan rumus *Dependency Measure*. *Dependency Measure* digunakan untuk membangun hubungan sebab-akibat setiap individu. Rumus *Dependency Measure*, sebagai berikut (Medeiros A. K., 2006):

$$D(t_1, t_2) = \begin{cases} \frac{l2l(t_1, t_2) + l2l(t_2, t_1)}{l2l(t_1, t_2) + l2l(t_2, t_1) + 1} & \text{if } t_1 \neq t_2 \text{ and } l2l(t_1, t_2) > 0 \\ \frac{\text{follows}(t_1, t_2) - \text{follows}(t_2, t_1)}{\text{follows}(t_1, t_2) + \text{follows}(t_2, t_1) + 1} & \text{if } t_1 t_2 \text{ and } l2l(t_1, t_2) = 0 \\ \frac{l1l(t_1, t_2)}{l1l(t_1, t_2) + 1} & \text{if } t_1 = t_2 \end{cases} \quad (1)$$

Keterangan:

$l1l(t_1, t_2)$ = *short loop* = frekuensi *short loop* $t_1 t_1$ yang muncul pada catatan kejadian

$l2l(t_2, t_1)$ = frekuensi *length loop* $t_1 t_2 t_1$ yang muncul pada catatan kejadian

$\text{follows}(t_1, t_2)$ = frekuensi $t_1 t_2$ muncul dalam catatan kejadian

$\text{causal}(t_1, t_2)$ = menunjukkan nilai matriks kausal dalam baris t_1 dan kolom t_2 adalah 1

Setelah menghitung nilai hubungan antar aktivitas, maka dapat dibuat *causal matrix* yang nantinya akan berperan dalam membentuk sebuah individu. *Causal matrix* menggambarkan sebuah individu yang berupa model proses. Berdasarkan *causal matrix* tersebut kemudian dibentuk populasi awal. Proses terbentuknya individu secara berturut-turut ditunjukkan pada Gambar 2.10, Gambar 2.11, dan Gambar 2.12.

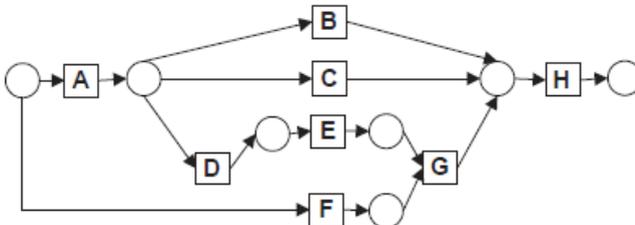
		INPUT								
		true	A	A	A	D	D	$E \wedge F$	$B \vee C \vee G$	
→	A	B	C	D	E	F	G	H		OUTPUT
A	0	1	1	1	0	0	0	0		$B \vee C \vee D$
B	0	0	0	0	0	0	0	1		H
C	0	0	0	0	0	0	0	1		H
D	0	0	0	0	1	1	0	0		$E \wedge F$
E	0	0	0	0	0	0	1	0		G
F	0	0	0	0	0	0	1	0		G
G	0	0	0	0	0	0	0	1		H
H	0	0	0	0	0	0	0	0		true

Gambar 2.10 Contoh causal matrix

Sumber: (Medeiros, Weijters, & Aalst, 2005)

ACTIVITY	INPUT	OUTPUT
A	{}	{{B, C, D}}
B	{{A}}	{{H}}
C	{{A}}	{{H}}
D	{{A}}	{{E, F}}
E	{{D}}	{{G}}
F	{{D}}	{{G}}
G	{{E, F}}	{{H}}
H	{{B, C, G}}	{}

Gambar 2.11 Individu yang dihasilkan causal matrix
 Sumber: (Medeiros, Weijters, & Aalst, 2005)



Gambar 2.12 Individu yang berupa model proses
 Sumber: (Medeiros, Weijters, & Aalst, 2005)

3. Menghitung *fitness* individu

Langkah selanjutnya adalah menghitung *fitness* seperti cara berikut:

a. *Fitness Complete*

Fitness Complete digunakan untuk mengukur kelengkapan individu (model) yang dibentuk. Model proses dinyatakan lengkap ketika dapat membentuk model berdasarkan semua *trace* yang ada dalam *log*.

Rumus *Fitness Complete* sebagai berikut (Medeiros A. K., 2006):

$$PF_{complete}(L, CM) = \frac{allParsedActivities(L, CM) - punishment}{numActivitiesLog(L)} \quad (2)$$

Dimana

$$punishment = \frac{\frac{allMissingTokens(L, CM)}{numTracesLog(L) - numTracesMissingTokens(L, CM) + 1} + \frac{allExtraTokensLeftBehind(L, CM)}{numTracesLog(L) - numTracesExtraTokensLeftBehind(L, CM) + 1}}{numTracesLog(L) - numTracesExtraTokensLeftBehind(L, CM) + 1} \quad (3)$$

Keterangan

$allParsedActivities(L, CM)$	= menunjukkan jumlah tugas yang ada dalam event log yang dapat diuraikan tanpa masalah dengan matriks kausal (CM)
$numActivitiesLog(L)$	= menunjukkan jumlah tugas pada log
$allMissingTokens(L, CM)$	= menunjukkan jumlah token yang hilang dalam trace
$AllExtraTokensLeftBehind(L, CM)$	= menunjukkan jumlah token yang tidak dikonsumsi setelah parsing berhenti + jumlah token dari tempat akhir - 1
$numTracesLog(L)$	= Menunjukkan jumlah trace yang ada pada log

$NumTracesMissingTokens(L, CM)$ = menunjukkan jumlah trace yang kehilangan token

$NumTracesExtraTokensLeftBehind(L, CM)$ = Menunjukkan jumlah trace dimana terdapat token yang ditinggal saat parsing.

b. *Fitness Precise*

Fitness Precise digunakan untuk menghitung ketepatan individu (model) yang dibentuk. Model proses dinyatakan tepat ketika tidak membentuk model diluar perilaku yang ada pada log. Rumus *Fitness - PF_{precise}* sebagai berikut (Medeiros A. K., 2006):

$$PF_{precise}(L, CM, CM[]) = \frac{allEnabledActivites(L, CM)}{\max(allEnabledActivities(L, CM[]))} \quad (4)$$

Keterangan:

$allEnabledActivities(L, CM)$ = menunjukan jumlah aktivitas yang diaktifkan selama proses parsing pada log yang dilakukan oleh matriks kausal

$AllEnabledActivities(L, CM[])$ = mengembalikan nilai maksimal dari jumlah aktivitas yang diaktifkan oleh model proses selama proses parsing pada log

c. Kombinasi *Fitness Complete* dan *Precise*

Model yang memenuhi syarat kelengkapan (*completeness*) belum tentu memenuhi syarat ketepatan (*preciseness*). Oleh sebab itu perlu mengkombinasikan kedua syarat tersebut untuk mendapat model proses yang *completeness* dan *precesiness*. Untuk memastikan model proses sudah memnuhi kedua persyaratan tersebut bisa diukur dengan menggunakan rumus *Fitness* -F berikut (Medeiros A. K., 2006):

$$F(L, CM, CM[]) = PF_{complete}(L, CM) - \kappa * PF_{precise}(L, CM, CM[]) \quad (5)$$

Keterangan:

κ = bilangan real di antara $0 > x \leq 1$, k merupakan beratnya (*weighs*) hukuman untuk perilaku diluar logl

4. Hentikan algoritma jika memenuhi beberapa hal berikut:
 - Menemukan individu dengan *fitness* = 1
 - Menghitung generasi n, dimana n adalah jumlah maksimum generasi yang diperbolehkan
 - Individu *fittest* tidak berubah selama beberapa generasi n/2 berturut-turut

Jika algoritma berhenti, maka kembali ke populasi awal. Jika tidak maka akan berlanjut ke langkah selanjutnya yaitu membentuk generasi baru dengan operasi genetika.
5. Langkah terakhir adalah membentuk generasi selanjutnya menggunakan elitism dan operasi genetika. Elitism mengatur tingkat persentase kekuatan individu pada generasi saat yang akan disalin pada generasi berikutnya. Misalnya, tingkat elitism sebesar 0.02 atau 2%. Angka tersebut menunjukkan bahwa sebesar 2% individu terbaik dalam populasi akan disalin ke populasi berikutnya. Operasi genetika yang digunakan untuk membentuk individu baru dalam generasi selanjutnya adalah mutasi dan *crossover*.

2.8.2. Algoritma Genetika

Algoritma genetika adalah sebuah teknik pencarian yang meniru proses evolusi (Medeiros, Weijters, & Aalst, Using Genetic Algorithms to Mine Process Models: Representation, Operators and Results, 2005). Kelebihan algoritma genetika adalah mampu mendeteksi adanya *short-loop*, kejadian tersembunyi (*hidden activity*), *robust* terhadap *log* yang mengandung *noise*, *incompleteness*, serta mampu mendeteksi adanya pilihan tidak bebas (*non-free choice*). Namun algoritma ini kurang stabil dengan *AND split/join* terutama jika terjadi model percabangan paralel yang panjang dan tak mampu mendeteksi adanya kejadian duplikat (*duplicate task*).

Sama halnya dengan algoritma *duplicate genetic*, terdapat 3 fokus utama algoritma genetika, yaitu (Medeiros, Weijters, & Aalst, Using Genetic Algorithms to Mine Process Models: Representation, Operators and Results, 2005):

1. Gambaran internal

Gambaran internal GA menampilkan hubungan ketergantungan antara *task* dan *semantic* dari *split/join task*. Pendefinisian gambaran internal disebut sebagai kausal matriks.

2. Pengukuran *fitness*

Pengukuran *fitness* mengevaluasi kualitas model proses atau individu dalam ruang pencarian pada catatan kejadian. Algoritma genetika mencari individu-individu yang memiliki nilai *fitness* maksimal. Pengukuran *fitness* memastikan bahwa individu yang memiliki *fitness* maksimal dapat mengurangi seluruh *instance* proses (*traces*) dalam *log*, sehingga tidak hanya perilaku yang dapat diturunkan dari *trace* tersebut.

3. Operator genetika

Operator genetika yang digunakan dalam algoritma genetika adalah *crossover* dan *mutation*. Konsep utama dari operator tersebut adalah hubungan kausalitas (yaitu dependensi) antara *task* materi genetik yang akan dimanipulasi. *Crossover* pada GA bekerja dengan memilih aktivitas pada dua *parent* dan

secara acak menukar beberapa bagian dari kumpulan masukan atau keluaran aktivitas. Aktivitas disini mengacu pada tugas-tugas yang ada dalam gambaran internal sebuah individu.

2.8.3. Perbedaan Algoritma *Duplicate Genetic* dan Algoritma Genetika

Secara umum, proses antara DGA dan GA adalah sama. Namun terdapat beberapa proses yang membedakannya, yaitu (Medeiros A. K., 2006):

1. Terdapat penentuan jumlah duplikat tiap *task* yang akan dimiliki masing-masing individu. Penentuan ini dilakukan saat menentukan populasi awal. Fungsi penentuan ini adalah untuk menentukan ruang pencarian DGA.
2. Pada langkah menghitung *fitness* individu, terdapat penambahan pengukuran *fitness* untuk DGA, yaitu $PF_{folding}$. DGA akan membentuk model dengan *duplicate task* dimana *duplicate task* tersebut tidak memiliki elemen masukan yang digunakan bersama. Hal itu disebut sebagai batasan DGA. Penambahan *folding requirement* pada *fitness* DGA berfungsi untuk menghindari solusi yang terlalu spesifik, dengan kata lain *folding requirement* akan memberikan “hukuman” pada model jika tidak memenuhi batasan yang ada. Semakin banyak pelanggaran yang dilakukan individu terhadap batasan duplikasi, maka “hukuman” yang diberikan juga akan semakin berat. Rumus *folding requirement* sebagai berikut (Medeiros A. K., 2006):

$$PF_{folding}(CM, CM[]) = \frac{DuplicateSharingElements(CM)}{mac(DuplicateSharingElements(CM[]))} \quad (6)$$

Dimana,

- $DuplicateSharingElements(CM)$ menunjukkan jumlah *tuples* yang berbeda dalam *causal matrix* yang dipetakan dalam label yang sama
- $DuplicateSharingElements(CM[])$ menunjukkan nilai $DuplicateSharingElements(CM)$ dari masing-masing individu dalam sebuah populasi.

Sehingga nilai F atau *fitness* untuk model yang dihasilkan dengan DGA dihitung dengan rumus (Medeiros A. K., 2006):

$$F_{DGA}(L, CM, CM[]) = F(L, CM, CM[]) - \gamma * PF_{folding}(CM, CM[]) \quad (7)$$

Dimana γ adalah beratnya (*weight*) “hukuman” yang diberikan kepada individu yang memiliki *duplicate task* diluar batasan.

Pada langkah 4, algoritma genetika akan melakukan pembersihan individu dari *task* yang tidak digunakan seperti *invisible task*. Untuk DGA, pembersihan individu tak hanya dilakukan untuk *invisible task*. Penghapusan juga dilakukan terhadap duplikasi bayangan sebelum individu tersebut dikembalikan ke populasi awal.

2.9. Pengukuran Performa Model

Pengukuran performa model dapat dilakukan dengan mengukur dari tiga dimensi yaitu *fitness*, presisi, dan struktural.

2.9.1. *Fitness*

Fitness mengukur seberapa besar perilaku (*log* peristiwa) sesuai dengan model proses. Pengukuran *fitness* dapat dilakukan dengan menggunakan rumus seperti berikut (Rozinat & Aalst, 2009):

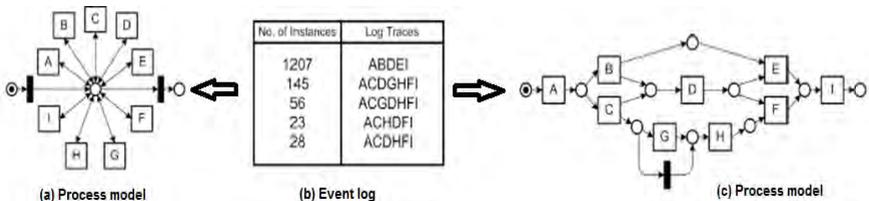
$$f = \frac{1}{2} \left(1 - \frac{\sum_{i=1}^k n_i m_i}{\sum_{i=1}^k n_i c_i} \right) + \frac{1}{2} \left(1 - \frac{\sum_{i=1}^k n_i r_i}{\sum_{i=1}^k n_i p_i} \right) \quad (8)$$

Keterangan:

- k = jumlah *trace* yang berbeda dari *log*, untuk setiap *log trace* I
($1 \leq i \leq k$)
 - n_i = Jumlah proses *instance trace* i
 - m_i = jumlah *missing* token
 - r_i = Jumlah token tersisa
 - c_i = Jumlah token yang dikonsumsi
 - p_i = Jumlah token yang diproduksi
- Range ukuran *fitness* yaitu $0 \leq fitness \leq 1$

2.9.2. Presisi

Presisi mengukur ketepatan model proses yang dihasilkan dengan melihat dari berapa banyak *trace* yang mungkin terbentuk. Pengukuran ini tidak berasal dari *log*. Suatu model dikatakan presisi apabila tidak terdapat perilaku yang berlebihan. Model yang tidak presisi adalah model yang “*underfitting*”. *Underfitting* merupakan masalah yang mana model terlalu mengeneralisasi perilaku di dalam *log* (contohnya, dalam model terdapat perilaku yang sangat berbeda dari apa yang dilihat pada *log*). Hal ini mengakibatkan model menjadi kurang informatif karena tidak lagi menggambarkan proses yang sebenarnya.



Gambar 2.13 Model proses dengan presisi yang berbeda

Pada Gambar 2.13 (a) menunjukkan model proses yang tidak presisi. Model tersebut terlalu general dan terdapat perilaku yang berlebihan, yaitu perilaku yang sangat berbeda dari apa yang ada pada catatan kejadian. Gambar 2.13 (c) menunjukkan model proses yang presisi dengan tidak adanya perilaku yang berlebihan.

Pengukuran presisi dapat dilakukan dengan menggunakan rumus *Advanced Behavioral Appropriateness* seperti berikut (Rozinat & Aalst, 2009):

$$a'_B = \left(\frac{S_F^l \cap S_F^m}{2 \cdot |S_F^m|} \right) + \left(\frac{S_P^l \cap S_P^m}{2 \cdot |S_P^m|} \right) \quad (9)$$

Keterangan:

S_F^m = hubungan “*Sometimes follows*” untuk proses model

S_P^m = hubungan “*Sometimes precedes*” untuk proses model

S_F^l = hubungan “*Sometimes follows*” untuk catatan kejadian

S_P^l = hubungan “*Sometimes precedes*” untuk catatan kejadian

Range Advanced Behavioral Appropriateness yaitu $0 \leq a'_B \leq 1$

Penjelasan relasi “*Follow*” dan “*Precedes*” sebagai berikut:

- *Follows relations* : Dua aktivitas (x,y) yang “*Always Follows*”, “*Never Follows*”, atau “*Sometimes Follows*” berhubungan dalam kasus jika x dieksekusi minimal sekali, y dieksekusi *always*, *never*, atau *sometimes* secara cepat dan berturut-turut.
- *Precedes relations* : Dua aktivitas (x,y) yang “*Always Precedes*”, “*Never Precedes*”, atau “*Sometimes Precedes*” berhubungan dalam kasus jika y dieksekusi minimal sekali, x dieksekusi beberapa waktu sebelumnya secara berturut-turut.

2.9.3. Struktural

Pengukuran struktural menunjukkan kemampuan model dalam menangani proses *XOR* dan *AND*. *XOR* menunjukkan bahwa dalam model terdapat aktivitas bercabang yang dalam percabangannya dapat memilih salah satu aktivitas yang akan dikerjakan. *AND* menunjukkan bahwa dalam model terdapat aktivitas yang dapat dikerjakan bersama-sama. Perhitungan struktural dapat dilakukan dengan rumus *Advanced Structural Appropriateness* seperti berikut (Rozinat & Aalst, 2009):

$$a'_S = \frac{|T| - (|T_{DA}| + |T_{IR}|)}{|T|} \quad (10)$$

Keterangan:

T = kumpulan transisi dari model Petri Net

T_{DA} = kumpulan *alternative duplicate task*

T_{IR} = kumpulan *redundant invisible task*

Range Advanced Behavioral Appropriateness yaitu $0 \leq a'_S \leq 1$

Jika nilai semakin mendekati 1, maka struktur dikatakan baik sebab jumlah *duplicate task* dan *redundant invisible task* yang ada dalam model proses semakin sedikit.

Alternative duplicate task adalah tugas rangkap yang kemunculannya tidak pernah terjadi bersama-sama dalam satu urutan eksekusi, sedangkan *redundant invisible task* adalah tugas yang tidak tampak dan dapat dihapus dari model tanpa mengubah perilaku. Kedua hal tersebut perlu dihindari sebab mampu membuat struktur model proses berkembang dan mengurangi kejelasan perilaku.

2.10. Enterprise Resource Planning (ERP)

ERP merupakan suatu sistem informasi perusahaan yang memungkinkan perusahaan mengotomatiskan dan mengintegrasikan berbagai proses bisnis utamanya (Hall, 2008). Menurut O'Brien (2005), sistem *ERP* mampu menghasilkan manfaat bisnis yang signifikan bagi perusahaan. Manfaat tersebut, yaitu:

- Kualitas dan Efisiensi
ERP menciptakan kerangka kerja untuk mengintegrasikan dan meningkatkan proses bisnis internal perusahaan atau organisasi sehingga menghasilkan peningkatan signifikan kualitas serta efisiensi layanan pelanggan, produksi, dan distribusi.
- Penurunan Biaya
Banyak perusahaan melaporkan penurunan signifikan dalam biaya pemrosesan transaksi dan *hardware*, *software*, serta karyawan pendukung TI, jika dibandingkan dengan sistem warisan yang tidak terintegrasi yang digantikan oleh sistem *ERP* baru mereka.
- Pendukung Keputusan
ERP menyediakan informasi mengenai kinerja bisnis yang sangat penting secara cepat untuk para manajer agar dapat secara signifikan meningkatkan kemampuan dalam mengambil keputusan.
- Kelincahan Perusahaan
Pengimplementasian *ERP* menghasilkan struktur organisasi, tanggung jawab manajerial, dan peran kerja yang lebih fleksibel, dan karenanya menghasilkan organisasi serta tenaga kerja yang lebih lincah.

ERP memiliki beberapa modul yang dapat dipilih oleh perusahaan sesuai dengan kebutuhan. Modul-modul yang dipilih oleh perusahaan kemudian akan disesuaikan dengan proses bisnis yang ada.

Integrasi dalam *ERP* pada dasarnya menghubungkan modul-modul yang ada sehingga mampu mendukung proses bisnis perusahaan.

2.11. Perangkat Lunak SAP

SAP (Systems, Applications, and Products in Data Processing) adalah salah satu aplikasi *ERP* yang banyak digunakan di dunia saat ini. *SAP* pada umumnya diinstal di seluruh lapisan perusahaan dan di beberapa komputer pengguna guna memberikan gambaran yang jelas dan lengkap terkait keseluruhan proses (Piessens, 2011).

Terdapat beberapa modul yang ada dalam *SAP*, mulai dari modul keuangan (*Financial/FI*), modul pengelolaan material (*Material Management/MM*), modul pengelolaan pelanggan (*Customer Relationship Management/CRM*), modul pengelolaan penjualan (*Sales and Distribution/SD*), modul produksi (*Production and Planning/PP*), hingga modul pengelolaan gudang (*Warehouse Management/WM*).

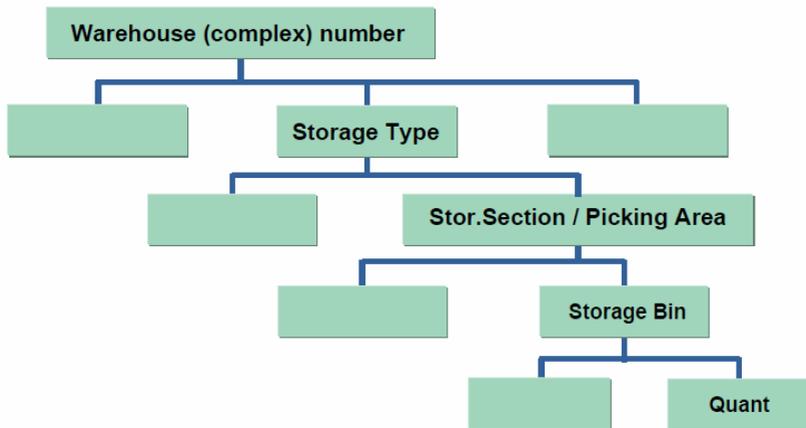
2.11.1. Modul Warehouse Management

Modul *Warehouse Management (WM)* adalah modul yang berada dibawah modul *materials management (MM)*. Fungsi dasar dari *WM*, yaitu:

- Manajemen persediaan untuk tingkat *storage bin*
- Memetakan dan mengontrol seluruh pergerakan barang
- Memantau proses dari pergerakan barang
- Menghubungkan dengan system eksternal menggunakan

interface

Struktur *warehouse* dapat dilihat pada Gambar 2.14 seperti dibawah ini:



Gambar 2.14 Struktur Warehouse
 Sumber: (AG, 2001)

Storage type adalah tempat atau ruangan atau zona *warehouse* yang didefinisikan dalam *Warehouse Management*. Satu *storage type* mengandung satu atau lebih *storage bin*. *Storage section* adalah organisasional subdivisi dari *storage* yang mengelompok bersama dengan *storage bin* dengan fitur yang sama. *Storage bin* adalah tempat atau ruangan yang berada di dalam *storage type*, dan juga merupakan unit terkecil dalam *warehouse*. *Quant* adalah stok tiap material dengan fitur yang sama dalam satu *storage bin*.

2.12. Penelitian Sebelumnya

Pengerjaan tugas akhir ini dilakukan dengan mengacu beberapa penelitian yang dilakukan sebelumnya, seperti:

1. *Event Log Extraction from SAP ECC 6.0 D.A.M* oleh Piessens tahun 2011. Penelitian ini menjelaskan cara mengekstraksi catatan kejadian dari SAP ECC 6.0.
2. *Conformance Checking of Processes based on Monitoring Real Behaviour* oleh A.K. Alves de Medeiros, A.J.M.M. Weijters, dan W.M.P. van der Aalst tahun 2009.

Penelitian ini menjelaskan cara mengukur atau mengevaluasi model proses yang dihasilkan sebuah algoritma.

3. Tugas akhir “Pemodelan dan analisis *bottleneck* proses bisnis perencanaan produksi di PT.XYZ pada SAP dengan algoritma genetika” oleh Rivia Atmajaningtyas Utami tahun 2013.
Tugas akhir ini membahas mengenai pendeteksian *bottleneck* yang terdapat dalam proses aktual perencanaan produksi dengan menggunakan algoritma genetika. Kemudian dari model yang terbentuk, dilakukan analisis perbandingan model proses dengan alur kerja perencanaan produksi. Analisis ini dilakukan untuk mengetahui perbedaan model proses aktual dengan alur kerja yang didefinisikan. Selain itu juga dilakukan analisis *bottleneck*. Analisis ini dilakukan untuk mengetahui letak dan penyebab terjadinya *bottleneck* dalam proses perencanaan produksi. Pendeteksian *bottleneck* dilakukan menghitung selisih antara waktu mulai aktivitas dengan waktu selesainya aktivitas tersebut. Pendeteksian ini dilakukan dengan bantuan ProM dan data catatan kejadian yang digunakan berasal dari SAP.
4. Tesis “Analisis kinerja algoritma penggalian proses untuk pemodelan proses bisnis perencanaan produksi dan pengadaan material pada PT.XYZ dengan kriteria *control-flow*” oleh Rachmadita Andreswari tahun 2013.
Tesis ini membahas mengenai model proses yang dihasilkan dari dua proses bisnis, yaitu perencanaan produksi dan pengadaan material. Analisis dilakukan dengan membandingkan model proses menggunakan empat algoritma, yaitu algoritma *alpha++*, algoritma *duplicate task genetic*, algoritma genetika, dan algoritma *heuristic miner*. Perbandingan model proses yang dihasilkan dilihat dari faktor proses bisnis, penggunaan algoritma, pengukuran kinerja, karakteristik model, dan waktu. Data catatan kejadian yang digunakan berasal dari SAP.
5. Tesis “*Genetic Process Mining*” oleh Ana Karla Alves de Medeiros tahun 2006.
Tesis ini membahas mengenai kemungkinan mengembangkan algoritma *control-flow* penggalian proses yang mampu menemukan semua struktur dasar *control-flow* yaitu, *sequences*,

parallelism, choices, loops, non-free-choice, invisible task, dan duplicate task serta *robust* terhadap *noisy logs*. Tesis ini menggunakan algoritma genetika untuk melakukan penggalian proses.

Halaman ini sengaja dikosongkan.

BAB III

METODE Pengerjaan Tugas Akhir

Pengerjaan tugas akhir ini dilakukan dalam empat tahap, yaitu persiapan, penggalian proses, pengukuran model, dan analisis hasil. Ringkasan metodologi pengerjaan tugas akhir ditunjukkan pada Gambar 3.1.

Berikut ini adalah penjelasan detail untuk setiap fase dan aktivitas pada metodologi pengerjaan Tugas Akhir:

3.1. Persiapan

Fase ini merupakan fase awal sebelum dilakukan penggalian proses. Fase ini adalah fase yang penting sebab kebenaran pada hasil penggalian proses bergantung pada fase persiapan. Terdapat tiga tahapan dalam fase persiapan, yaitu studi literatur, pengumpulan data, dan standardisasi catatan kejadian.

3.1.1. Studi Literatur

Pada tahap ini dilakukan kajian terhadap beberapa referensi terkait dengan materi penggalian proses, model proses, algoritma *duplicate genetic*, pengukuran model, ekstraksi catatan kejadian, ProM, Petri Net. Sumber referensi yang digunakan dalam studi literatur ini seperti *paper*, buku, artikel di internet, untuk kemudian dijelaskan kembali pada bab Tinjauan Pustaka.

3.1.2. Pengumpulan Data

Tahap ini bertujuan untuk mengumpulkan data-data yang diperlukan terkait pembentukan model. Terdapat dua aktivitas dalam pengumpulan data, yaitu:

a. Wawancara

Wawancara dilakukan terhadap *main planner* dan koordinator tiap tahap proses bisnis divisi PDC PT.XYZ. Hasil dari wawancara ini berupa informasi proses bisnis PDC dan kondisi yang ada dalam divisi PDC. Selain itu wawancara juga dilakukan dengan manajer *SAP* untuk memastikan apakah

memungkinkan dilakukan ekstraksi langsung dari *database SAP*.

b. Ekstraksi Data

Ekstraksi data merupakan aktivitas yang dilakukan untuk mengambil data catatan kejadian dari *SAP*. Hasil dari aktivitas ini adalah data mentah catatan kejadian yang berbentuk CSV. Langkah ekstraksi data sudah dijelaskan pada tinjauan pustaka sub-bab 2.8.

3.1.3. Standardisasi Catatan Kejadian

Tahap ini bertujuan untuk mengubah data mentah catatan kejadian (format CSV) menjadi data yang siap diolah ke ProM. Dalam melakukan standardisasi catatan kejadian harus memperhatikan terlebih dahulu apakah data mentah yang didapatkan sudah memiliki struktur catatan kejadian atau belum. Jika belum, maka data mentah harus disusun ke dalam bentuk catatan kejadian dengan atribut yang telah dijelaskan pada tinjauan pustaka sub-bab 2.5. Apabila sudah berstruktur catatan kejadian, maka data tersebut dapat langsung dikonversikan ke dalam format MXML. Hasil dari standardisasi catatan kejadian adalah file catatan kejadian dengan format MXML.

3.2. Penggalan Proses

Fase ini merupakan fase utama dalam pembuatan model proses

3.2.1. Pembuatan Model Proses

Langkah yang dilakukan setelah mendapatkan catatan kejadian dalam format MXML adalah membangun model proses bisnis dengan menggunakan algoritma *duplicate genetic*. Catatan kejadian (format MXML) dimasukkan ke dalam ProM kemudian memilih algoritma *duplicate genetic* dan mengatur parameter yang akan digunakan dalam pembuatan model proses. Keluaran dari langkah ini adalah model proses yang berbentuk Petri Net.

3.3. Pengukuran Model

Tahap ini dilakukan setelah model proses dari ProM terbentuk. Tujuan pengukuran model adalah untuk mengetahui kebenaran dari

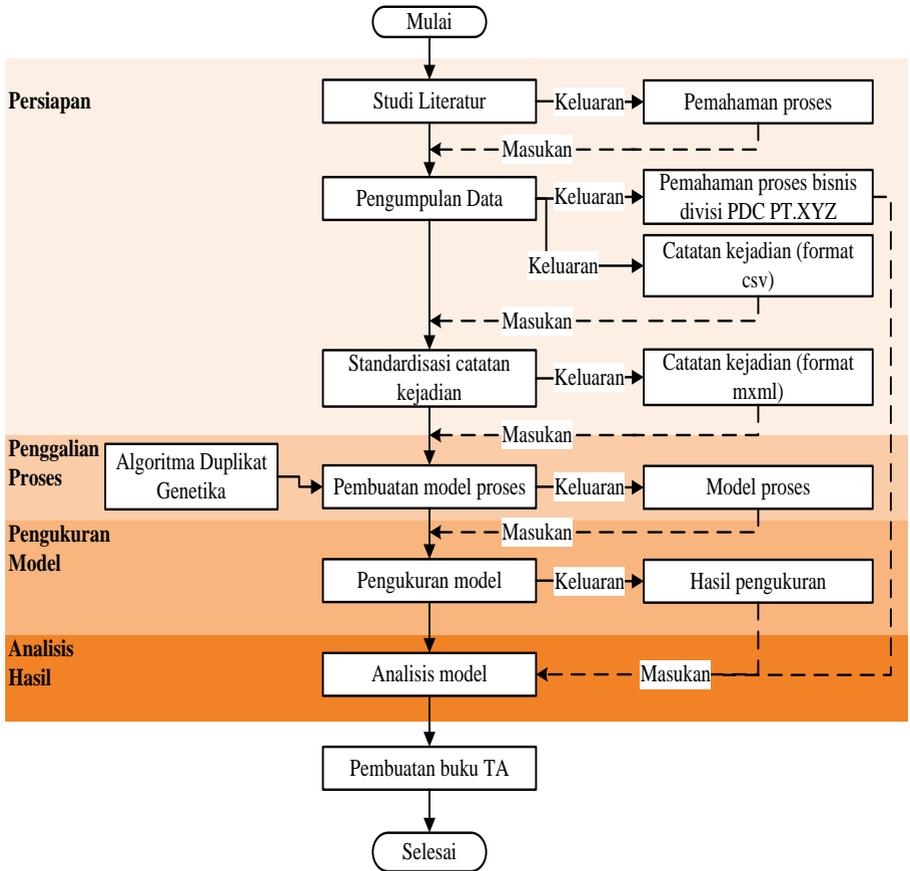
model yang dihasilkan. Pengukuran dilakukan dengan menggunakan tiga dimensi, yaitu *fitness*, presisi, dan struktural seperti yang dijelaskan pada sub-bab 2.10.

3.4. Analisis Hasil

Tahap ini dilakukan untuk menganalisis keluaran dari model proses berdasarkan algoritma *duplicate genetic*. Hasil dari analisis ini kemudian dibandingkan dengan proses bisnis yang telah didefinisikan perusahaan dan juga akan diidentifikasi pengaruh *volume* dan variasi terhadap *lead time* penyelesaian pengepakan di PDC.

3.5. Pembuatan buku Tugas Akhir

Membuat dokumentasi keseluruhan proses sesuai dengan permasalahan yang diangkat.



Gambar 3.1 Metodologi pengerjaan TA

BAB IV

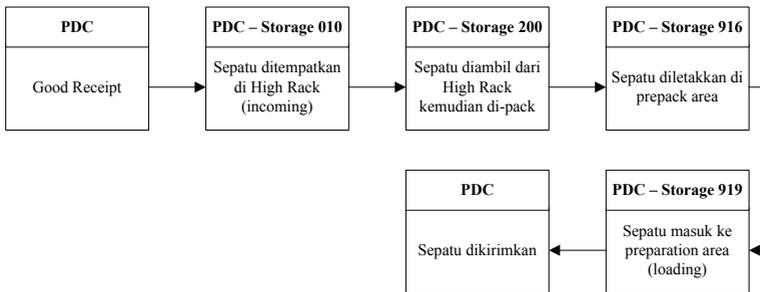
PEMODELAN PROSES BISNIS

4.1. Studi Kasus

PT. XYZ merupakan anak perusahaan PT. XYZ Internasional yang bergerak dibidang manufaktur. PT. XYZ yang memiliki spesialisasi dalam memproduksi *shoe* dan *upper shoe*, yaitu bagian atas sepatu. Dalam menjalankan proses bisnis PT. XYZ didukung oleh system informasi terintegrasi, yaitu SAP. SAP ini mampu mengintegrasikan proses bisnis tiap divisi yang ada dalam PT. XYZ.

4.1.1. Divisi *Production Distribution Center (PDC)*

Divisi PDC adalah divisi yang bertugas dalam penyimpanan *finished good*. PDC didirikan pada Mei 2005 dan bertempat di Hall 2. Latar belakang pendirian PDC yaitu untuk memfasilitasi pertumbuhan jumlah sepatu yang dihasilkan dari mesin *injection*. Proses utama PDC adalah *stock-pick-pack* dan *loading for export* (termasuk proses perijinan Bea Cukai). PDC dibangun diatas lahan seluas 6.006 m², dengan jumlah pegawai sebanyak 117 orang dan terbagi atas 3 *shift*. PDC mampu *packing* sebanyak 180.000 pasang sepatu per minggu. Barang yang disimpan dalam PDC didasarkan pada musim. Terdapat 4 musim yaitu, *spring*, *summer*, *autumn*, dan *winter*. Musim ini kemudian dikelompokkan lagi menjadi 2, yaitu *spring-summer* (SS) dan *autumn-winter* (AW). Adapun kapasitas penyimpanan PDC sebesar 750.000 pasang sepatu untuk SS dan 550.000 pasang sepatu untuk AW. Gambar 4.1 menunjukkan proses bisnis secara umum di PDC.



Gambar 4.1 Proses bisnis PDC secara umum

Dari Gambar 4.1 tersebut, proses bisnis didalam PDC kemudian dapat dijabarkan lagi menjadi 4 tahapan, yaitu *incoming* PDC, *packing plan creation*, *packing* PDC, dan *outgoing* PDC. *Incoming* merupakan proses awal yang dilakukan dalam PDC. Pada proses bisnis ini, produk atau sepatu yang datang akan diperiksa terlebih dahulu kelengkapannya. Apabila lengkap dan sesuai, maka sepatu akan diletakkan di *high rack*.

Proses berikutnya setelah *incoming* adalah pembuatan rencana *packing* (*packing plan creation*). PT.XYZ merupakan perusahaan yang bertipe MTS/MTO. Tipe perusahaan yang seperti ini menyebabkan PT.XYZ dalam membuat rencana pengepakan berdasarkan pada pesanan atau *delivery order* yang ada. Pembuatan rencana pengepakan dilakukan dengan melihat terlebih dahulu jumlah stok sepatu atau barang yang tersedia dalam gudang PDC. Aktivitas mencocokkan antara jumlah pesanan dengan ketersediaan stok sepatu dalam PDC dilakukan oleh sistem secara otomatis yang dinamakan *SAP Allocation Running* (*SAP A-Run*). Setelah sistem mengonfirmasi tersedianya jumlah stok sepatu sesuai dengan jumlah pesanan yang ada, maka perencanaan pengepakan dapat dilakukan. Saat aktivitas ini dilakukan, maka akan terbentuk pula sebuah catatan pengiriman (*delivery note*). *Delivery note* ini berisi perintah-perintah pengambilan barang dari rak, pengepakan, hingga pengiriman barang. Rencana pengepakan memuat beberapa

pengelompokan sepatu yang didasarkan pada beberapa kriteria tertentu untuk mempermudah proses pengiriman kepada pelanggan.

Proses berikutnya adalah pengepakan. Pengepakan dilakukan dengan sistem *picking wave*. Satu *picking wave* berisi satu rencana pengepakan. *Packing PDC* merupakan proses pengambilan sepatu untuk di-*pack*. Saat *packing*, sepatu-sepatu dengan berbagai macam artikel yang telah dikelompokkan ke dalam sebuah *delivery* kemudian akan dikelompokkan lagi ke dalam sebuah *M-Group*. Proses pembuatan *M-Group*, pada awalnya kantor pusat yang berada di Denmark akan mengirimkan *quantity target* sejumlah *pairs* tertentu ke PDC. Target tersebut didasarkan pada *stock development* yang mana *stock development* ini bisa didasarkan pada peramalan stok dan pesanan pelanggan.

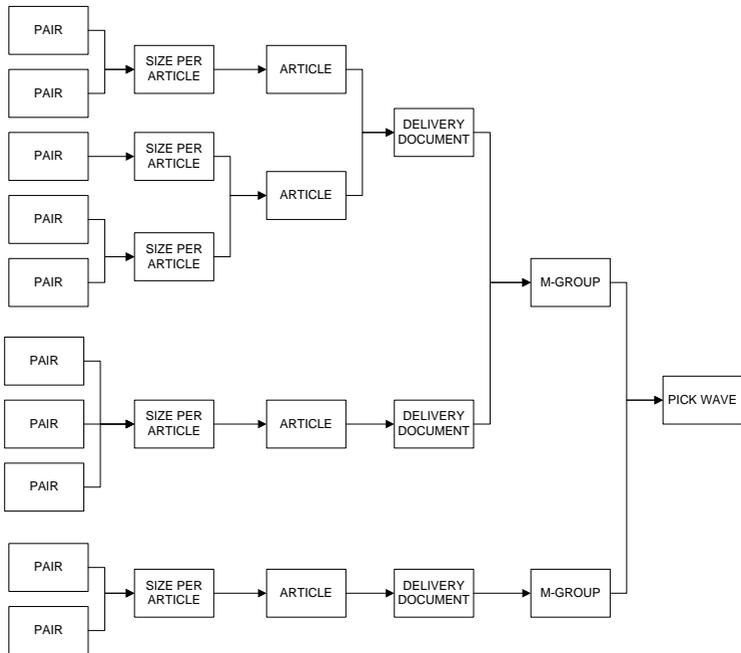
Pihak PDC kemudian akan memecah target *pairs* tersebut ke dalam beberapa puluh ribu *pairs*, misalnya dari 170.000 *pairs* yang ditargetkan oleh kantor pusat kemudian oleh PDC dipecah menjadi 3 target, yaitu 50.000 *pairs*, 70.000 *pairs*, dan 50.000 *pairs*. Pemecahan tersebut merupakan rencana yang didasarkan pada kebutuhan target dan strategi kapasitas. Dari 3 target tersebut kemudian dipilih mana yang akan dikerjakan terlebih dahulu disesuaikan dengan *due date delivery*, misalnya yang akan dikerjakan lebih dahulu adalah target 50.000 *pairs*. Setelah itu akan dilakukan perhitungan agregat 10 minggu. Dari 10 minggu itu sistem akan melakukan pencarian *pairs* yang dipesan untuk rentang waktu dari hari Senin hingga Minggu pada *week* yang pertama. Rentang waktu ini disebut pula sebagai periode 1. Dari *pairs* yang dipesan di periode pertama kemudian akan di-*generate delivery note*, misal, pada periode pertama *pairs* yang dipesan hanya 7.000 *pairs* yang mana dari 7.000 *pairs* tersebut terdapat 100 *delivery note*. Dikarenakan pada periode pertama jumlah *pairs* yang dipesan belum mencukupi hingga 50.000 *pairs* maka dilakukan pencarian lagi untuk *week* yang kedua dengan rentang waktu hari Senin hingga Minggu atau ini biasa disebut sebagai periode kedua. Kemudian total *pairs* yang sudah tersedia di periode pertama dijumlahkan dengan yang ada pada periode kedua. Sistem akan melakukan pencarian terus

untuk periode-periode berikutnya hingga penjumlahan *pairs* tiap periode tersebut memenuhi target 50.000 *pairs*. Apabila sudah terpenuhi, maka dari kumpulan *delivery note* yang ter-generate kemudian akan dikelompokkan untuk dijadikan *M-Group*. Pengelompokkan ke dalam *M-Group* didasarkan pada:

- a. Metode transportasi atau *Mode of Transport (Mot)*
- b. Tanggal tenggat pengiriman yang sama
- c. Rute pengiriman yang sama
- d. Pelanggan yang sama
- e. Delivery yang sama
- f. *Season* yang sama
- g. dll

Jika beberapa kriteria tersebut tidak terpenuhi, maka dalam *M-Group* hanya terdiri dari satu *delivery* saja. Pengelompokkan-pengelompokkan ini dapat dilihat pada Gambar 4.2.

Satu *M-Group* dapat terdiri dari beberapa artikel dengan *volume* tiap artikel yang berbeda-beda. Satu *M-Group* bisa terdiri atas sedikit artikel atau banyak artikel. Saat proses *packing*, artikel yang sama diselesaikan terlebih dahulu untuk semua *M-Group* setelah itu baru menyelesaikan artikel yang lain. *M-Group* yang terbentuk kemudian dijadikan dalam satu *picking wave*. Pada dasarnya, dalam satu *picking wave* terdapat beberapa artikel dengan *volume* yang berbeda-beda. Artikel-artikel yang akan di-*pack* kemudian akan diurutkan mulai dari yang memiliki *volume* paling banyak hingga yang paling sedikit. Setelah itu, artikel-artikel tersebut akan dibagi pengerjaannya tiap *shift* karyawan seperti yang ditunjukkan pada Tabel 4.1.



Gambar 4.2 Bagan pengelompokkan sepatu dalam 1 picking wave

Tabel 4.1 Pembagian artikel tiap shift karyawan dalam 1 wave

Artikel	Total Volume	Shift
63151401705	1556	a
26055301001	1449	b
26055301072	1418	c
21401301065	915	a
21401302004	855	b
21020301001	821	c
52115401001	804	a
27001301072	681	b

Artikel	Total Volume	Shift
21022301001	667	c
21401301001	666	a
73459201001	648	b
21402301065	606	c
21026358795	563	a
21352301001	534	b
26000358685	509	c

Pada Tabel 4.1, terlihat terdapat beberapa artikel yang harus di-*pack* dalam satu *picking wave*. Artikel-artikel itu kemudian dibagi dalam 3 shift sebab shift karyawan dalam PDC adalah 3 shift. Proses pengepakan dilakukan mulai dengan shift pertama (shift a). Shift a akan mengerjakan artikel yang paling banyak lebih dahulu sesuai dengan pembagiannya. Artikel 63151401705 diselesaikan terlebih dahulu kemudian dilanjutkan dengan menyelesaikan artikel yang paling sedikit, yaitu artikel 21202635795. Proses ini berlanjut dengan alur pengerjaan bergantian dari yang paling banyak, kemudian yang paling sedikit, lalu yang paling banyak lagi, setelah itu yang paling sedikit, dan seterusnya. Artikel yang akan di-*pack* akan diturunkan pallet-nya dari *high rack* atau bisa disebut sebagai proses *picking*. Setelah itu dilakukan pemindaian menggunakan PDA untuk mengonfirmasi *pallet-pallet* tersebut. Sepatu atau artikel yang ada pada *pallet* kemudian diambil dan dipindai menggunakan PDA, apabila pemindaian sudah dilakukan maka artikel tersebut diletakkan pada *trolley*. Artikel diletakkan ke dalam *trolley* dan diurutkan sesuai dengan *size-nya*.

Sementara itu, artikel yang tidak di-*pack* dikembalikan ke *high rack* dan dilakukan pemindaian lokasi artikel tersebut. Untuk artikel yang kuantitasnya kecil, maka dilakukan penggabungan dengan artikel lain yang juga berkuantitas kecil dalam satu *pallet*. Setelah yakin bahwa artikel telah lengkap di-*pick* maka dicetak *packing label* (berisikan keterangan artikel, jumlah tiap artikel, jumlah

material yang terdapat dalam karton, dll). Artikel yang telah diletakkan pada *trolley* lantas diambil dan dimasukkan ke dalam *outerbox* sesuai dengan daftar *packing label*-nya. Apabila artikel telah dimasukkan ke dalam *outerbox*, maka dilakukan pemindaian terhadap *packing label* dan *shoes label*. Pemindaian ini dilakukan untuk memastikan bahwa yang tertera dalam *packing label* sama dengan jumlah *shoes label*. Setelah itu, secara otomatis *delivery note* tercetak sesuai dengan *packing label*. *Delivery note* tersebut kemudian ditempelkan pada *outerbox* lalu karton yang telah ditempel *delivery note* tersebut ditutup dengan menggunakan *tape*. Karton yang telah ditutup lantas ditimbang untuk mengetahui volume dan beratnya. Sementara itu, akan tercetak secara otomatis *address label (ADS)* yang berisi informasi hasil timbangan tersebut. ADS tersebut kemudian ditempelkan pada karton. Karton yang telah selesai di-*pack* kemudian diletakkan pada *pallet* dan ditempatkan pada *prepack area*. Detail prosedur *packing* PDC PT. XYZ dapat dilihat pada Gambar 1.1. Setelah proses pengepakan selesai maka dilanjutkan oleh proses *outgoing* PDC, yaitu proses persiapan untuk kemudian barang didistribusikan.

4.2. Persiapan

Persiapan merupakan fase awal sebelum dilakukan penggalan proses. Fase ini adalah fase yang penting sebab kebenaran pada hasil penggalan proses bergantung pada fase persiapan. Terdapat tiga tahapan dalam fase persiapan, yaitu studi literatur, pengumpulan data, dan standarisasi catatan kejadian.

Penjelasan dari setiap aktivitas yang telah dilakukan adalah sebagai berikut.

4.2.1. Pengumpulan Data

Tahap ini bertujuan untuk mengumpulkan data-data yang diperlukan terkait pembentukan model. Terdapat dua aktivitas dalam pengumpulan data, yaitu wawancara dan ekstraksi data.

4.2.1.1. Wawancara

Masukan dari proses wawancara adalah hasil dari studi literatur mengenai penggalian proses, model proses, algoritma *duplicate genetic*, pengukuran model, ekstraksi catatan kejadian, ProM, Petri Net.

Proses wawancara dilakukan selama 2 bulan bersamaan dengan proses ekstraksi data. Narasumber yang diwawancarai antara lain:

- a. Fungsional PDC
- b. Fungsional SAP modul *Warehouse*
- c. Manajer *Production Distribution Center* (PDC)

Dari hasil wawancara terdapat beberapa informasi yang diperoleh mengenai proses bisnis PDC di PT. XYZ. Beberapa aktivitas yang dilakukan oleh PDC PT. XYZ, yaitu:

1. *Good Receipt* merupakan aktivitas awal sepatu yang sudah jadi diterima dari produksi. Pada tahapan ini, sepatu diperiksa terlebih dahulu ketika sampai di PDC. Pemeriksaan meliputi artikel apa saja yang diterima dan kuantitas tiap artikel. Apabila artikel sudah sesuai, maka sepatu-sepatu tersebut diletakkan ke dalam *high rack*. Dan apabila tidak sesuai akan dikembalikan ke produksi.
2. *Incoming to High Rack* merupakan aktivitas meletakkan sepatu ke *high rack*. Sepatu-sepatu diletakkan dalam *high rack* sesuai dengan penomoran dua angka pertama artikelnya. Apabila sepatu telah diletakkan ke dalam *high rack*, kemudian dilakukan pemindaian kode sepatu dan kode alamat rak dengan menggunakan PDA sehingga setiap sepatu memiliki alamat logika yang menunjukkan lokasi sepatu.
3. *Packing Plan Creation* merupakan aktivitas membuat rencana *packing*. Pembuatan rencana *packing* dilakukan dengan menjalankan *A-Run* (SAP Allocation Running) untuk memeriksa stok yang ada dalam PDC setiap tengah malam waktu Denmark. Apabila sepatu yang akan di-*pack*

tersedia, maka akan dibuat dokumen *delivery*. Dari dokumen *delivery* tersebut akan dibentuk *M-Group*. Kemudian apabila semua telah lengkap dan sesuai, rencana *packing* siap untuk dilaksanakan.

4. *Picking* merupakan aktivitas pengambilan sepatu dari *high rack* untuk kemudian siap di-*pack*. *Picking* didasarkan pada perintah pengambilan yang terdapat pada rencana pengepakan. Metode *picking* dilakukan dengan mengambil satu artikel yang sama untuk semua pesanan terlebih dahulu dari berbagai *pallet* penyimpanan *material* untuk diselesaikan. Apabila artikel tersebut sudah selesai, maka lanjut ke artikel berikutnya.
5. *Packing* merupakan aktivitas mengemas sepatu-sepatu yang sudah ada dalam *innerbox* ke dalam *outerbox*. Proses pengepakan dilakukan dengan sistem *picking wave*. *Picking wave* merupakan sistem atau alur pengambilan sepatu dari *high rack* hingga sepatu selesai di-*pack*. Dalam *picking wave*, artikel yang sama diselesaikan terlebih dahulu untuk semua *M-Group* setelah itu baru menyelesaikan artikel yang lain. Sepatu-sepatu yang sudah di-*pack* akan diletakkan ke dalam *high rack* untuk siap di-*load*.
6. *Loading* merupakan aktivitas menyiapkan sepatu-sepatu yang telah di-*pack* untuk siap dikirim
7. *Shipment* merupakan proses mengirimkan sepatu ke pelanggan.

4.2.1.2. Ekstraksi Data

Masukan dari tahap ekstraksi data diperoleh dari keluaran hasil wawancara sebelumnya, yaitu pengetahuan mengenai aktivitas yang terjadi pada proses bisnis *packing* di PDC. Setelah memperoleh daftar aktivitas, langkah selanjutnya adalah mengekstraksi data yang berhubungan dengan aktivitas-aktivitas tersebut. Data yang diambil adalah data bulan Maret 2014, dimulai dari tanggal 1 Maret 2014 hingga tanggal 31 Maret 2014. Proses ekstraksi data adalah sebagai berikut:

4.2.1.2.1. Fase Persiapan

Pada fase ini dilakukan pengidentifikasian aktivitas dan atribut yang akan digunakan dalam proses ekstraksi, kemudian dilakukan pemetaan aktivitas dengan tabel database SAP.

a. Penentuan Aktivitas

Pada langkah ini ditentukan aktivitas-aktivitas yang relevan dengan proses. Berdasarkan hasil wawancara yang telah dijabarkan sebelumnya, aktivitas-aktivitas yang relevan dengan proses bisnis *packing* adalah:

- a) *Good Receipt (Start.Good Receipt)*
- b) *Incoming to High Rack (Start.High Rack)*
- c) *Picking (Start.Picking)*
- d) *Packing (Start.Packing)*
- e) *Loading (Start.Loading)*

b. Pemetaan Aktivitas

Pada langkah ini dipetakan aktivitas-aktivitas yang telah didefinisikan pada langkah penentuan aktivitas. Salah satu langkah memetakan aktivitas adalah dengan melihat relasi antar tabel. Tabel dasar dalam modul *Warehouse* (WM) dapat dilihat pada Tabel 4.2.

Tabel 4.2 Tabel Dasar modul Warehouse

Nama Tabel	Keterangan
LTAK	Transfer Order Header
LTAP	Transfer Order Item
LQUA	Quants
LEIN	Storage Unit Table
LTBK	Transfer Requirement Header
LTBP	Transfer Requirement Item
LAGP	Storage Bins
LIPS	Sales and Distribution Document Delivery : item data

Setelah mengetahui tabel dasar modul WM, maka dapat dilakukan pemetaan aktivitas. Pemetaan aktivitas dengan tabel WM dapat dilihat pada Tabel 4.3.

Tabel 4.3 Pemetaan aktivitas dengan atribut pada tabel WM

Aktivitas	Nama Tabel	Atribut
<ul style="list-style-type: none"> - <i>Start.Good Receipt</i> - <i>Start.High Rack</i> - <i>Start.Picking</i> - <i>Start.Packing</i> - <i>Start.Loading</i> 	<ul style="list-style-type: none"> - LTAP 	<ul style="list-style-type: none"> - Transfer Order Number - Item (SD) - Plant - Batch - Stock Category - Confirmation - Confirmation Date - Confirmation Time - User - GR Date - Source Storage Type - Source Storage Bin - Source target quantity - Destination Storage Type - Destination Storage Bin - Destination Target Quantity - Return Target Quantity - Source Storage Unit - Destination Storage Unit
	<ul style="list-style-type: none"> - LIPS 	<ul style="list-style-type: none"> - Material Number

Aktivitas	Nama Tabel	Atribut
		- Group - Delivery

Pada aktivitas *start.good receipt*, *start.high rack*, *start.picking*, *start.packing*, dan *start.loading* terdapat pula atribut *wave* yang tidak dapat ditemukan dalam tabel SAP, hal ini karena *wave* hanyalah sebuah atribut yang menandakan satu alur pengambilan sepatu hingga selesai di-*pack*. Sehingga data mengenai atribut *wave* didapatkan dari data *packing plan* bulan Maret 2014 yang diberikan oleh narasumber.

c. Memilih Atribut

Setelah memetakan aktivitas, langkah berikutnya adalah memilih atribut berdasarkan pemetaan aktivitas dengan tabel WM. Atribut yang dipilih akan digunakan untuk membentuk catatan kejadian yang terdiri dari id kasus, aktivitas, dan keterangan waktu. Atribut dari tabel WM yang dipilih ditunjukkan pada Tabel 4.4.

Tabel 4.4 Daftar atribut yang diekstrak

Nama Tabel	Atribut
LTAP	- Confirmation Date - Confirmation Time - GR Date
LIPS	- Material Number

Terdapat atribut yang tidak dapat dipetakan dalam tabel WM yaitu *wave*. Hal ini karena atribut tersebut tidak disimpan pada tabel.

4.2.1.2.2. Fase Ekstraksi

Pada fase ini dilakukan penentuan *outline* dari proses dan mengumpulkan seluruh informasi proses guna mengekstraksi catatan kejadian. Terdapat beberapa aktivitas dalam fase ekstraksi, yaitu:

a. Memilih aktivitas yang akan diekstrak

Langkah ini dilakukan untuk memastikan atribut yang dipilih telah memberikan gambaran yang jelas terkait dengan proses. Atribut-atribut yang benar-benar dipilih untuk diekstrak dapat dilihat pada Tabel 4.5.

Tabel 4.5 Daftar atribut yang pasti dipilih untuk diekstrak

Aktivitas	Nama Tabel	Atribut
<ul style="list-style-type: none"> - <i>Start.Good Receipt</i> - <i>Start.High Rack</i> - <i>Start.Picking</i> - <i>Start.Packing</i> - <i>Start.Loading</i> 	- LTAP	<ul style="list-style-type: none"> - Confirmation Date - Confirmation Time - GR Date
	- LIPS	- Material Number

b. Menentukan Skenario

Langkah berikutnya adalah menentukan skenario yang mungkin terjadi. Skenario yang mungkin terjadi dalam proses bisnis *packing* di PDC ditunjukkan pada Tabel 4.6.

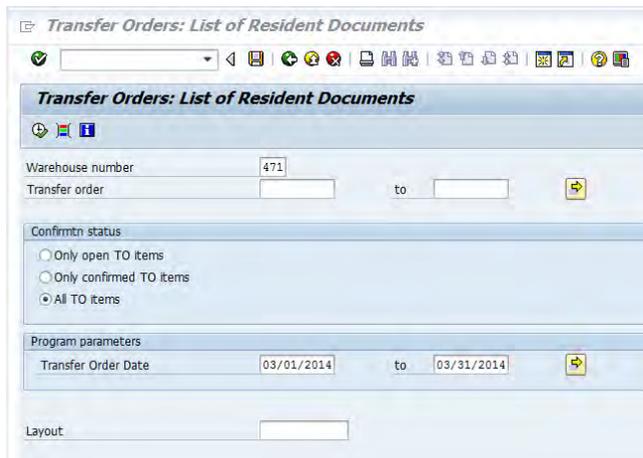
Tabel 4.6 Alur skenario yang mungkin terjadi

No.	Alur Skenario
1.	Start.Good Receipt → Start.High Rack → Start.Picking → Start.Packing → Start.Loading

c. Membangun Catatan Kejadian

Langkah terakhir adalah membangun catatan kejadian. Pembangunan catatan kejadian pada tugas akhir ini sedikit berbeda dari yang dijelaskan pada sub-bab 2.7 karena PT. XYZ tidak menyimpan data catatan kejadian pada database SAP-nya sehingga catatan kejadian dibentuk sendiri berdasarkan data yang diperoleh dari hasil *query* ke database SAP PT. XYZ. Langkah-langkah mengekstrak data adalah sebagai berikut:

- 1) Masuk ke SAP Easy Access, kemudian pada kotak kode transaksi, masukkan kode transaksi LT23. Kode ini merupakan kode yang digunakan untuk mengakses data *Transfer Order*. Setelah itu tekan enter pada *keyboard*.
- 2) Pada *Transfer Orders: List of Resident Documents* isi 471 untuk *warehouse number*. 471 merupakan nomor untuk gudang PDC. Kemudian centang *All TO items* untuk menampilkan semua *Transfer Order* dan isi rentang waktu *transfer order* yang ingin ditampilkan pada *field Transfer Order Date*. Untuk tugas akhir ini, rentang waktu data yang digunakan mulai dari tanggal 1 Maret 2014 hingga 31 Maret 2014.



Gambar 4.3 *Transfer Orders: List of Resident Documents*

- 3) Setelah itu akan muncul data *transfer order*. Data tersebut dapat diunduh dengan menekan tombol *spreadsheet* kemudian pilih format *file* untuk disimpan. Data disimpan dalam format .XML dan diberi nama 919-919, 200 ke 916 Maret, incoming, dan 010 ke 200 Maret. Penamaan *file* tersebut disesuaikan dengan posisi *transfer order*.

- 4) Data-data tersebut kemudian dijadikan 1 *file* dengan nama sesuai dengan nomor *wave*-nya agar lebih mudah untuk diolah. Terdapat 4 *wave* yang digunakan sehingga terdapat 4 *file* data, yaitu 21145, 21110, 21180, dan 21207.
- 5) Dari *file* 21110, 21145, 21180, dan 21207 kemudian disusun catatan kejadian. Seperti yang telah dijelaskan sebelumnya bahwa catatan kejadian terdiri atas minimal 3 atribut, yaitu id kasus, aktivitas, dan keterangan waktu sehingga dari file 21110, 21145, 21180, dan 21207, diambil kolom:
 - a. *Wave* berasal dari atribut *picking wave group* yang ada pada data rencana pengepakan (*packing plan*)
 - b. *Artikel* berasal dari atribut *material number*
 - c. *Tanggal* berasal dari atribut *confirmation date* dan *GR Date*
 - d. *Waktu* berasal dari atribut *confirmation time*
 - e. *Aktivitas* merupakan nama aktivitas
 - f. *Konfirmasi* merupakan banyaknya *confirmation time* yang dilakukan untuk 1 *material* dalam 1 *wave*.

Hasil pengambilan kolom ditunjukkan pada Gambar 4.4.

A	B	C	D	E	F
wave	artikel	tanggal	waktu	konfirmasi	aktivitas
21110	24262301001	3/21/2014	07:57:24	1	Start.Picking
21110	24262301001	3/21/2014	07:57:24	2	Start.Picking
21110	24262301001	3/21/2014	07:57:24	3	Start.Picking
21110	24262301001	3/21/2014	07:57:24	4	Start.Picking
21110	24262301001	3/21/2014	07:57:24	5	Start.Picking
21110	24262301001	3/21/2014	07:57:24	6	Start.Picking
21110	24262301001	3/21/2014	07:57:24	7	Start.Picking
21110	24262301001	3/21/2014	07:57:24	8	Start.Picking
21110	24262301001	3/21/2014	07:57:24	9	Start.Picking
21110	24262301001	3/22/2014	01:20:57	2	Start.Packing

Gambar 4.4 Tampilan data *file* 21145

- 6) Untuk menghasilkan id kasus pada catatan kejadian, maka kolom *wave*, artikel, dan konfirmasi digabung dengan memanfaatkan *formula* yang ada pada excel.

A	B	C	D	E	F	G	H
wave	artikel	tanggal	waktu	konfirmasi	aktivitas		
21110	24262301001	3/21/2014	07:57:24	1	Start.Picking	=CONCATENATE(A2,B2,E2)	
21110	24262301001	3/21/2014	07:57:24	2	Start.Picking	211102426	3/21/2014 7:57:24
21110	24262301001	3/21/2014	07:57:24	3	Start.Picking	211102426	3/21/2014 7:57:24
21110	24262301001	3/21/2014	07:57:24	4	Start.Picking	211102426	3/21/2014 7:57:24
21110	24262301001	3/21/2014	07:57:24	5	Start.Picking	211102426	3/21/2014 7:57:24
21110	24262301001	3/21/2014	07:57:24	6	Start.Picking	211102426	3/21/2014 7:57:24

Gambar 4.5 *Formula* menggabungkan kolom

- 7) Setelah menggabungkan kolom *wave*, artikel, dan konfirmasi untuk membentuk *id*, langkah selanjutnya adalah menggabungkan kolom *date* dan *time* dengan menggunakan bantuan excel untuk menghasilkan keterangan waktu. Penggabungan ini dilakukan agar keterangan waktu pada catatan kejadian dapat dibaca oleh perangkat lunak *ProM*.
- 8) Hasil penggabungan kolom pada langkah sebelumnya ditunjukkan pada Gambar 4.6. *File* ini kemudian disimpan dengan format excel atau CSV dan diberi nama *EventLog*.

A	B	C
id kasus	keterangan waktu	aktivitas
21207073131587651	3/27/2014 4:51:12	Start.Packing
21207073131587652	3/27/2014 4:51:22	Start.Packing
21207073131587653	3/27/2014 4:51:45	Start.Packing
21207073131587654	3/27/2014 4:52:00	Start.Packing
21207073131587655	3/27/2014 4:52:28	Start.Packing
21207073131587656	3/27/2014 4:52:48	Start.Packing
21207073131587651	3/27/2014 0:52:48	Start.Picking
21207073131587652	3/27/2014 0:52:48	Start.Picking
21207073131587653	3/27/2014 0:52:48	Start.Picking

Gambar 4.6 Hasil penggabungan kolom

4.2.2. Standardisasi Catatan Kejadian

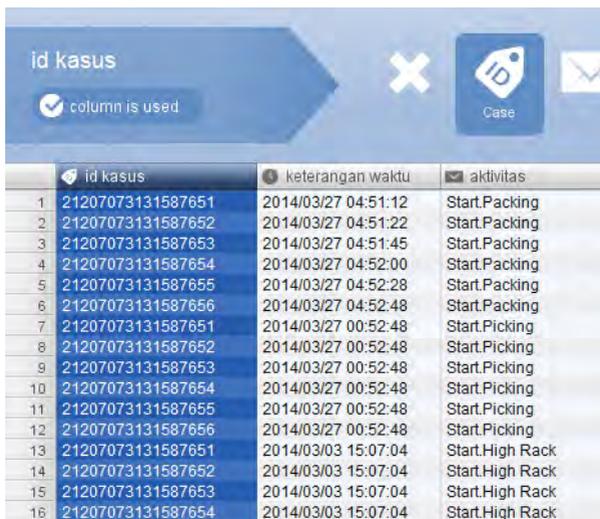
Langkah berikutnya setelah membangun catatan kejadian adalah mengubah catatan kejadian yang masih berekstensi CSV atau excel menjadi .mxml. Potongan catatan kejadian yang didapatkan pada langkah pengumpulan data ditunjukkan pada Tabel 4.7.

Tabel 4.7 Potongan Catatan Kejadian

id kasus	keterangan waktu	aktivitas
21207073131587651	3/27/2014 4:51:12	Start.Packing
21207073131587652	3/27/2014 4:51:22	Start.Packing
21207073131587653	3/27/2014 4:51:45	Start.Packing
21207073131587654	3/27/2014 4:52:00	Start.Packing
21207073131587655	3/27/2014 4:52:28	Start.Packing
21207073131587656	3/27/2014 4:52:48	Start.Packing
21207073131587651	3/27/2014 0:52:48	Start.Picking
21207073131587652	3/27/2014 0:52:48	Start.Picking
21207073131587653	3/27/2014 0:52:48	Start.Picking
21207073131587654	3/27/2014 0:52:48	Start.Picking
21207073131587655	3/27/2014 0:52:48	Start.Picking
21207073131587656	3/27/2014 0:52:48	Start.Picking
21207073131587651	3/3/2014 15:07:04	Start.High Rack
21207073131587652	3/3/2014 15:07:04	Start.High Rack
21207073131587653	3/3/2014 15:07:04	Start.High Rack
21207073131587654	3/3/2014 15:07:04	Start.High Rack
21207073131587655	3/3/2014 15:07:04	Start.High Rack
21207073131587656	3/3/2014 15:07:04	Start.High Rack
21207073131587651	3/3/2014 0:01:00	Start.Good Receipt
21207073131587652	3/3/2014 0:01:00	Start.Good Receipt
21207073131587653	3/3/2014 0:01:00	Start.Good Receipt
21207073131587654	3/3/2014 0:01:00	Start.Good Receipt

id kasus	keterangan waktu	aktivitas
21207073131587655	3/3/2014 0:01:00	Start.Good Receipt
21207073131587656	3/3/2014 0:01:00	Start.Good Receipt

Catatan kejadian diubah menggunakan perangkat lunak Disco. Disco membaca catatan kejadian dalam bentuk CSV atau excel ditunjukkan pada Gambar 4.7.



The screenshot shows the Disco application interface. At the top, there is a search bar labeled 'id kasus' with a checkmark and the text 'column is used'. To the right of the search bar are icons for a close button (X), a case icon (ID Case), and an email icon. Below the search bar is a table with three columns: 'id kasus', 'keterangan waktu', and 'aktivitas'. The table contains 16 rows of data, with the first row highlighted in blue. The data in the table is as follows:

	id kasus	keterangan waktu	aktivitas
1	21207073131587651	2014/03/27 04:51:12	Start.Packing
2	21207073131587652	2014/03/27 04:51:22	Start.Packing
3	21207073131587653	2014/03/27 04:51:45	Start.Packing
4	21207073131587654	2014/03/27 04:52:00	Start.Packing
5	21207073131587655	2014/03/27 04:52:28	Start.Packing
6	21207073131587656	2014/03/27 04:52:48	Start.Packing
7	21207073131587651	2014/03/27 00:52:48	Start.Picking
8	21207073131587652	2014/03/27 00:52:48	Start.Picking
9	21207073131587653	2014/03/27 00:52:48	Start.Picking
10	21207073131587654	2014/03/27 00:52:48	Start.Picking
11	21207073131587655	2014/03/27 00:52:48	Start.Picking
12	21207073131587656	2014/03/27 00:52:48	Start.Picking
13	21207073131587651	2014/03/03 15:07:04	Start.High Rack
14	21207073131587652	2014/03/03 15:07:04	Start.High Rack
15	21207073131587653	2014/03/03 15:07:04	Start.High Rack
16	21207073131587654	2014/03/03 15:07:04	Start.High Rack

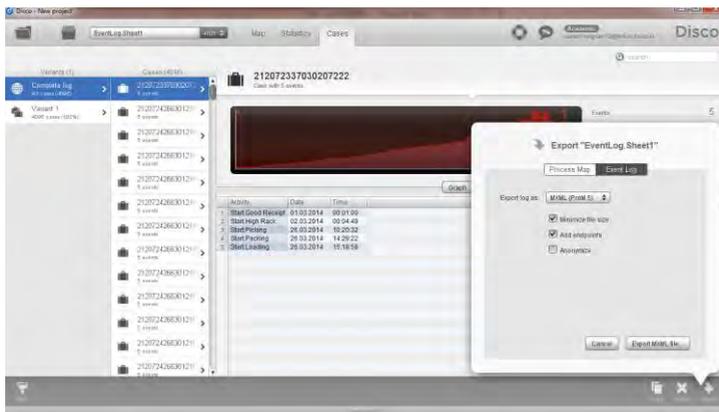
Gambar 4.7 Tampilan data di Disco

Pada Gambar 4.7 terlihat bahwa Disco mampu mendeteksi atribut-atribut yang terdapat pada *file* excel catatan kejadian. Hal ini bisa dilihat dari kolom id kasus yang langsung dikenali sebagai id, kolom keterangan waktu sebagai keterangan waktu atau *timestamp*, dan kolom aktivitas sebagai aktivitas atau *activity*.

Selain ketiga atribut yang telah disebutkan sebelumnya, Disco juga mampu mendeteksi atribut lain, yaitu *resource* dan *other*. *Resource* mengacu pada kolom originator dan *other* mengacu pada

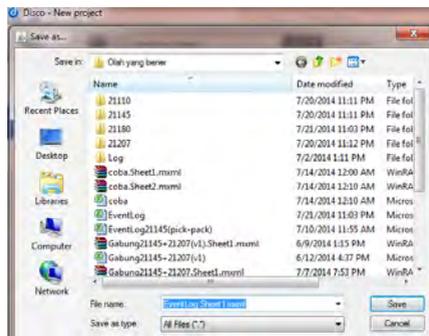
kolom lainnya yaitu kolom selain *id*, *timestamp*, *activity*, dan *resource*.

Untuk melakukan konversi dari CSV atau excel ke format .mxml dapat dilakukan dengan menekan tombol *Start Import*. Selanjutnya hasil konversi tersebut diekspor ke dalam format .mxml. Ekspor hasil konversi ke dalam format .mxml ditunjukkan pada Gambar 4.8.



Gambar 4.8 Ekspor hasil konversi dengan Disco

Kemudian hasil ekspor diberi nama dan disimpan dalam folder yang diinginkan, seperti ditunjukkan pada Gambar 4.9.



Gambar 4.9 Proses menyimpan hasil konversi

Keluaran hasil konversi yang berformat *.mxml ditunjukkan pada Gambar 4.10.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- MXML version 1.0 -->
<!-- Created by Fluxicon Disco (http://fluxicon.com/disco/ -->
<!-- (c) 2012 Fluxicon Process Laboratories - http://fluxicon.com/ -->
<WorkflowLog xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://is.tm.tue.nl/research/processmining/WorkflowLog.xsd">
  <Source program="Fluxicon Disco"/>
  <Process id="Gabung21145+21207.Sheet1.mxml.gz" description="Converted to MXML by Fluxicon Disco">
    <ProcessInstance id="21145158571826007358752">
      <AuditTrailEntry>
        <WorkflowModelElement>Good Receipt</WorkflowModelElement>
        <EventType>complete</EventType>
        <Timestamp>2014-03-14T00:01:00.000+07:00</Timestamp>
      </AuditTrailEntry>
      <AuditTrailEntry>
        <WorkflowModelElement>High Rack</WorkflowModelElement>
        <EventType>complete</EventType>
        <Timestamp>2014-03-14T16:15:16.000+07:00</Timestamp>
      </AuditTrailEntry>
      <AuditTrailEntry>
```

Gambar 4.10 Potongan hasil konversi catatan kejadian berformat *.mxml

Setiap kasus pada catatan kejadian yang berformat .mxml memiliki pembatas yaitu tag <ProcessInstance> </ProcessInstance> dan untuk aktivitas satu dengan lainnya dalam sebuah kasus dibatasi dengan tag <AuditTrailEntry> </AuditTrailEntry>. Contoh log aktivitas pada sebuah kasus ditunjukkan pada Gambar 4.11.

```
<AuditTrailEntry>
  <WorkflowModelElement>Good Receipt</WorkflowModelElement>
  <EventType>complete</EventType>
  <Timestamp>2014-03-14T00:01:00.000+07:00</Timestamp>
</AuditTrailEntry>
```

Gambar 4.11 Potongan hasil konversi catatan kejadian sebuah aktivitas

Berdasarkan Gambar 4.11 dapat diketahui bahwa aktivitas yang terdapat pada potongan hasil konversi tersebut adalah aktivitas *Start.Good Receipt*. Hal ini dapat diketahui karena atribut yang

menandakan suatu aktivitas dibatasi oleh tag `<WorkflowModelElement>` dan atribut keterangan waktu dibatasi oleh tag `<Timestamp>`.

4.3. Penggalian Proses

Pada fase ini dibangun model proses dengan menggunakan perangkat lunak ProM 5.2 dan Algoritma *Duplicate Genetic*. Langkah-langkah pada fase ini dijelaskan sebagai berikut.

4.3.1. Masukan

Masukan pada tahap ini berupa catatan kejadian yang telah disimpan dalam format *.mxml seperti yang ditunjukkan pada Gambar 4.10.

4.3.2. Proses

Catatan kejadian yang telah berformat *.mxml diimpor ke dalam perangkat lunak ProM. ProM kemudian membaca catatan kejadian tersebut seperti yang ditunjukkan pada Gambar 4.12.



Gambar 4.12 Tampilan ProM membaca catatan kejadian

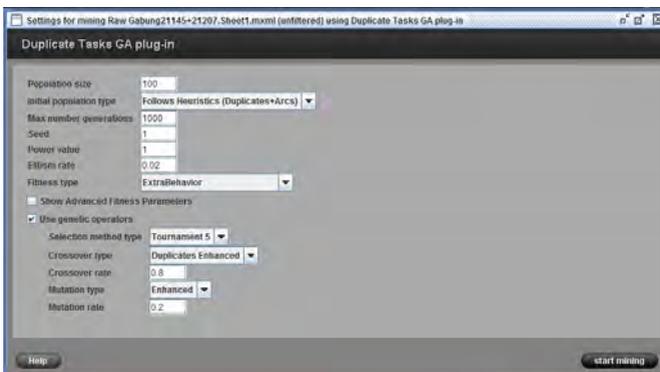
Setelah itu tekan tombol *Start analyzing this log* untuk memulai penggalian proses. Selanjutnya pilih algoritma yang digunakan untuk melakukan penggalian proses. Pada tugas akhir ini, algoritma yang

digunakan adalah Algoritma *Duplicate Genetic* seperti yang ditunjukkan pada Gambar 4.13.



Gambar 4.13 Pemilihan Algoritma *Duplicate Genetic* pada ProM

Kemudian muncul parameter yang perlu diatur untuk menghasilkan model proses. Parameter yang perlu diatur ditunjukkan pada Gambar 4.14. Selanjutnya untuk melakukan penggalan proses, pilih tombol *Start Mining* yang terletak di pojok kanan bawah.



Gambar 4.14 Tampilan mengatur parameter pada ProM

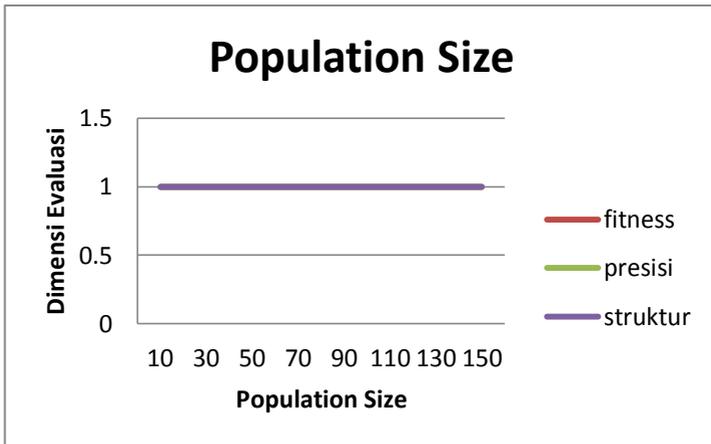
Sebelum itu, dilakukan percobaan dengan mengubah-ubah parameter ProM algoritma *Duplicate Genetic* untuk mengetahui pengaruh dari masing-masing parameter terhadap model yang akan dihasilkan. Percobaan dilakukan dengan mengubah nilai salah satu parameter (nilainya dinaikkan atau diturunkan) sementara nilai parameter lainnya adalah default ProM. Detail percobaan parameter ditunjukkan pada Tabel 4.8.

Pop. Size	Initial Pop.Type	Max Number Generations	Seed	Power Value	Elitism Rate	<i>Fitness</i> Type	Selection Method Type	Cross over Type	Cross over Rate	Mutation Type	Mutation Rate
Default	Causal Heuristic (<i>Duplicate</i>)	Default	Default	Default	Default	Default	Default	Default	Default	Default	Default
Default	Default	100	Default	Default	Default	Default	Default	Default	Default	Default	Default
Default	Default	300	Default	Default	Default	Default	Default	Default	Default	Default	Default
Default	Default	500	Default	Default	Default	Default	Default	Default	Default	Default	Default
Default	Default	700	Default	Default	Default	Default	Default	Default	Default	Default	Default
Default	Default	900	Default	Default	Default	Default	Default	Default	Default	Default	Default
Default	Default	1000	Default	Default	Default	Default	Default	Default	Default	Default	Default
Default	Default	1100	Default	Default	Default	Default	Default	Default	Default	Default	Default

Dari percobaan parameter tersebut didapatkan hasil sebagai berikut:

- Population Size

Population size merupakan jumlah individu yang ditetapkan untuk dicari. Adapun individu yang dicari adalah individu terkuat. Misal, dalam ProM ditetapkan population size sebesar 100. Hal ini berarti bahwa akan ditetapkan 100 individu terkuat untuk dicari. Pengaruh parameter ini dapat dilihat pada Gambar 4.15.



Gambar 4.15 Grafik parameter population size

Berdasarkan Gambar 4.15 dapat dilihat bahwa perubahan parameter population size terhadap nilai *fitness*, presisi, dan struktur ternyata tidak memberikan pengaruh. Nilai *fitness*, presisi, dan struktur tetap stabil pada angka 1.

- Initial Population Type

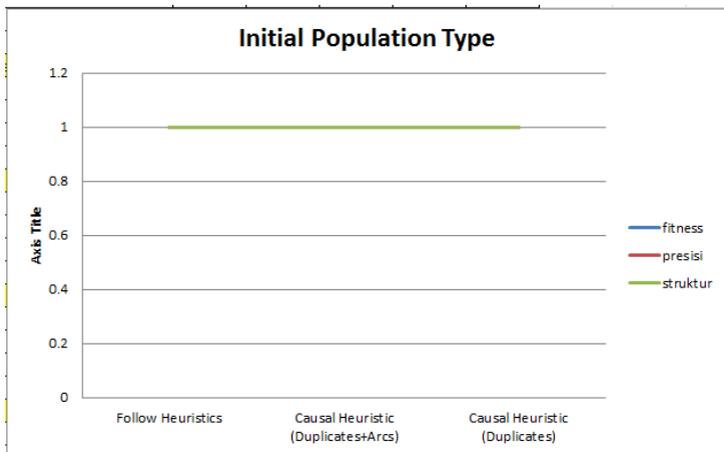
Initial population type mengatur bagaimana populasi seharusnya dibangun. Populasi dibangun berdasarkan hubungan ketergantungan antar aktivitas dan ada tidaknya aktivitas duplikat. Dalam initial population type terdapat 3 jenis pilihan, yaitu:

- *Causal Heuristics (Duplicate+Arcs)*

Heuristik digunakan untuk mengatur hubungan ketergantungan antar aktivitas. Heuristik ini didasarkan pada *causal information* yang digunakan untuk mengatur jumlah duplikat tiap aktivitas.

- *Causal Heuristics (Duplicate)*
 Pada pilihan ini, hubungan ketergantungan antar aktivitas diatur secara acak. Penemuan didasarkan pada *causal information* yang digunakan untuk mengatur jumlah duplikat tiap aktivitas.
- *Follow Heuristics (Dulicates+Arcs)*
 Heuristik digunakan untuk mengatur hubungan ketergantungan antar aktivitas. Heuristik didasarkan pada *follows information* yang digunakan untuk mengatur jumlah duplikat tiap aktivitas.

Untuk melihat pengaruh parameter ini ditunjukkan pada Gambar 4.16.



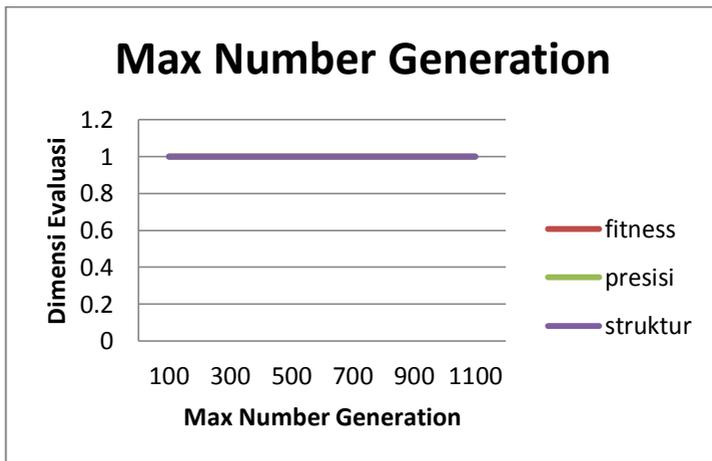
Gambar 4.16 Grafik parameter initial population type

Berdasarkan Gambar 4.16 dapat dilihat bahwa perubahan parameter initial population type terhadap nilai *fitness*, presisi, dan struktur ternyata tidak memberikan

pengaruh. Nilai *fitness*, presisi, dan struktur tetap stabil pada angka 1.

- Maximum Number Generation

Maximum numer generation mengatur jumlah maksimum berapa kali n yang dapat diiterasi oleh algoritma genetika. Parameter ini berhubungan dengan kriteria menghentikan. Dengan kata lain, algoritma *duplicate genetic* berhenti ketika (i) menemukan individu yang memiliki nilai *fitness* maksimal, (ii) diiterasi sebanyak n kali, atau (iii) individu terkuat tidak harus mengubah iterasi $n/2$ dalam satu waktu. Untuk mengetahui pengaruh parameter ini ditunjukkan pada Gambar 4.17.

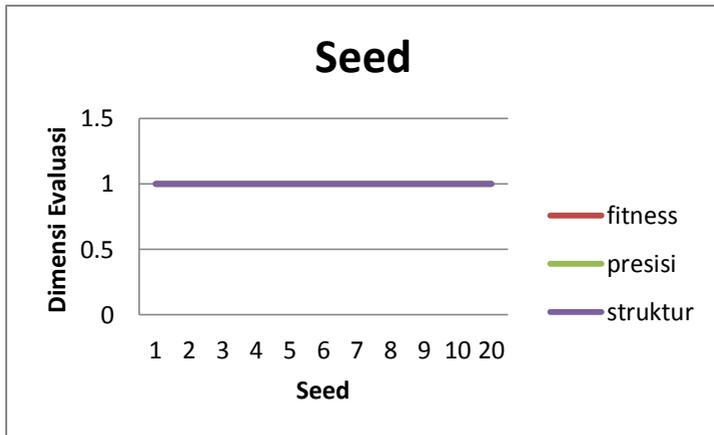


Gambar 4.17 Grafik parameter maximum number generation

Berdasarkan Gambar 4.17 dapat dilihat bahwa perubahan parameter maximum number generation terhadap nilai *fitness*, presisi, dan struktur ternyata tidak memberikan pengaruh. Nilai *fitness*, presisi, dan struktur tetap stabil pada angka 1.

- Seed

Mengatur titik yang digunakan untuk menghasilkan nomor acak. Untuk mengetahui pengaruh parameter ini ditunjukkan pada Gambar 4.18.

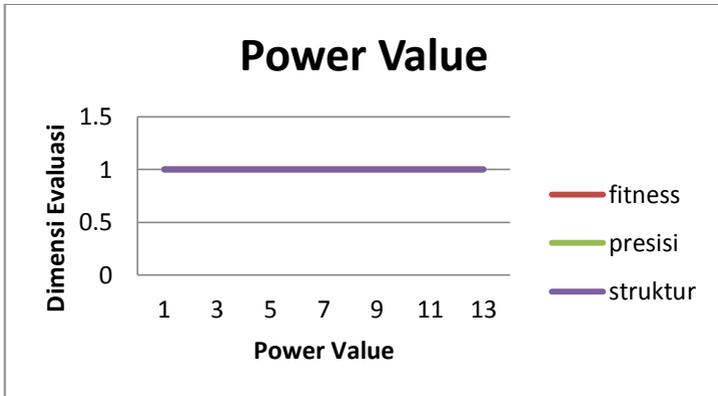


Gambar 4.18 Grafik parameter seed

Berdasarkan Gambar 4.18 dapat dilihat bahwa perubahan parameter seed terhadap nilai *fitness*, presisi, dan struktur ternyata tidak memberikan pengaruh. Nilai *fitness*, presisi, dan struktur tetap stabil pada angka 1.

- Power Value

Mengatur *power value* yang digunakan oleh heuristik untuk membangun populasi awal. Parameter ini digunakan untuk mengontrol “pengaruh” dari heuristic dalam probabilitas pengaturan hubungan ketergantungan antar dua aktivitas dengan rumus $(D(t_1, t_2))^p$. Nilai power value harus ganjil untuk menghindari hasil yang negatif (Medeiros, Weijters, & Aalst, 2005). Untuk melihat pengaruh parameter ini ditunjukkan pada Gambar 4.19.



Gambar 4.19 Grafik parameter power value

Berdasarkan Gambar 4.19 dapat dilihat bahwa perubahan parameter power value terhadap nilai *fitness*, presisi, dan struktur ternyata tidak memberikan pengaruh. Nilai *fitness*, presisi, dan struktur tetap stabil pada angka 1.

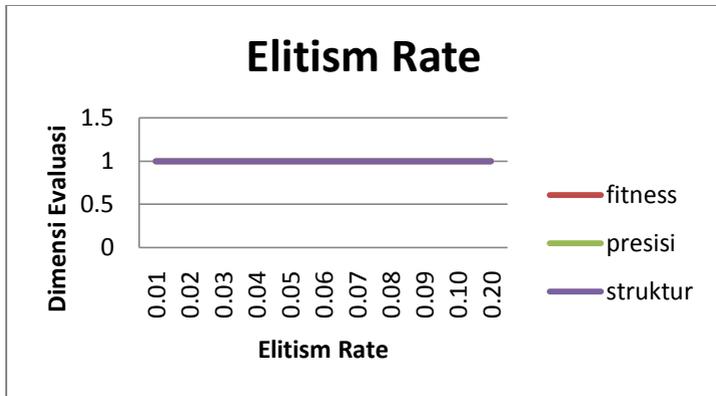
- Elitism Rate

Mengatur persentase individu terkuat dalam suatu generasi yang akan disalin ke generasi selanjutnya. Misalnya, elitism rate sebesar 0,02 artinya sebesar 2% individu terbaik dalam populasi disalin ke populasi selanjutnya. Untuk melihat pengaruh parameter ini ditunjukkan pada Gambar 4.20.

Berdasarkan Gambar 4.20 dapat dilihat bahwa perubahan parameter elitism rate terhadap nilai *fitness*, presisi, dan struktur ternyata tidak memberikan pengaruh. Nilai *fitness*, presisi, dan struktur tetap stabil pada angka 1.

- *Fitness* Type

Mengatur jenis *fitness* yang digunakan oleh DGA untuk menilai kualitas individu. Kualitas individu pada dasarnya diatur dengan menjalankan *log traces*. Untuk melihat pengaruh parameter ini ditunjukkan pada Gambar 4.21.



Gambar 4.20 Grafik parameter elitism rate

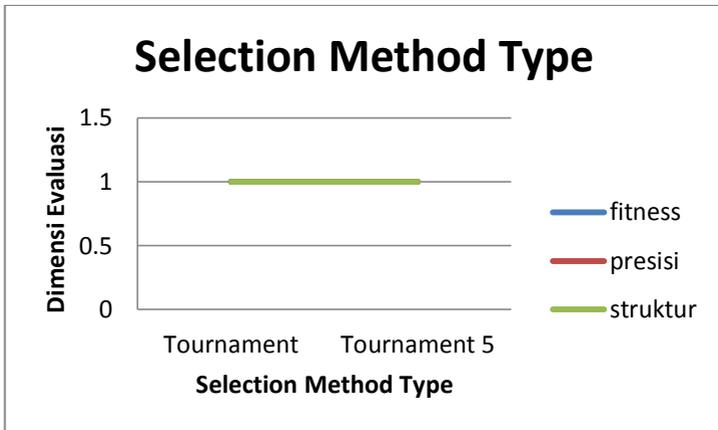


Gambar 4.21 Grafik parameter *fitness* type

Berdasarkan Gambar 4.21 dapat dilihat bahwa perubahan parameter *fitness* type terhadap nilai *fitness*, presisi, dan struktur ternyata tidak memberikan pengaruh. Nilai *fitness*, presisi, dan struktur tetap stabil pada angka 1.

- Selection Method Type

Mengatur bagaimana *parents* untuk operator genetic akan dipilih. Terdapat dua *method*, yaitu *Tournament* dan *Tournament5*. *Tournament* bekerja dengan secara acak memilih dua individu dalam populasi dan mengembalikan individu terkuat dalam waktu 75% dari waktu yang ada, dan individu yang kurang kuat dikembalikan dalam kurun waktu 25% dari waktu yang ada. *Tournament5* secara acak memilih lima individu dalam populasi dan selalu mengembalikan individu terkuat. Untuk melihat pengaruh parameter ini ditunjukkan pada Gambar 4.22.



Gambar 4.22 Grafik parameter selection method type

Berdasarkan Gambar 4.22 dapat dilihat bahwa perubahan parameter selection method type terhadap nilai *fitness*, presisi, dan struktur ternyata tidak memberikan pengaruh. Nilai *fitness*, presisi, dan struktur tetap stabil pada angka 1.

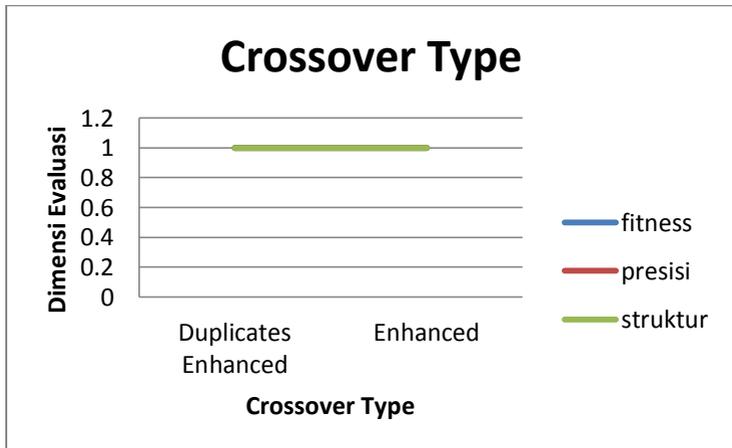
- Crossover Type

Mengatur bagaimana dua *parent* (individu terpilih) akan dikombinasikan kembali. Terdapat dua jenis *crossover*, yaitu *Enhanced* dan *Duplicates Enhanced*. Perbedaan keduanya terletak pada tingkat bekerjanya, *Enhanced*

bekerja pada tingkat aktivitas sedangkan *duplicates enhanced* bekerja pada tingkat duplikat.

- *Enhanced*: Menukar subset dalam masukan/keluaran dari kumpulan aktivitas yang memiliki identifier yang sama. Dengan kemungkinan yang sama, subset yang ditukar dapat: (i) mencakup kumpulan masukan/keluaran, (ii) menggabungkan dengan beberapa subset yang tidak ditukar dalam kumpulan masukan/keluaran, atau (iii) menambah beberapa subset yang tidak ditukar yang beririsan dengan subset yang ditukar.
- *Duplicates Enhanced*: Sama dengan jenis “*Enhanced*”, tapi bekerja pada tingkat duplikat. Dengan kata lain bahwa dua identifier aktivitas yang berbeda dapat dipetakan pada label yang sama.

Untuk melihat pengaruh parameter ini ditunjukkan pada Gambar 4.23.



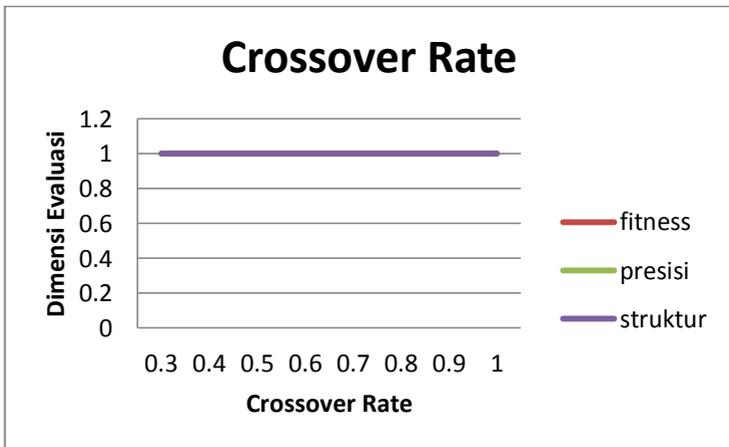
Gambar 4.23 Grafik parameter crossover type

Berdasarkan Gambar 4.23 dapat dilihat bahwa perubahan parameter crossover type terhadap nilai *fitness*,

presisi, dan struktur ternyata tidak memberikan pengaruh. Nilai *fitness*, presisi, dan struktur tetap stabil pada angka 1.

- Crossover Rate

Mengatur kemungkinan dua *parent* yang akan dikombinasikan kembali untuk menghasilkan dua *offsprings* untuk generasi selanjutnya. Jika kemungkinannya sama dengan 0, maka setelah *crossover*, *offsprings* sama dengan *parents*. Untuk melihat pengaruh parameter ini ditunjukkan pada Gambar 4.24.



Gambar 4.24 Grafik parameter crossover rate

Berdasarkan Gambar 4.24 dapat dilihat bahwa perubahan parameter crossover rate terhadap nilai *fitness*, presisi, dan struktur ternyata tidak memberikan pengaruh. Nilai *fitness*, presisi, dan struktur tetap stabil pada angka 1.

- Mutation Rate

Mengatur kemungkinan suatu individu akan dimutasi. Untuk melihat pengaruh parameter ini ditunjukkan pada Gambar 4.25.



Gambar 4.25 Grafik parameter mutation rate

Berdasarkan Gambar 4.25 dapat dilihat bahwa perubahan parameter mutation rate terhadap nilai *fitness*, presisi, dan struktur ternyata tidak memberikan pengaruh. Nilai *fitness*, presisi, dan struktur tetap stabil pada angka 1.

Berdasarkan grafik masing-masing parameter dapat diketahui pengaruh parameter-parameter tersebut untuk menghasilkan nilai *fitness*, presisi, dan struktur yang optimum (nilai optimum = 1). Hasil percobaan membuktikan bahwa tidak terdapat pengaruh perubahan parameter terhadap nilai *fitness*, presisi, dan struktur model proses. Baik parameter default maupun parameter percobaan memiliki nilai *fitness*, presisi, dan struktur yang optimum. Untuk lebih jelas dapat dilihat pada lampiran A.

4.3.3. Keluaran

Hasil model proses yang dibentuk dengan parameter default ditunjukkan pada Gambar 4.26.



Gambar 4.26 Model proses yang dihasilkan oleh parameter default

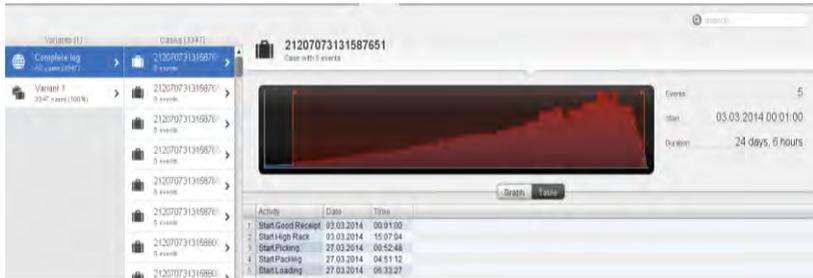
Berdasarkan Gambar 4.26 dapat diketahui bahwa model proses yang terbentuk hanya berupa garis lurus. Hal ini menunjukkan bahwa model proses yang ada dalam PDC bersifat berurutan sehingga suatu aktivitas tak akan bisa lanjut ke aktivitas berikutnya apabila aktivitas sebelumnya belum selesai dikerjakan.

4.4. Pengukuran Model

Tahap ini dilakukan setelah model proses dari ProM terbentuk. Tujuan pengukuran model adalah untuk mengetahui kebenaran dari model yang dihasilkan. Pengukuran dilakukan menggunakan 3 dimensi, yaitu *fitness*, presisi, dan struktural. Cara penghitungan untuk ketiga dimensi ini telah dijelaskan pada bab 2 sebelumnya.

4.4.1. Masukan Pengukuran Model

Masukan pengukuran model adalah model proses yang telah dihasilkan melalui perangkat lunak ProM dan skenario. Skenario adalah rangkaian aktivitas yang mungkin terjadi. Pembentukan skenario yang mungkin terjadi dilakukan dengan menggunakan Disco. Disco membentuk skenario berdasarkan *log trace* yang ada pada catatan kejadian. Kemudian *log trace* yang sama dikelompokkan tanpa memandang waktu. Setelah itu, masing-masing skenario dihitung frekuensi kemunculannya. Skenario yang terbentuk pada Disco ditunjukkan pada Gambar 4.27.



Gambar 4.27 Skenario yang terbentuk pada Disco

Berdasarkan Gambar 4.27 terlihat bahwa hanya terbentuk satu skenario. Hal ini menunjukkan bahwa urutan aktivitas yang mungkin terjadi hanya satu. Skenario yang terbentuk ditunjukkan pada Tabel 4.9.

Tabel 4.9 Rincian skenario

Skenario	Log trace	Jumlah Aktivitas	Frekuensi	Total Aktivitas
1	Good Receipt → High Rack → Picking → Packing → Loading	5	4096	20480

Berdasarkan Tabel 4.9 dapat dilihat bahwa terdapat satu skenario yang terbentuk dari catatan kejadian. Skenario memiliki jumlah aktivitas yang menandakan banyaknya aktivitas yang terdapat dalam satu skenario dan ditunjukkan pada kolom “Jumlah Aktivitas”. Selain itu, skenario juga memiliki frekuensi yang menandakan jumlah kemunculan skenario dalam catatan kejadian dan ditunjukkan pada kolom “Frekuensi”. Kolom “Total Aktivitas” dihitung dari perkalian antara “Jumlah Aktivitas” dan “Frekuensi”.

4.4.2. Proses Pengukuran Model

Pengukuran performa model dilakukan dengan mengukur dari tiga dimensi, yaitu *fitness*, presisi, dan struktural.

4.4.2.1. Perhitungan *Fitness*

Perhitungan *fitness* dilakukan untuk mengetahui seberapa besar kesesuaian antara catatan kejadian dan model proses yang dihasilkan. Perhitungan *fitness* dilakukan dengan menggunakan rumus (8) pada sub-bab 2.9.1. Pengukuran *fitness* didasarkan pada skenario yang terbentuk dari penggalian proses. Skenario ini merupakan masukan yang digunakan untuk mengukur nilai *fitness*. Cara kerja proses perhitungan *fitness* adalah dengan melakukan aktivitas *log replay*. *Log replay* merupakan pengulangan kembali kasus pada data catatan kejadian dengan menyesuaikan pada alur model proses yang terbentuk. Pada *log replay* akan dihitung pergerakan token-token dalam model.

4.4.2.1.1. Masukan Perhitungan *Fitness*

Seperti yang telah dijelaskan sebelumnya bahwa masukan perhitungan *fitness* adalah skenario yang terbentuk dari penggalian proses. Dari hasil penggalian proses didapatkan skenario seperti pada Tabel 4.10.

Tabel 4.10 Skenario pada catatan kejadian

No.	<i>Log trace</i>	Frekuensi
1.	Start.Good Receipt → Start.High Rack → Start.Picking → Start.Packing → Start.Loading	4096

4.4.2.1.2. Proses Perhitungan *Fitness*

Pada proses perhitungan *fitness* terdapat empat variabel yang digunakan dalam perhitungan, yaitu jumlah token yang diproduksi (p), jumlah token yang dikonsumsi (c), jumlah token yang hilang (m), dan jumlah token yang tersisa (r). Untuk mempermudah proses perhitungan, maka terdapat beberapa langkah yang perlu dilakukan, yaitu:

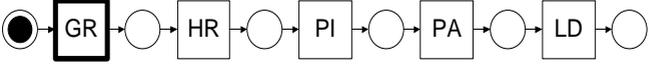
- a. Mengubah tampilan model proses sehingga berbentuk garis lurus mengikuti aktivitas skenario.
- b. Menginisiasi nama aktivitas agar lebih mudah divisualisasikan, sehingga menjadi:

- Start.Good Receipt : GR
 Start.High Rack : HR
 Start.Picking : PI
 Start.Packing : PA
 Start.Loading : LD
- Jika transisi berwarna kuning  menandakan bahwa transisi telah dilewati sebanyak satu kali
 - Jika transisi berwarna hijau  menandakan bahwa transisi telah dilewati sebanyak dua kali
 - Jika transisi berwarna abu-abu  menandakan bahwa transisi telah dilewati sebanyak tiga kali
 - Jika header transisi tebal  menandakan bahwa transisi tersebut akan dituju
 - Place dilambangkan dengan 
 - Token dilambangkan dengan 
 - Busur dilambangkan dengan  menandakan penghubung antar transisi

Log Replay untuk Skenario

Tabel 4.11 menunjukkan *log replay* terjadinya aktivitas *Start.Good Receipt*. Sebelum *Start.Good Receipt* dimulai, model memproduksi 1 token sehingga nilai P menjadi 1 dan total P sama dengan 1.

Tabel 4.11 Log replay: Start.Good receipt

Alur:	Start.Good Receipt → Start.High Rack → Start.Picking → Start.Packing → Start.Loading		
			
M			0
R			0
C		0	0
P		1	1

Tabel 4.12 menunjukkan *log replay* terjadinya aktivitas *Start.High Rack*. Saat aktivitas terjadi, model memproduksi lagi 1 token dan mengonsumsi 1 token. Sehingga total nilai P menjadi 2 dan nilai C sama dengan 1.

Tabel 4.12 *Log replay: High Rack*

Alur:	Start.Good Receipt → Start.High Rack → Start.Picking → Start.Packing → Start.Loading			Σ
			M	0
			R	0
			C	1
			P	1
				2

Tabel 4.13 menunjukkan *log replay* aktivitas *start.picking*. Pada saat aktivitas terjadi, akan dikonsumsi 1 token dan model memproduksi lagi 1 token. Hal ini menyebabkan total nilai C sebesar 2 dan nilai P sama dengan 3.

Tabel 4.13 *Log replay: Start.Picking*

Alur:	Start.Good Receipt → Start.High Rack → Start.Picking → Start.Packing → Start.Loading			Σ
			M	0
			R	0
			C	1
			P	1
				2
				3

Tabel 4.14 menunjukkan *log replay* aktivitas *start.packing*. Pada saat aktivitas terjadi, dikonsumsi 1 token dan model memproduksi lagi 1 token. Hal ini menyebabkan total nilai C sebesar 3 dan nilai P sama dengan 4.

Tabel 4.14 *Log replay: Start.Packing*

Alur:	Start.Good Receipt → Start.High Rack → Start.Picking → Start.Packing → Start.Loading			Σ
			M	0

	R		0
	C	1	3
	P	1	4

Tabel 4.15 menunjukkan *log replay* aktivitas *start.loading*. Pada saat aktivitas terjadi, dikonsumsi 1 token dan model memproduksi lagi 1 token. Hal ini menyebabkan total nilai C sebesar 4 dan nilai P sama dengan 5.

Tabel 4.15 Log replay: Start.Loading

Alur:	Start.Good Receipt → Start.High Rack → Start.Picking → Start.Packing → Start.Loading		
			Σ
	M		0
	R		0
	C	1	4
	P	1	5

Setelah melewati aktivitas *start.loading*, model memproduksi kembali 1 token dan mengonsumsi 1 token. Sehingga total nilai C adalah 5 dan nilai P adalah 6. Pada skenario ini tidak terdapat token yang tersisa dan hilang sehingga nilai R dan M adalah 0.

Tabel 4.16 Log replay: End

Alur:	Start.Good Receipt → Start.High Rack → Start.Picking → Start.Packing → Start.Loading		
			Σ
	M		0
	R		0
	C	1	5
	P	1	6

Total dari *log replay* skenario diatas ditunjukkan pada Tabel 4.17.

Tabel 4.17 Rekapitulasi jumlah token skenario 1

\sum Token yang hilang	0
\sum Token tersisa	0
\sum Token yang dikonsumsi	5
\sum Token yang diproduksi	6

Dengan menggunakan rumus (8) dan data pada Tabel 4.17, nilai *fitness* untuk skenario 1 sebagai berikut:

$$f = \frac{1}{2} \left(1 - \frac{\sum_{i=1}^k n_i m_i}{\sum_{i=1}^k n_i c_i} \right) + \frac{1}{2} \left(1 - \frac{\sum_{i=1}^k n_i r_i}{\sum_{i=1}^k n_i p_i} \right)$$

$$f = \frac{1}{2} \left(1 - \frac{4096 * 0}{4096 * 5} \right) + \frac{1}{2} \left(1 - \frac{4096 * 0}{4096 * 6} \right)$$

$$f = \frac{1}{2} (1) + \frac{1}{2} (1)$$

$$f = \frac{1}{2} + \frac{1}{2}$$

$$f = 1$$

4.4.2.1.3. Keluaran Perhitungan *Fitness*

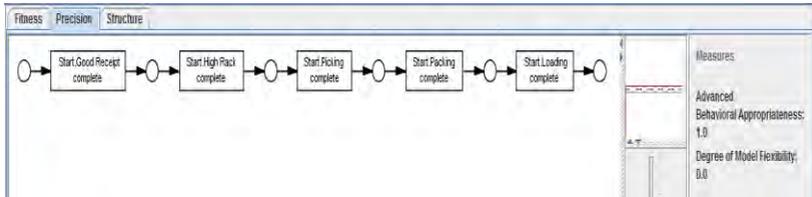
Berdasarkan hasil dari proses perhitungan *fitness*, didapatkan nilai *fitness* untuk model adalah 1. Hal ini menunjukkan bahwa kasus atau *case* yang terdapat dalam catatan kejadian sesuai dengan model proses yang dihasilkan.

4.4.2.2. Perhitungan Presisi

Perhitungan presisi dilakukan untuk mengetahui ketepatan model proses yang dihasilkan. Ketepatan yang dimaksud adalah seberapa besar kemungkinan muncul kasus yang tidak terdapat pada catatan kejadian. Pada model yang dihasilkan dapat dilihat bahwa terbentuk model yang berupa garis lurus. Hal ini menunjukkan bahwa model sudah sesuai dengan catatan kejadian dan tidak

membentuk “perilaku yang berlebihan”. Model dikatakan membentuk “perilaku yang berlebihan” apabila dalam model tersebut terdapat perilaku lebih atau sangat berbeda dibandingkan dengan yang terdapat pada catatan kejadian.

Berdasarkan hal tersebut, sudah dapat dipastikan bahwa model memiliki nilai presisi sama dengan 1.



Gambar 4.28 Nilai presisi model pada ProM

4.4.2.3. Perhitungan Struktural

Perhitungan struktural merupakan perhitungan yang menilai model proses berdasarkan jumlah *duplicate task* dan *redundant invisible task*. Semakin sedikit jumlah aktivitas duplikat dan aktivitas bayangan, maka semakin baik struktur model proses yang dihasilkan. Perhitungan struktural dapat dilakukan berdasarkan rumus (10) yang telah dijabarkan pada sub-bab 2.9.3.

4.4.2.3.1. Masukan Perhitungan Struktural

Masukan perhitungan struktural berupa variabel-variabel yang telah dijabarkan pada rumus (10), yaitu:

$|T|$: Kumpulan transisi dari model Petri Net

$|T_{DA}|$: Kumpulan *alternative duplicate task*

$|T_{IR}|$: Kumpulan *redundant invisible task*

Dari Gambar 4.26, diperoleh nilai untuk setiap variabel sebagai berikut:

$|T|$: 5

$|T_{DA}|$: 0

$|T_{IR}|$: 0

4.4.2.3.2. Proses Perhitungan Struktural

Dengan memasukkan nilai setiap variable ke dalam rumus (10), didapatkan nilai struktural sebagai berikut:

$$a'_s = \frac{|T| - (|T_{DA}| + |T_{IR}|)}{|T|}$$

$$a'_s = \frac{|5| - (|0| + |0|)}{|5|}$$

$$a'_s = \frac{5}{5}$$

$$a'_s = 1$$

4.4.2.3.3. Keluaran Perhitungan Struktural

Berdasarkan hasil dari proses perhitungan struktural, didapatkan nilai struktural model proses adalah 1. Hal ini menunjukkan bahwa struktur model proses baik dengan tidak adanya aktivitas duplikat dan aktivitas bayangan yang terjadi.

4.4.3. Keluaran Pengukuran Model

Setelah melakukan pengukuran performa model, didapatkan nilai-nilai seperti yang ditunjukkan pada Tabel 4.18.

Tabel 4.18 Ringkasan Hasil Pengukuran Performa Model

	<i>Fitness</i>	Presisi	Struktural
Model Proses	1	1	1

Dari hasil ringkasan tabel yang ditunjukkan diatas, didapatkan nilai *fitness* untuk model proses adalah 1. Hal ini menunjukkan bahwa model proses yang dihasilkan sudah sesuai atau menggambarkan catatan kejadian.

Dimensi pengukuran presisi untuk model proses adalah 1. Hal ini menunjukkan bahwa model proses yang dihasilkan presisi atau tepat dengan catatan kejadian.

Dimensi pengukuran struktural untuk model proses adalah 1. Hal ini menunjukkan bahwa struktur yang dihasilkan oleh model

proses tersebut sudah tepat atau baik sebab tidak ada aktivitas duplikat atau aktivitas bayangan yang terbentuk.

Halaman ini sengaja dikosongkan

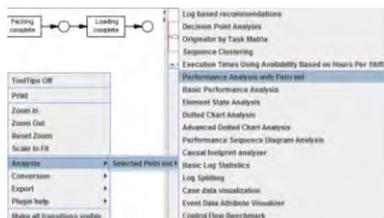
BAB V ANALISIS MODEL

Pada bab ini dijelaskan mengenai analisis dari model proses yang dihasilkan. Analisis tersebut meliputi analisis *bottleneck* dan analisis pengaruh volume dan variasi artikel terhadap *lead time* penyelesaian pengepakan.

5.1. Analisis *Bottleneck*

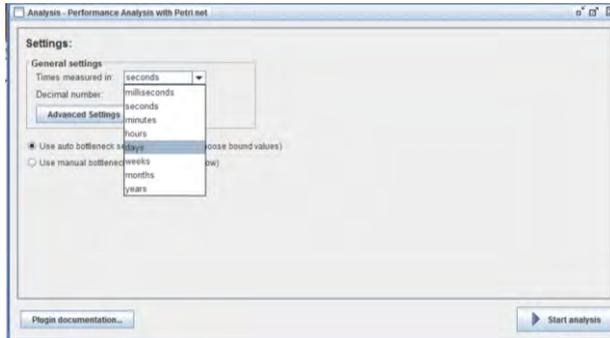
Analisis *bottleneck* dilakukan untuk mengetahui terjadinya waktu tunggu yang lama pada aktivitas untuk menuju ke aktivitas selanjutnya. Dengan demikian, diharapkan mampu membantu untuk mengetahui letak dan penyebab terjadinya waktu tunggu lama dalam proses pergerakan artikel di PDC. Untuk mengetahui terjadinya *bottleneck* pada model, dapat dilakukan dengan bantuan ProM dengan cara:

- Klik kanan pada model kemudian pilih Analysis → Selected Petri Net → Performance Analysis with Petri net, seperti yang ditunjukkan pada Gambar 5.1.



Gambar 5.1 Melakukan performance analysis with petri net

- Setelah itu akan muncul dialog box seperti yang ditunjukkan pada Gambar 5.2. Ubah time measure menjadi “days” agar waktu tunggu dihitung per hari. Kemudian tekan tombol Start Analysis



Gambar 5.2 Mengubah time measured

- Kemudian akan muncul hasil *bottleneck* seperti yang ditunjukkan Gambar 5.3.



Gambar 5.3 Hasil *bottleneck* model proses bisnis

Keterangan:

- Warna magenta : *bottleneck* dengan tingkat tinggi
- Warna kuning : *bottleneck* dengan tingkat medium
- Warna biru : *bottleneck* dengan tingkat rendah

Berdasarkan Gambar 5.3 diperoleh informasi *bottleneck* seperti yang ditunjukkan pada Tabel 5.1. Sebelum itu, pada Tabel 5.1 aktivitas pada model diinisialkan sebagai berikut:

- GR : Start.Good Receipt
- HR : Start.High Rack
- PI : Start.Picking
- PA : Start.Packing
- LD : Start.Loading

Tabel 5.1 Informasi *bottleneck*

Aktivitas	Waktu Tunggu			Waktu yang dilalui		
	Rata-Rata	Min	Maks	Rata-Rata	Min	Maks
GR→HR	13,92 jam	1,2 menit	2,28 hari	7,9 hari	13,92 jam	26,32 hari
HR→PI	6,91 hari	57,6 menit	25,19 hari			
PI→PA	6,72 jam	13,4 menit	2,73 hari			
PA→LD	3,36 jam	5,43 menit	22,32 jam			

Dari informasi *bottleneck* yang didapatkan, diketahui bahwa *bottleneck* terjadi pada aktivitas Start.High Rack ke Start.Picking. Lama waktu *bottleneck* adalah 6,91 hari. Setelah mengetahui *bottleneck* secara keseluruhan, kemudian dilakukan analisis *bottleneck* kembali untuk masing-masing *wave*. Analisis ini dilakukan untuk mengetahui apakah memang benar *bottleneck* yang terjadi terletak pada proses Start.High Rack ke Start.Picking. Analisis *bottleneck* ini dilakukan dengan bantuan ProM dengan cara seperti yang telah dijelaskan sebelumnya. Terdapat empat *wave* yang dianalisis, yaitu *wave* 21145, *wave* 21207, *wave* 21180, dan *wave* 21110. Hasil analisis *bottleneck* dari kedua *wave* ditunjukkan pada Tabel 5.2.

Dari analisis tersebut akan didapatkan dua informasi waktu, yaitu waktu tunggu dan waktu yang dilalui. Waktu tunggu merupakan lamanya waktu proses mulai suatu aktivitas dilakukan hingga aktivitas tersebut selesai dilakukan, misalnya mulai dari Start.Good Receipt hingga Start.High Rack. Sedangkan waktu yang dilalui merupakan lamanya waktu proses yang mulai dari dimulainya aktivitas paling awal hingga aktivitas paling akhir selesai dalam suatu model proses, misalnya mulai dari Start.Good Receipt hingga Start.Loading.

Perhitungan baik waktu tunggu maupun waktu yang dilalui dilakukan dengan melihat waktu tiap kasus. Untuk nilai maksimum, maka dari seluruh kasus yang ada dalam model proses dicari waktu

nilai maksimumnya. Begitu pula untuk nilai minimum dan rata-rata. Nilai minimum didapatkan dengan menghitung waktu paling minimum dari seluruh kasus yang ada dalam model proses, sedangkan rata-rata dihitung dengan merata-rata nilai seluruh kasus.

Tabel 5.2 Perbandingan *bottleneck* pada empat wave

Hasil <i>Bottleneck</i>	Aktivitas	Waktu Tunggu			Waktu yang dilalui		
		Maks	Min	Rata-Rata	Maks	Min	Rata-Rata
<p><i>Wave 21145</i></p>	GR→HR	1,21 hari	2,64 jam	13,92 jam	22,37 hari	14,16 jam	5,85 hari
	HR→PI	21,27 hari	57,6 menit	4,79 hari			
	PI→PA	2,73 hari	14,4 menit	8,4 jam			
	PA→LD	19,92 jam	8,15 menit	3,36 jam			
<p><i>Wave 21207</i></p>	GR→HR	2,28 hari	1,2 menit	13,68 jam	26,32 hari	13,92 jam	8,39 hari
	HR→PI	25,19 hari	2,64 jam	7,39 hari			
	PI→PA	1,11 hari	13,59 menit	6,72 jam			
	PA→LD	22,32 jam	5,43 menit	3,6 jam			

Hasil <i>Bottleneck</i>	Aktivitas	Waktu Tunggu			Waktu yang dilalui		
		Maks	Min	Rata-Rata	Maks	Min	Rata-Rata
<p><i>Wave 21110</i></p>	GR → HR	2,28 hari	1,2 jam	13,44 jam	20,64 hari	13,92 jam	10,36 hari
	HR → PI	19,47 hari	7,44 jam	9,58 hari			
	PI → PA	18,72 jam	28,8 menit	2,64 jam			
	PA → LD	9,36 jam	28,8 menit	2,64 jam			
<p><i>Wave 21180</i></p>	GR → HR	1,14 hari	1,92 jam	14,4 jam	25,15 hari	1,07 hari	7,57 hari
	HR → PI	24,19 hari	4,32 jam	6,57 hari			
	PI → PA	1,7 hari	1,2 jam	6,72 jam			
	PA → LD	17,28 jam	7,65 menit	3,12 jam			

Keterangan:

- Warna magenta : *bottleneck* dengan tingkat tinggi
- Warna kuning : *bottleneck* dengan tingkat medium
- Warna biru : *bottleneck* dengan tingkat rendah
- Bagian yang diberi warna magenta merupakan aktivitas dimana *bottleneck* terjadi

Berdasarkan Tabel 5.2 diperoleh informasi bahwa:

- *Bottleneck* pada masing-masing *wave* terjadi pada proses Start.High Rack menuju Start.Picking.
- *Bottleneck* paling lama terjadi pada *wave* 21110 dengan rata-rata 9,58 hari dari keseluruhan waktu tunggu dalam satu *wave*.
- *Bottleneck* ini terjadi karena sepatu menunggu datangnya pesanan dulu, setelah itu baru dilakukan *picking*.
- Rata-rata waktu sepatu diam di *high rack* berkisar antara 4 hingga 10 hari.

Berdasarkan Gambar 5.3 dan Tabel 5.2 diatas, sudah dapat dilihat bahwa *bottleneck* terjadi pada proses Start.High Rack menuju Start.Picking. Adanya *bottleneck* dari proses Start.High Rack menuju Start.Picking terjadi karena barang atau sepatu menunggu adanya *delivery order*. Seperti yang telah diketahui bahwa PT. XYZ merupakan perusahaan yang menerapkan mekanisme kerja *push* untuk *production* dan *pull* untuk *delivery*. Mekanisme *push* untuk *production* berarti dalam menjalankan proses produksi tergantung pada peramalan atau prognosis, sedangkan metode *pull* untuk *delivery* berarti dalam proses pengiriman barang tergantung pada permintaan pelanggan. Adanya metode *push* ini menyebabkan aktivitas *good receipt* dan *put in high rack* berdasarkan pada *production order* yang ada dalam prognosis. Prognosis atau peramalan adalah perkiraan jumlah produk yang akan diproduksi. Setelah itu, barang akan diam lama di *high rack* menunggu datangnya *delivery order (sales order)*. *Sales order* adalah pesanan produk oleh pelanggan.

Alur proses metode *push* dan *pull* dapat digambarkan misalnya pada *wave* 21110, pada tanggal 1 Maret 2014 pihak

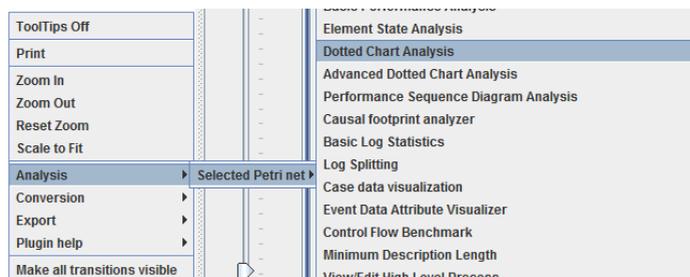
produksi telah memproduksi sejumlah artikel atau sepatu berdasarkan prognosis. Sepatu-sepatu yang telah diproduksi tersebut kemudian ditempatkan ke gudang dan dimasukkan ke dalam *high rack*. Sementara itu, pihak *production planning* melakukan prognosis lagi untuk pihak produksi agar memproduksi sejumlah sepatu. Sepatu-sepatu yang baru diproduksi ini kemudian ditempatkan lagi ke dalam gudang dan dimasukkan ke dalam *high rack*. Padahal sepatu yang diproduksi sebelumnya belum sepenuhnya dipesan. Sepatu ini akan tetap diam di *high rack* hingga datangnya *delivery order*. Pada tanggal 21 Maret, sepatu-sepatu yang telah dipesan mulai dilakukan pengepakan untuk kemudian dikirimkan ke pelanggan. Adanya proses menunggu ini menyebabkan proses dari *high rack* menuju *picking* menjadi lama, yaitu rata-rata 10,36 hari.

Dari penjelasan tersebut, dapat diketahui bahwa sebenarnya proses barang berdiam lama di *high rack* dapat berakibat pada menumpuknya barang di gudang. Dampak ke depannya biaya yang dikeluarkan PT. XYZ untuk menyimpan barang semakin tinggi. Selain itu, hal ini juga berdampak pada perputaran uang sehingga menjadi lambat dikarenakan barang tidak segera dijual.

5.2. Analisis Waktu Tunggu

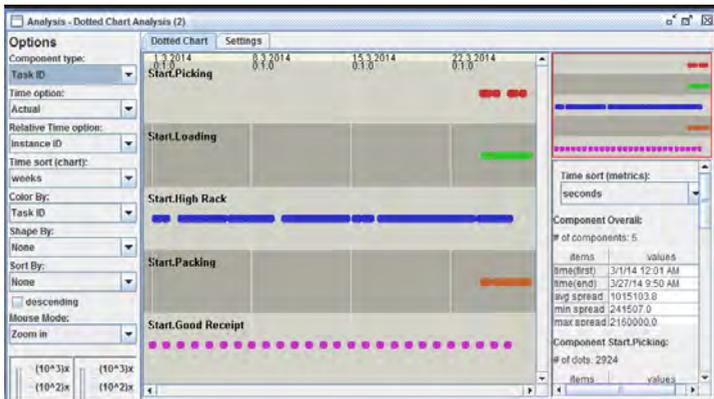
Analisis ini dilakukan untuk mengetahui lamanya waktu tunggu dari suatu aktivitas menuju aktivitas selanjutnya. Analisis ini dilakukan dengan melihat dotted chart yang ada pada ProM. Untuk menampilkan dotted chart, langkahnya sebagai berikut:

- Klik kanan pada model kemudian pilih Analysis → Selected Petri Net → Dotted Chart Analysis.



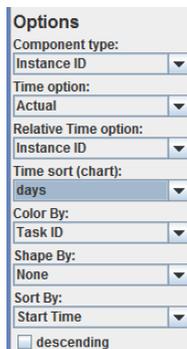
Gambar 5.4 Melakukan dotted chart analysis

- Kemudian akan muncul dotted chart seperti pada Gambar 5.5. Pada bagian *options*, lakukan pengaturan sesuai dengan yang diinginkan.



Gambar 5.5 Tampilan dotted chart analysis

- Pada analisis ini, akan dilihat pergerakan artikel tiap *instance* dan diurutkan berdasarkan waktu kedatangan tiap *instance* dalam satuan hari sehingga pengaturan *component type* diubah menjadi *instance ID*, *sort by Start Time*, dan *time sort (chart)* ke dalam *days*.



Gambar 5.6 Pengaturan dotted chart analysis

- Setelah itu, akan muncul dotted chart sesuai dengan data yang digunakan. Pada dotted chart tersebut, terdapat banyak titik dengan warna yang berbeda. Titik-titik tersebut menandakan *instance* dan warna tersebut menandakan aktivitas.

Set colors	
Start.Picking:	push to change
Start.Loading:	push to change
Start.High Rack:	push to change
Start.Packing:	push to change
Start.Good Receipt:	push to change

Gambar 5.7 Pengaturan warna komponen

Analisis dotted chart ini dilakukan pada empat *wave*, yaitu *wave* 21110, *wave* 21145, *wave* 21180, dan *wave* 21207. Perbandingan dotted chart kedua *wave* tersebut ditunjukkan pada Tabel 5.3.

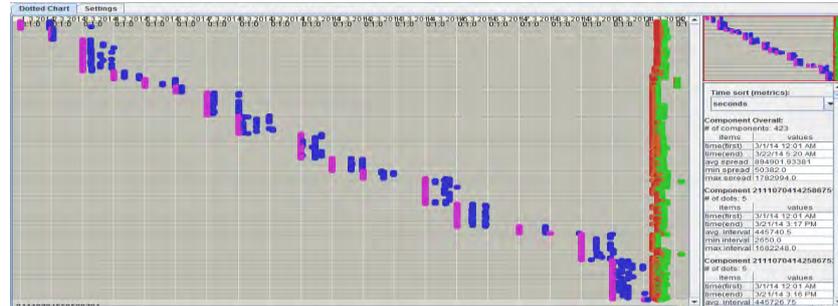
Berdasarkan Tabel 5.3 dapat dilihat bahwa artikel yang telah diterima di gudang PDC kemudian akan dimasukkan ke dalam *high rack*. Di *high rack*, artikel akan tetap diam hingga aktivitas *picking* mulai dilakukan. Dari keempat *wave* tersebut, diketahui bahwa rata-rata waktu tunggu Start.High Rack ke Start.Picking untuk *wave* 21145 adalah 4,79 hari, untuk *wave* 21207 adalah 7,39 hari, untuk *wave* 21180 adalah 6,57 hari, dan untuk *wave* 21110 adalah 9,58 hari. Adanya waktu tunggu yang lama ini dikarenakan mekanisme kerja perusahaan yang menerapkan *push* untuk *production* dan *pull* untuk *delivery*. Artikel-artikel yang telah diproduksi sesuai dengan peramalan kemudian akan diletakkan dalam gudang. Sementara satu artikel telah diletakkan pada gudang, kemudian akan diletakkan lagi suatu artikel yang baru diproduksi ke dalam gudang. Padahal, artikel yang sebelumnya sudah ada di gudang belum dikeluarkan karena menunggu datangnya *delivery order*. Dampaknya, artikel yang terdapat dalam gudang semakin banyak dan menumpuk.

Tabel 5.3 Perbandingan dotted chart dua wave

Dotted Chart	
<p>Wave 21145</p> 	<p>Rata-rata waktu tunggu Start.High Rack ke Start.Picking = 4,79 hari</p>
<p>Wave 21207</p> 	<p>Rata-rata waktu tunggu Start.High Rack ke Start.Picking = 7,39 hari</p>

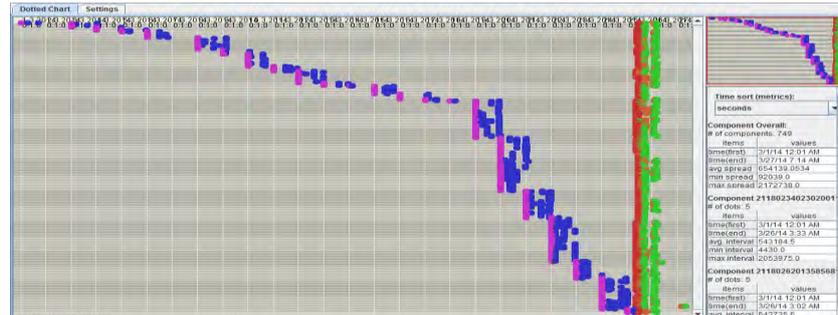
Dotted Chart

Wave 21110



Rata-rata waktu tunggu Start.High Rack ke Start.Picking = 9,58 hari

Wave 21180

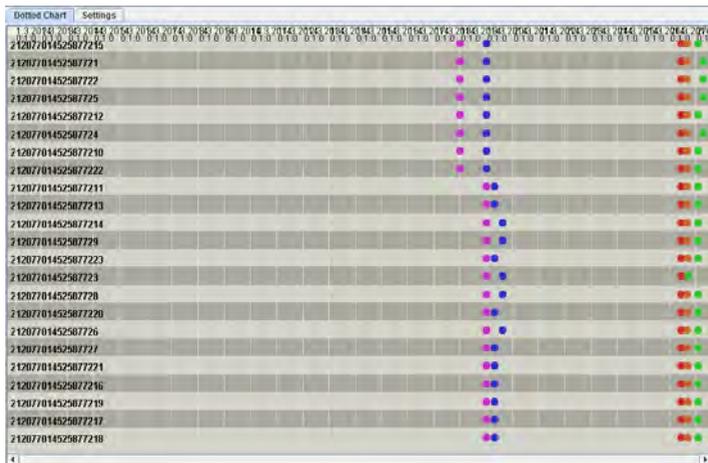


Rata-rata waktu tunggu Start.High Rack ke Start.Picking = 6,57 hari

Keterangan:

- Titik berwarna magenta : start.good receipt
- Titik berwarna biru : start.high rack
- Titik berwarna merah : start.picking
- Titik berwarna oranye : start.packing
- Titik berwarna hijau : start.loading
- Sumbu vertical (x) : *instance* atau sepatu
- Sumbu horizontal (y) : keterangan waktu (hari)

Selanjutnya dari dotted chart tersebut dilakukan analisis untuk tiap *instance* dalam 1 artikel. *Instance* dalam 1 artikel dapat pula diartikan sebagai sepatu. Adapun artikel yang dijadikan sampel dalam analisis ini adalah artikel 70145258772 pada *wave* 21207. Tampilan dotted chart ditunjukkan pada Gambar 5.8.



Gambar 5.8 Dotted chart artikel 70145258772

Keterangan:

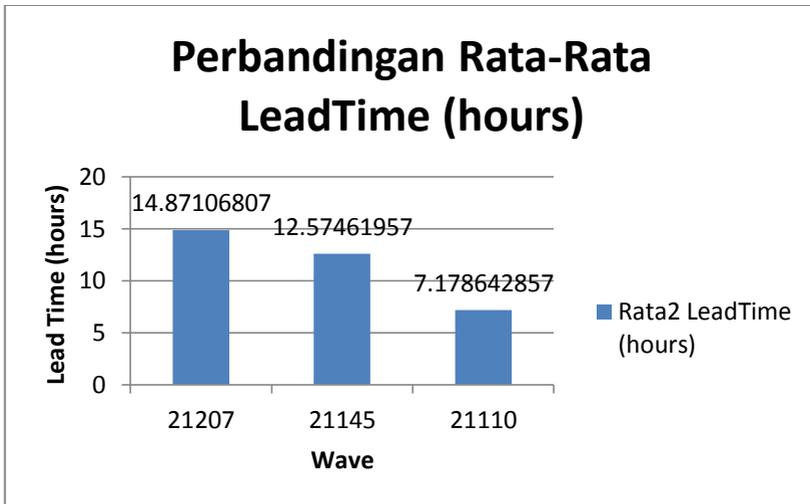
- Titik berwarna magenta : start.good receipt
- Titik berwarna biru : start.high rack
- Titik berwarna merah : start.picking
- Titik berwarna oranye : start.packing

- Titik berwarna hijau : start.loading
- Sumbu vertical (x) : *instance* atau sepatu
- Sumbu horizontal (y) : keterangan waktu (jam)

Berdasarkan Gambar 5.8 dapat diketahui bahwa meskipun suatu artikel telah diterima lebih dulu di gudang dan telah diletakkan ke dalam *high rack* lebih dulu, ternyata tidak menjamin bahwa artikel tersebut akan lebih dulu dikeluarkan. Hal ini tampak pada sepatu nomor 1 yang pada tanggal 19 Maret 2014 pukul 00:08:48 mulai masuk *high rack* sedangkan keluar dari *high rack* atau mulai di-*picking* pada tanggal 26 Maret 2014 pukul 10:24:30. Sementara itu, terdapat pula sepatu nomor 6 yang mulai masuk *high rack* pada tanggal 19 Maret 2014 pukul 14:35:07, sedangkan mulai di-*pick* pada tanggal 26 Maret 2014 pukul 10:24:38. Dari informasi tersebut diketahui bahwa artikel dengan sepatu nomor 1 meskipun lebih dahulu diletakkan dalam *high rack* pada kenyataannya waktu pengerjaan sepatu tersebut hampir bersamaan dengan sepatu nomor 6 yang waktu masuk *high rack*-nya lebih terakhir. Terjadinya hal ini dikarenakan tidak terdapatnya aturan yang jelas untuk urutan proses yang mana artikel yang lebih dulu masuk *high rack* seharusnya dikerjakan lebih dahulu. Akibatnya, artikel dengan sepatu nomor 1 harus menunggu di *high rack* lebih lama dibandingkan dengan sepatu nomor 6. Tidak adanya urutan ini dapat mengakibatkan barang terlupakan sehingga barang menjadi tertinggal di gudang.

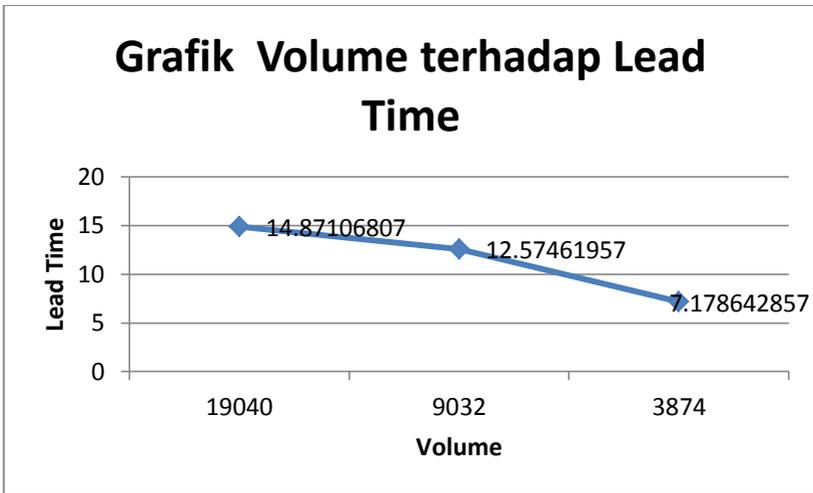
5.3. Analisis Variasi terhadap Lead Time

Analisis ini dilakukan untuk mengetahui pengaruh variasi artikel dalam satu *picking wave* terhadap *lead time* penyelesaian pengepakan. Grafik perbandingan *lead time* ditunjukkan pada Gambar 5.9.



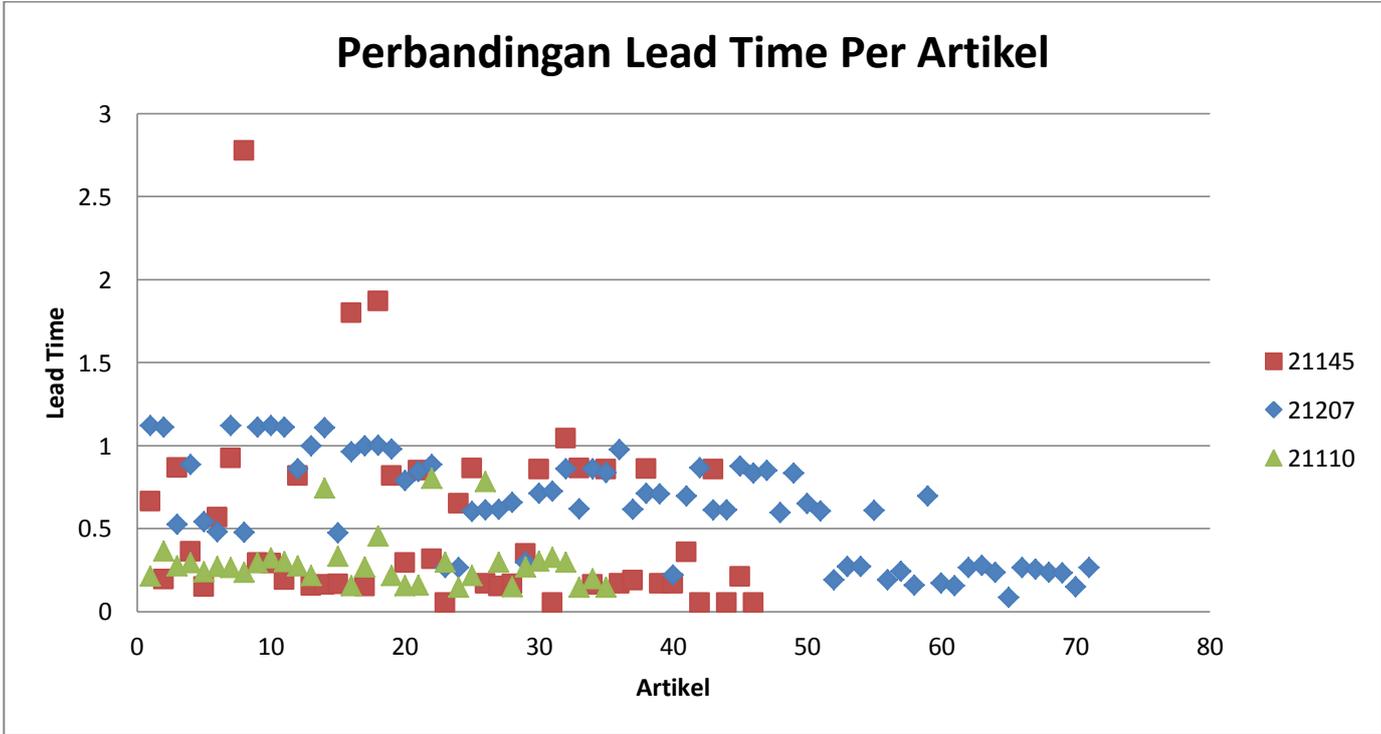
Gambar 5.9 Perbandingan rata-rata lead time

Dari Gambar 5.9 diketahui bahwa *wave* 21207 memiliki rata-rata *lead time* lebih panjang dibandingkan dengan *wave* 21145 dan *wave* 21110. Rata-rata *lead time* *wave* 21207 sebesar 14,87 jam, untuk *wave* 21145 rata-ratanya sebesar 12,57 jam, dan *wave* 21110 rata-ratanya sebesar 7,18 jam. Adanya perbedaan *lead time* penyelesaian pengepakan ini dipengaruhi oleh banyaknya variasi artikel dan volume artikel dalam satu *wave*. Pada *wave* 21207 terdapat 71 variasi artikel dengan total volume 19040 *pairs*, *wave* 21145 terdapat 46 variasi artikel dengan total volume 9032 *pairs*, dan *wave* 21110 terdapat 35 variasi artikel dengan total volume 3874 *pairs*.



Gambar 5.10 Grafik volume terhadap lead time

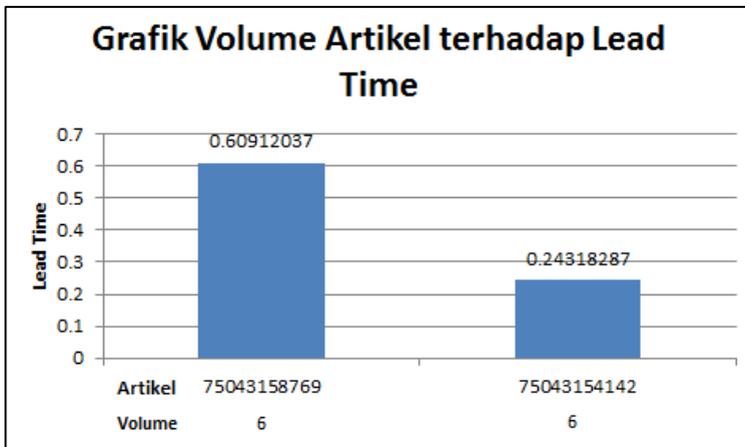
Banyaknya variasi dan volume artikel dalam satu *wave* menyebabkan semakin banyak pula proses *pick-pack* yang harus dilakukan. Jika dalam *wave* 21145 proses *pick-pack* hanya dilakukan sebanyak kurang lebih 46 kali, maka pada *wave* 21207 proses *pick-pack* dilakukan sebanyak kurang lebih 71 kali. Hal ini mengakibatkan bertambahnya waktu atau *lead time* penyelesaian pengepakan. Detail proses pengepakan dapat dilihat pada Gambar 1.1.



Gambar 5.11 Perbandingan lead time per artikel

Pada Gambar 5.11 dapat dilihat persebaran tiap artikel dalam satu *wave* terhadap *lead time* penyelesaian pengepakan dalam satuan hari. Titik berwarna biru menunjukkan persebaran artikel untuk *wave* 21207, titik berwarna merah menunjukkan persebaran artikel untuk *wave* 21145, dan titik berwarna hijau menunjukkan persebaran artikel untuk *wave* 21110. Dari persebaran tersebut tampak jelas bahwa *wave* 21207 memiliki variasi artikel lebih banyak dibandingkan dengan *wave* 21145 dan *wave* 21110.

Adanya kontribusi volume tiap artikel terhadap lamanya *lead time* penyelesaian pengepakan dalam satu *wave* ternyata tidak terlalu berpengaruh, seperti yang ditunjukkan pada Gambar 5.12.

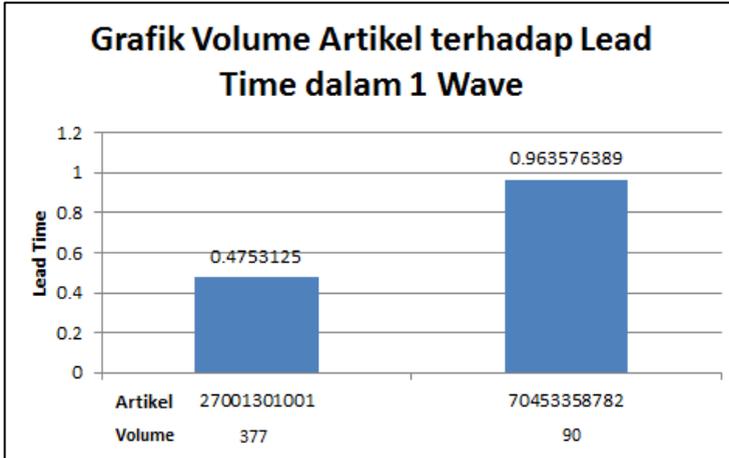


Gambar 5.12 Grafik volume terhadap lead time

Pada Gambar 5.12 diketahui terdapat dua artikel dalam satu *wave* yang masing-masing memiliki volume yang kecil. Artikel 75043158769 dengan volume 6 dan artikel 75043154142 dengan volume 6. Meskipun keduanya memiliki volume artikel yang kecil, pada kenyataannya keduanya memiliki *lead time* yang berbeda. Artikel 75043158769 lebih panjang *lead time*-nya dibandingkan dengan artikel 75043154142.

Selain itu, terdapat pula dalam satu *wave* artikel yang bervolume lebih sedikit tetapi *lead time* penyelesaiannya tidak lebih

cepat dibandingkan dengan artikel yang bervolume lebih banyak, seperti yang ditunjukkan pada Gambar 5.12.



Gambar 5.13 Grafik volume artikel terhadap lead time dalam 1 wave

Pada Gambar 5.13 diketahui terdapat dua artikel dari *wave* yang sama yaitu *wave* 21207. Terlihat bahwa artikel 27001301001 dengan volume 377 *pairs* memiliki *lead time* 0,47 hari atau 11,28 jam dan artikel 70453358782 dengan volume 90 *pairs* memiliki *lead time* 0,96 hari atau 23,4 jam. Meskipun artikel 70453358782 memiliki volume yang lebih sedikit, ternyata *lead time*-nya lebih panjang dibandingkan dengan artikel 27001301001.

Dari Gambar 5.12 dan Gambar 5.13, adanya kejadian ini disebabkan bisa saja pada saat pengepakan, terdapat beberapa *pairs* dari salah satu artikel yang tertinggal. Misalnya, beberapa *pairs* artikel 70453358782 tertinggal sehingga pengepakan untuk artikel 70453358782 belum selesai pada saat itu juga. Sementara itu proses pengepakan sudah masuk ke artikel selanjutnya. Dikarenakan *pairs* dari artikel 70453358782 yang tertinggal tersebut hanya berjumlah sedikit, maka pengerjaannya tidak terlalu menjadi prioritas sehingga proses pengerjaannya dilakukan diakhir. Hal ini mengakibatkan *lead time* pengepakan untuk artikel 70453358782 lebih panjang dibandingkan dengan artikel 27001301001.

a. Perbandingan Artikel dan Volume Sama

Pada bagian ini dilakukan perbandingan *lead time* artikel yang sama dari dua *wave*. Masing-masing artikel memiliki jumlah volume yang sama, yaitu 12. Adapun artikel yang digunakan adalah artikel 75045158768 dari *wave* 21145 dan *wave* 21180 seperti yang ditunjukkan pada Gambar 5.14.



Gambar 5.14 Perbandingan artikel dan volume sama

Pada Gambar 5.14 diketahui bahwa meskipun memiliki jenis artikel dan jumlah volume artikel yang sama, ternyata *lead time* yang dihasilkan artikel dari kedua *wave* tersebut berbeda. *Lead time* yang dihasilkan oleh artikel 75045158768 pada *wave* 21180 lebih lama dibandingkan dengan artikel 75045158768 pada *wave* 21145.

Hal ini dikarenakan pada *wave* 21145, artikel 75045158768 setelah diambil dari *high rack* kemudian langsung dilakukan pengepakan sehingga *lead time* untuk *wave* 21145 lebih pendek. Sedangkan pada *wave* 21180, artikel setelah diambil dari *high rack* tidak langsung dilakukan pengepakan. Artikel harus menunggu selama beberapa jam untuk dilakukan pengepakan. Adanya hal ini disebabkan penanganan yang berbeda untuk tiap

wave. Penanganan yang berbeda berarti dalam *wave* tersebut pembagian *shift* dan *M-Group* juga berbeda. Misalnya, pada salah satu *shift* ritme bekerjanya lebih cepat dibandingkan dengan *shift* lainnya sehingga hal ini mempengaruhi *lead time* artikel tersebut.

b. Perbandingan Artikel Beda dan Volume Sama

Pada bagian ini dilakukan perbandingan terhadap 4 artikel dari 4 *wave* yang mana artikel tersebut memiliki volume yang sama.



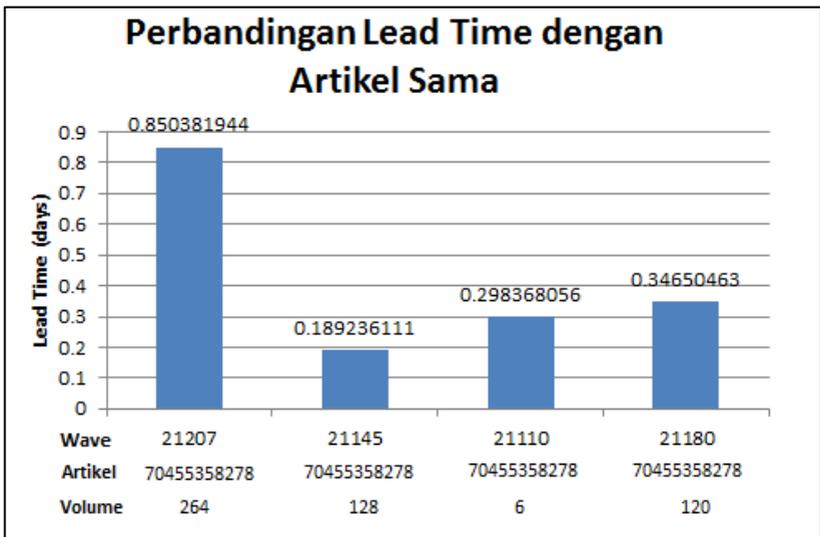
Gambar 5.15 Perbandingan artikel beda dan volume sama

Berdasarkan Gambar 5.15 diketahui bahwa meskipun artikel-artikel tersebut memiliki volume yang sama, namun *lead time* artikel tersebut tidaklah sama. Artikel pada *wave* 21110 memiliki *lead time* yang lebih panjang dibandingkan dengan *lead time* *wave* lainnya. Adanya perbedaan *lead time* untuk artikel yang bervolume sama dalam *wave* yang berbeda

disebabkan tidak adanya standar waktu pengepakan untuk artikel per volume sehingga suatu artikel dengan volume yang sama bisa dikerjakan lebih cepat dibandingkan dengan artikel lainnya.

c. Perbandingan Artikel Sama dan Volume Beda

Pada bagian ini dilakukan perbandingan untuk 4 *wave* yang mana memiliki artikel yang sama namun volume tiap artikel berbeda.



Gambar 5.16 Perbandingan artikel sama dan volume beda

Berdasarkan Gambar 5.16 diketahui bahwa dari keempat *wave* tersebut terdapat satu artikel yang sama, yaitu artikel 70455358278. Artikel tersebut memiliki volume yang berbeda untuk tiap *wave*-nya. Diketahui bahwa pada *wave* 21207, *wave* 21145, *wave* 21110, dan *wave* 21180 volumenya secara berurutan adalah 264, 128, 6, dan 120. Dari keempat *wave* tersebut, ternyata *wave* 21207 memiliki *lead time* yang paling panjang. Hal ini menunjukkan bahwa tidak terdapat pengaruh volume artikel terhadap *lead time* pengepakan. Artikel yang bervolume lebih kecil bisa jadi *lead time*-nya lebih panjang

dibandingkan dengan artikel yang bervolume lebih banyak. Adanya hal ini disebabkan penanganan tiap artikel dalam suatu *wave* yang berbeda. Penanganan ini meliputi pembagian *shift*, pegawai, ataupun *M-Group*.

Halaman ini sengaja dikosongkan

BAB VI KESIMPULAN DAN SARAN

Bab penutup ini berisi kesimpulan yang diperoleh dari hasil pengerjaan Tugas Akhir dan saran untuk pengembangan penelitian yang dilakukan dalam tugas akhir ini.

6.1. Kesimpulan

Beberapa kesimpulan yang diperoleh dari Tugas Akhir adalah sebagai berikut:

- a. Pembentukan catatan kejadian untuk proses bisnis *warehouse* dilakukan dengan mengekstraksi database melalui *query* pada tabel LTAP (Transfer Order Item) dan tabel LIPS (Sales and Distribution Document Delivery : item data). Untuk memunculkan data pada tabel maka dilakukan *query* dengan memfilter berdasarkan nomor *warehouse* dan tanggal *transfer order*.
- b. Pengukuran model untuk tiga dimensi, yaitu *fitness*, presisi, dan struktural dengan menggunakan Algoritma *Duplicate Genetic* menghasilkan nilai *fitness* 1, presisi 1, dan struktural 1. Hal ini menunjukkan bahwa Algoritma *Duplicate Genetic* mampu membentuk model proses sesuai dengan catatan kejadian yang ada dan tidak memunculkan aktivitas ganda atau bayangan. Dengan kata lain, pembentukan model menggunakan Algoritma *Duplicate Genetic* sudah menggambarkan proses bisnis PDC PT.XYZ yang sebenarnya.
- c. Berdasarkan analisis *bottleneck* diketahui bahwa *bottleneck* terjadi pada aktivitas menyimpan sepatu di *high rack*. Rata-rata lama *bottleneck* adalah 7,63 hari. *Bottleneck* pada aktivitas menyimpan sepatu di *high rack* disebabkan diterapkannya metode *push* untuk *production* dan *pull* untuk *delivery*. Metode *pull* untuk *delivery* pada Gambar 1.1 terdapat pada bagan yang diberi warna oranye, sedangkan metode *push* untuk *production* terjadi saat sepatu dimasukkan ke dalam *high rack*.

- d. Berdasarkan analisis variasi artikel terhadap *lead time* diketahui bahwa *wave* yang memiliki lebih banyak variasi artikel maka semakin panjang *lead time* penyelesaiannya. Panjangnya *lead time* ini disebabkan semakin banyaknya aktivitas atau proses yang harus dilakukan dalam satu *wave*.
- e. Besarnya volume tiap artikel tidak mempengaruhi *lead time* penyelesaian pengepakan artikel. Pada kenyataannya, panjang pendek *lead time* penyelesaian suatu artikel dapat disebabkan oleh penanganan artikel pada proses pengepakan.

6.2. Saran

Dari tugas akhir kali terdapat beberapa rekomendasi untuk pengembangan PT.XYZ dan saran yang diberikan terkait dengan pengembangan tugas akhir ini antara lain:

Rekomendasi

Rekomendasi ini ditujukan pada proses pengepakan seperti yang digambarkan di Gambar 1.1, khususnya bagan yang berwarna oranye. PT. XYZ sebaiknya dalam proses pengepakan, tidak hanya bertarget pada berapa banyak volume yang dapat di-*pack*. Namun juga seharusnya terdapat standar waktu pengepakan artikel per volume. Adanya standar waktu ini bisa meningkatkan kinerja pegawai dalam melakukan pengepakan.

Saran

Melakukan percobaan parameter Algoritma *Duplicate Genetic* dengan skenario yang bervariasi untuk lebih mengetahui pengaruh masing-masing parameter tersebut dalam pembentukan model proses.

DAFTAR PUSTAKA

- Aalst, W. M. (2011). *Process Mining Discovery, Conformance and Enhancement of Business Processes*. Heidelberg: Springer.
- Aalst, W., & et.al. (2012). Process Mining Manifesto. *BPM 2011 Workshops Proceedings* (pp. 169-194). Springer-Verlag.
- Aalst, W., Weijters, A., & Maruste, L. (2004). Workflow Mining Discovering Process Models from Event Logs. *IEEE Transactions on Knowledge and Data*, (pp. 16(9), 1128-1142).
- AG, S. (2001). *Warehouse Management Guide*.
- Andreswari, R. (2013). *Analisis kinerja algoritma penggalian proses untuk pemodelan proses bisnis perencanaan produksi dan pengadaan material pada PT.XYZ dengan kriteria control-flow*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Arsad, N. (2013). *Pembuatan Model Proses menggunakan Algoritma Heuristic Miner untuk Analisis Interaksi Proses Bisnis Perencanaan Produksi dan Pengadaan Material di PT.XYZ*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Erikson, H.-E., & Penker, M. (2000). *Business Modeling with UML: Business Pattern at Work*. New York: John Wiley & Sons.
- Hall, J. A. (2008). In *Accounting Information Systems*. South-Western College Pub.
- Medeiros, A. A., Weijters, A., & Aalst, W. v. (2005). Using Genetic Algorithms to Mine Process Models: Representation, Operators and Results.

- Medeiros, A. A., Weijters, A., & Aalst, W. v. (2005). Using Genetic Algorithms to Mine Process Models: Representation, Operators, dan Results.
- Medeiros, A. K. (2006). *Genetic Process Mining*. Eindhoven: Eindhoven University of Technology.
- Medeiros, A. K. (2006). *Genetic Process Mining*. Eindhoven: CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN.
- Murata, T. (1989). Petri Nets: Properties, Analysis, and Applications. *Proceedings of the IEEE*, (p. 77(4)).
- O'Brien, J. A. (2005). *Introduction to Information Systems*. Northern Arizona: Mc Graw-Hill.
- Pine, B. J. (1993). *Mass Customization: The New Frontier in Business Competition*. Boston: Harvard Business School Press.
- Porter, M. E. (1998). *Competitive Advantage: Creating and Sustaining Superior Performance*. New York: Free Press.
- Rozinat, A., & Aalst, W. v. (2009). Conformance Checking of Processes Based on Monitoring Real Behavior.
- Utami, R. A. (2013). *Pemodelan dan Analisis Bottleneck Proses Bisnis Perencanaan Produksi di PT.XYZ pada SAP dengan Algoritma Genetika*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Verbeek, H., Dongen, B. v., Mendling, J., & Aalst, W. v. (2006). Interoperability in the ProM Framework. *Proceedings of the CAiSE'06 Workshops and Doctoral Consortium* (pp. 619-630). Luxembourg: Presses Universitaires de Namur.

RIWAYAT PENULIS



Suviani Ningrum lahir di kota Makassar, Sulawesi Selatan pada tanggal 8 Maret 1992. Setelah menuntaskan pendidikan formalnya di SDN Sidokare 2 Sidoarjo, SMPN 1 Candi Sidoarjo, dan SMAN 1 Porong. Penulis lolos SNMPTN pada tahun 2010 untuk Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Semasa perkuliahan, Penulis tergabung sebagai anggota Himpunan Mahasiswa Sistem Informasi sebagai staff Kewirausahaan 2011-2012. Penulis juga sempat menjadi asisten praktikum mata kuliah Kalkulus dan Aljabar Linear dan Perencanaan Sumber Daya Perusahaan. Ketertarikan penulis terhadap bidang Penggalian Proses (*Process Mining*) membawanya untuk memilih bidang minat Sistem Pendukung Keputusan dan Intelegensia Bisnis di Jurusan Sistem Informasi serta membuat tugas akhir yang berkaitan dengan bidang minat ini.

Halaman ini sengaja dikosongkan.

LAMPIRAN A

PERBANDINGAN MODEL PARAMETER DEFAULT DAN PARAMETER PERCOBAAN

Tabel A.1 Perbandingan model parameter *default* dan parameter percobaan

Model	Parameter	<i>Fitness</i>	Presisi	Struktur
<p>Parameter Default</p>  <pre> graph LR A(()) --> B[Start Good Receipt complete] B --> C(()) C --> D[Start High Rack complete] D --> E(()) E --> F[Start Picking complete] F --> G(()) G --> H[Start Packing complete] H --> I(()) I --> J[Start Loading complete] J --> K(()) </pre>	<ul style="list-style-type: none"> • Population size : 100 • Int.pop.size : follow heuristic (<i>duplicate</i>+arcs) • Max.num.generations : 1000 • Seed : 1 • Power value : 1 • Elitism rate : 0,02 • <i>Fitness</i> type : ExtraBehavior • Selection method type : Tournament 5 • Crossover type : Duplicates Enhanced • Crossover rate : 0,8 	1	1	1

	<ul style="list-style-type: none"> • Mutation type : Enhanced • Mutation rate : 0,2 			
<p>Parameter Percobaan</p>  <pre> graph LR Start(()) --> GR[Start Good Receipt complete] GR <--> HR[Start High Rack complete] HR <--> PK[Start Picking complete] PK <--> PA[Start Packing complete] PA <--> LO[Start Loading complete] LO --> End(()) </pre>	<ul style="list-style-type: none"> • Population size : 100 • Int.pop.size : follow heuristic (<i>duplicate</i>+arcs) • Max.num.generations : 1000 • Seed : 20 • Power value : 13 • Elitism rate : 0,2 • <i>Fitness</i> type : ExtraBehavior • Selection method type : Tournament 5 • Crossover type : <i>Duplicate</i> Enhanced • Crossover rate : 0,8 • Mutation type : Enhanced • Mutation rate : 0,2 	<p>1</p>	<p>1</p>	<p>1</p>

LAMPIRAN B
DATA HASIL EKSTRAKSI

Tabel B.1 Hasil ekstraksi *file 21145*

Wave	Artikel	Tanggal	Waktu	Konfir- masi	Aktivitas
21145	75043158769	3/2/2014	0:01:00	1	Start.Good Receipt
21145	75043158769	3/3/2014	01:07:10	1	Start.High Rack
21145	70414258675	3/3/2014	06:46:15	1	Start.High Rack
21145	70414258675	3/3/2014	06:46:15	2	Start.High Rack
21145	70414258675	3/3/2014	06:46:15	3	Start.High Rack
21145	70414258675	3/3/2014	06:46:15	4	Start.High Rack
21145	70414258675	3/3/2014	06:46:15	5	Start.High Rack
21145	70414258675	3/3/2014	06:46:15	7	Start.High Rack
21145	70414258675	3/3/2014	06:46:15	8	Start.High Rack
21145	70414258675	3/3/2014	0:01:00	1	Start.Good Receipt
21145	70414258675	3/3/2014	0:01:00	2	Start.Good Receipt
21145	70414258675	3/3/2014	0:01:00	3	Start.Good Receipt
21145	70414258675	3/3/2014	0:01:00	4	Start.Good Receipt
21145	70414258675	3/3/2014	0:01:00	5	Start.Good Receipt
21145	70414258675	3/3/2014	0:01:00	7	Start.Good Receipt
21145	70414258675	3/3/2014	0:01:00	8	Start.Good Receipt
21145	21400354893	3/4/2014	13:56:04	1	Start.High Rack
21145	21400354893	3/4/2014	0:01:00	1	Start.Good Receipt
21145	70414258675	3/4/2014	0:01:00	6	Start.Good Receipt
21145	24074358576	3/4/2014	23:44:48	1	Start.High Rack
21145	24074358576	3/4/2014	23:44:48	2	Start.High Rack
21145	24074358576	3/4/2014	23:44:48	3	Start.High Rack
21145	24074358576	3/4/2014	23:44:48	4	Start.High Rack
21145	24074358576	3/4/2014	23:44:48	5	Start.High Rack

Wave	Artikel	Tanggal	Waktu	Konfir- masi	Aktivitas
21145	24074358576	3/4/2014	23:44:48	6	Start.High Rack
21145	24074358576	3/4/2014	0:01:00	1	Start.Good Receipt
21145	24074358576	3/4/2014	0:01:00	2	Start.Good Receipt
21145	24074358576	3/4/2014	0:01:00	3	Start.Good Receipt
21145	24074358576	3/4/2014	0:01:00	4	Start.Good Receipt

Tabel B.2 Hasil ekstraksi file 21207

Wave	Artikel	Tanggal	Waktu	Konfir- masi	Aktivitas
21207	07313158765	3/3/2014	0:01:00	1	Start.Good Receipt
21207	07313158765	3/3/2014	0:01:00	2	Start.Good Receipt
21207	07313158765	3/3/2014	0:01:00	3	Start.Good Receipt
21207	07313158765	3/3/2014	0:01:00	4	Start.Good Receipt
21207	07313158765	3/3/2014	0:01:00	5	Start.Good Receipt
21207	07313158765	3/3/2014	0:01:00	6	Start.Good Receipt
21207	07313158765	3/3/2014	15:07:04	1	Start.High Rack
21207	07313158765	3/3/2014	15:07:04	2	Start.High Rack
21207	07313158765	3/3/2014	15:07:04	3	Start.High Rack
21207	07313158765	3/3/2014	15:07:04	4	Start.High Rack
21207	07313158765	3/3/2014	15:07:04	5	Start.High Rack
21207	07313158765	3/3/2014	15:07:04	6	Start.High Rack
21207	07313158765	3/27/2014	00:52:48	1	Start.Picking
21207	07313158765	3/27/2014	00:52:48	2	Start.Picking
21207	07313158765	3/27/2014	00:52:48	3	Start.Picking
21207	07313158765	3/27/2014	00:52:48	4	Start.Picking
21207	07313158765	3/27/2014	00:52:48	5	Start.Picking
21207	07313158765	3/27/2014	00:52:48	6	Start.Picking
21207	23370302072	3/1/2014	0:01:00	22	Start.Good Receipt

Wave	Artikel	Tanggal	Waktu	Konfir- masi	Aktivitas
21207	24266301218	3/1/2014	0:01:00	1	Start.Good Receipt
21207	24266301218	3/1/2014	0:01:00	2	Start.Good Receipt
21207	24266301218	3/1/2014	0:01:00	3	Start.Good Receipt
21207	24266301218	3/1/2014	0:01:00	4	Start.Good Receipt
21207	24266301218	3/1/2014	0:01:00	5	Start.Good Receipt
21207	07313158765	3/27/2014	04:51:12	1	Start.Packing
21207	07313158765	3/27/2014	04:51:22	2	Start.Packing
21207	07313158765	3/27/2014	04:51:45	3	Start.Packing
21207	07313158765	3/27/2014	04:52:00	4	Start.Packing
21207	07313158765	3/27/2014	04:52:28	5	Start.Packing

Tabel B.3 Hasil ekstraksi file 21110

Wave	Artikel	Tanggal	Waktu	Konfir- masi	Aktivitas
21110	70414258675	3/1/2014	22:44:46	1	Start.High Rack
21110	70414258675	3/1/2014	22:44:46	2	Start.High Rack
21110	70414258675	3/1/2014	22:44:46	3	Start.High Rack
21110	70414258675	3/1/2014	22:44:46	4	Start.High Rack
21110	70414258675	3/1/2014	22:44:46	5	Start.High Rack
21110	70414258675	3/1/2014	22:44:46	6	Start.High Rack
21110	70414258675	3/1/2014	22:44:46	8	Start.High Rack
21110	70414258675	3/1/2014	22:44:46	9	Start.High Rack
21110	70414258675	3/1/2014	22:44:46	10	Start.High Rack
21110	70414258675	3/1/2014	22:44:46	11	Start.High Rack
21110	70414258675	3/1/2014	0:01:00	1	Start.Good Receipt
21110	70414258675	3/1/2014	0:01:00	2	Start.Good Receipt
21110	70414258675	3/1/2014	0:01:00	3	Start.Good Receipt
21110	70414258675	3/1/2014	0:01:00	4	Start.Good Receipt

Wave	Artikel	Tanggal	Waktu	Konfirmasi	Aktivitas
21110	70414258675	3/1/2014	0:01:00	5	Start.Good Receipt
21110	70414258675	3/1/2014	0:01:00	6	Start.Good Receipt
21110	70414258675	3/1/2014	0:01:00	7	Start.Good Receipt
21110	70414258675	3/1/2014	0:01:00	8	Start.Good Receipt
21110	70414258675	3/1/2014	0:01:00	9	Start.Good Receipt
21110	70414258675	3/1/2014	0:01:00	10	Start.Good Receipt
21110	70414258675	3/1/2014	0:01:00	11	Start.Good Receipt
21110	75043158769	3/2/2014	01:28:08	1	Start.High Rack
21110	75043158769	3/2/2014	01:28:08	2	Start.High Rack
21110	75043158769	3/2/2014	01:28:08	3	Start.High Rack
21110	75043158769	3/2/2014	01:28:08	4	Start.High Rack
21110	75043158769	3/2/2014	01:28:08	5	Start.High Rack
21110	75043158769	3/2/2014	01:28:08	6	Start.High Rack
21110	75043158769	3/2/2014	01:28:08	7	Start.High Rack
21110	75043158769	3/2/2014	01:28:08	8	Start.High Rack

Tabel B.4 Hasil ekstraksi file 21180

Wave	Artikel	Tanggal	Waktu	Konfirmasi	Aktivitas
21180	07313158765	3/25/2014	10:58:26	1	Start.Picking
21180	07313158765	3/25/2014	10:58:26	2	Start.Picking
21180	07313158765	3/26/2014	02:50:27	1	Start.Packing
21180	07313158765	3/26/2014	02:53:54	2	Start.Packing
21180	07313158765	3/26/2014	03:55:48	1	Start.Loading
21180	07313158765	3/26/2014	04:28:39	2	Start.Loading
21180	07313158765	3/3/2014	15:07:04	1	Start.High Rack
21180	07313158765	3/3/2014	15:07:04	2	Start.High Rack
21180	07313158765	3/3/2014	0:01:00	1	Start.Good Receipt

Wave	Artikel	Tanggal	Waktu	Konfir- masi	Aktivitas
21180	07313158765	3/3/2014	0:01:00	2	Start.Good Receipt
21180	07313158800	3/25/2014	10:58:27	1	Start.Picking
21180	07313158800	3/25/2014	10:58:29	2	Start.Picking
21180	07313158800	3/26/2014	03:07:31	1	Start.Packing
21180	07313158800	3/26/2014	03:09:36	2	Start.Packing
21180	07313158800	3/26/2014	03:56:07	1	Start.Loading
21180	07313158800	3/26/2014	04:28:35	2	Start.Loading
21180	07313158800	3/8/2014	15:37:04	1	Start.High Rack
21180	07313158800	3/10/2014	02:52:56	2	Start.High Rack
21180	07313158800	3/8/2014	0:01:00	1	Start.Good Receipt
21180	07313158800	3/10/2014	0:01:00	2	Start.Good Receipt
21180	21020301001	3/25/2014	10:19:49	24	Start.Picking
21180	21020301001	3/25/2014	10:19:49	25	Start.Picking
21180	21020301001	3/25/2014	10:19:55	8	Start.Picking
21180	21020301001	3/25/2014	10:19:55	9	Start.Picking
21180	21020301001	3/25/2014	10:19:55	10	Start.Picking
21180	21020301001	3/25/2014	10:19:55	11	Start.Picking
21180	21020301001	3/25/2014	10:19:55	12	Start.Picking
21180	21020301001	3/25/2014	10:19:59	1	Start.Picking
21180	21020301001	3/25/2014	10:19:59	2	Start.Picking

B-6

Halaman ini sengaja dikosongkan

LAMPIRAN C
DATA CATATAN KEJADIAN

Tabel C.1 Catatan kejadian file EventLog

id kasus	keterangan waktu	aktivitas
212072337030207222	3/1/2014 0:01:00	Start.Good Receipt
21207242663012181	3/1/2014 0:01:00	Start.Good Receipt
21207242663012182	3/1/2014 0:01:00	Start.Good Receipt
21207242663012183	3/1/2014 0:01:00	Start.Good Receipt
21207242663012184	3/1/2014 0:01:00	Start.Good Receipt
21207242663012185	3/1/2014 0:01:00	Start.Good Receipt
21207242663012186	3/1/2014 0:01:00	Start.Good Receipt
21207242663012187	3/1/2014 0:01:00	Start.Good Receipt
21207242663012188	3/1/2014 0:01:00	Start.Good Receipt
21207242663012189	3/1/2014 0:01:00	Start.Good Receipt
212072426630121811	3/1/2014 0:01:00	Start.Good Receipt
212072426630121812	3/1/2014 0:01:00	Start.Good Receipt
212072426630121813	3/1/2014 0:01:00	Start.Good Receipt
212072426630121814	3/1/2014 0:01:00	Start.Good Receipt
212072426630121815	3/1/2014 0:01:00	Start.Good Receipt
212072426630121816	3/1/2014 0:01:00	Start.Good Receipt
212072426630121817	3/1/2014 0:01:00	Start.Good Receipt
21207262013585633	3/1/2014 0:01:00	Start.Good Receipt
21207262013585634	3/1/2014 0:01:00	Start.Good Receipt
21207262013585681	3/1/2014 0:01:00	Start.Good Receipt
21110704142586751	3/1/2014 0:01:00	Start.Good Receipt
21110704142586752	3/1/2014 0:01:00	Start.Good Receipt
21110704142586753	3/1/2014 0:01:00	Start.Good Receipt
21110704142586754	3/1/2014 0:01:00	Start.Good Receipt

id kasus	keterangan waktu	aktivitas
21110704142586755	3/1/2014 0:01:00	Start.Good Receipt
21110704142586756	3/1/2014 0:01:00	Start.Good Receipt
21110704142586757	3/1/2014 0:01:00	Start.Good Receipt
21110704142586758	3/1/2014 0:01:00	Start.Good Receipt
21110704142586759	3/1/2014 0:01:00	Start.Good Receipt
211107041425867510	3/1/2014 0:01:00	Start.Good Receipt
211107041425867511	3/1/2014 0:01:00	Start.Good Receipt
21207242663012181	3/1/2014 6:27:38	Start.High Rack
21207242663012181	3/1/2014 7:16:54	Start.High Rack
21207242663012182	3/1/2014 7:16:54	Start.High Rack
21207242663012183	3/1/2014 7:16:54	Start.High Rack
21207242663012184	3/1/2014 7:16:54	Start.High Rack
21207242663012185	3/1/2014 7:16:54	Start.High Rack
21207242663012186	3/1/2014 7:16:54	Start.High Rack
21207242663012187	3/1/2014 7:16:54	Start.High Rack
21207242663012188	3/1/2014 7:16:54	Start.High Rack
212072426630121812	3/1/2014 7:16:54	Start.High Rack
212072426630121813	3/1/2014 7:16:54	Start.High Rack
212072426630121814	3/1/2014 7:16:54	Start.High Rack
212072426630121815	3/1/2014 7:16:54	Start.High Rack
212072426630121816	3/1/2014 7:16:54	Start.High Rack
212072426630121817	3/1/2014 7:16:54	Start.High Rack
21207242663012189	3/1/2014 11:33:02	Start.High Rack
212072426630121811	3/1/2014 11:33:02	Start.High Rack
21110704142586751	3/1/2014 22:44:46	Start.High Rack