

USULAN TUGAS AKHIR

1. IDENTITAS PENGUSUL

NAMA : GALANG AMANDA DWI PAMUNGKAS
NRP : 5110100205
DOSEN WALI : Prof. Ir. Handayani Tjandrasa, M.Sc, Ph.D
DOSEN PEMBIMBING : 1. Prof. Ir. Drs, Ec. Riyanarto Sarno, M.Sc., PhD.
2. Abdul Munif, S.Kom., M.Sc.

2. JUDUL TUGAS AKHIR

“Rancang Bangun Aplikasi Pengonversi Model *Petri Net* Menjadi BPMN dan BPEL Untuk Implementasi *Web Services* Pada *Enterprise Resource Planning* (ERP)”

3. LATAR BELAKANG

Saat ini riset yang berkaitan tentang *Process Aware Information System* (PAIS) [1] sedang banyak dikembangkan. Beberapa diantaranya telah diimplementasikan dan digunakan oleh berbagai macam pihak baik itu perusahaan, industri, maupun organisasi lainnya yang memanfaatkan sistem yang berorientasi pada proses bisnis. Contoh sistem yang berorientasi pada proses bisnis adalah sistem manajemen alur kerja/*workflow* dan manajemen proses bisnis yang dibuat untuk mendukung proses operasional yang biasa disebut sebagai alur kerja. *Enterprise Resource Planning* (ERP) [2] merupakan salah satu contoh penerapan PAIS yang telah banyak dimanfaatkan oleh berbagai perusahaan.

Secara umum terdapat dua buah ruang lingkup dalam PAIS, yaitu: lingkup analisis desain dan implementasi. Lingkup analisis lebih mengarahkan pada bagaimana membuat sebuah proses menjadi lebih baik, salah satu contohnya adalah dengan melakukan simulasi penggalian proses/*process mining* dari sebuah rekaman

kejadian/*event log* untuk menghasilkan sebuah model proses yang kemudian digunakan untuk proses analisis. Lingkup implementasi lebih mengarahkan bagaimana memodelkan sebuah proses bisnis untuk kemudian dapat dijalankan dan diimplementasikan. Saat ini, implementasi proses dapat direalisasikan dengan menggunakan *Service Oriented Architecture* (SOA) [3], dimana sebuah proses direpresentasikan sebagai layanan-layanan yang berkesinambungan sehingga muncul *Web Services Business Process Execution Language* (WS-BPEL) [4] atau lebih disingkat BPEL sebagai standar dalam konteks implementasi proses. Kedua lingkup tersebut berjalan secara terpisah yang menyebabkan saat ini proses analisis desain sering ditinggalkan atau proses analisis hanya akan menghasilkan sebuah hasil analisis yang kemudian tidak diimplementasikan. Hal tersebut mendasari bahwa diperlukan adanya sebuah proses yang dapat menjembatani proses antar keduanya.

Karena berjalan secara terpisah, kedua lingkup juga memiliki standar yang berbeda untuk merepresentasi pemodelan prosesnya masing-masing. Pada umumnya, model proses yang dihasilkan akan direpresentasikan dengan *Petri Net Markup Language* (PNML) [5] yang merupakan sintaks berbasis *Extensible Markup Language* (XML) [6] yang menjadi standar pemodelan sebuah *Petri Net* pada lingkup analisis desain dan BPEL yang juga diseriliasasi dengan menggunakan XML pada lingkup implementasi. Untuk mengatasi hal tersebut perlu dibuat sebuah aplikasi yang mampu menangani adanya sebuah transformasi model, dalam hal ini adalah PNML menjadi BPEL yaitu dengan melakukan pemetaan dan penyesuaian untuk setiap elemen. Namun selain itu, penting bagi pihak pelaku bisnis untuk memahami proses yang bersangkutan mengingat *Petri Net* merupakan pemodelan yang berada pada level yang relatif rendah karena hanya memodelkan alur bagaimana sebuah proses itu berjalan dan kurang menggambarkan proses sebenarnya. Di sisi lain BPEL tidak memiliki standar notasi grafis. Untuk menjembatani celah tersebut, digunakan *Business Process Model and Notation* (BPMN) [7] yang merupakan standar dari pemodelan proses bisnis.

Solusi yang ditawarkan adalah dengan membuat aplikasi yang dapat memodelkan *Petri Net* menjadi model proses bisnis yang standar dan sesuai dengan proses sesungguhnya dalam bentuk BPMN sebelum dipetakan menjadi BPEL tanpa mengubah alur proses. Pengguna aplikasi dapat memodelkan BPMN dengan melakukan konfigurasi pada model PNML sesuai dengan proses bisnis sesungguhnya untuk kemudian dipetakan menjadi BPEL yang dapat di eksekusi. Tujuan pembuatan aplikasi ini utamanya adalah dapat menjembatani antara lingkup analisis desain dan implementasi. Selain itu dengan adanya pemodelan ke BPMN terlebih dahulu, alur proses dapat disesuaikan terlebih dahulu dengan proses bisnis sesungguhnya sebelum kemudian dipetakan menjadi BPEL untuk dapat dieksekusi tanpa sehingga proses yang dieksekusi nantinya adalah sesuai dengan proses bisnis sesungguhnya.

4. RUMUSAN MASALAH

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana memodelkan model proses *Petri Net* hasil dari analisis yang dapat merepresentasikan proses bisnis sesungguhnya?
2. Bagaimana memetakan sebuah model proses menjadi model proses yang dapat dieksekusi?

5. BATASAN MASALAH

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, diantaranya sebagai berikut:

1. BPMN yang digunakan dan dihasilkan adalah BPMN 2.0.
2. BPEL yang dihasilkan adalah BPEL 2.0.
3. Bahasa pemrograman yang digunakan adalah Java.
4. Aplikasi yang dikembangkan merupakan *plugin* ProM.
5. Data masukan berupa PNML yang menghasilkan BPMN dan BPEL.
6. Hasil BPEL akan divalidasi dengan menggunakan Eclipse BPEL Plugin.

6. TUJUAN PEMBUATAN TUGAS AKHIR

Tujuan dari tugas akhir ini adalah sebagai berikut:

1. Membuat *plugin* ProM yang dapat memetakan setiap elemen pada PNML menjadi elemen-elemen BPMN yang berkaitan dengan konfigurasi otomatis atau manual pada pemetaan setiap elemen.
2. Membuat *plugin* ProM yang dapat memetakan BPMN menjadi model BPEL yang dapat diimplementasikan.

7. MANFAAT TUGAS AKHIR

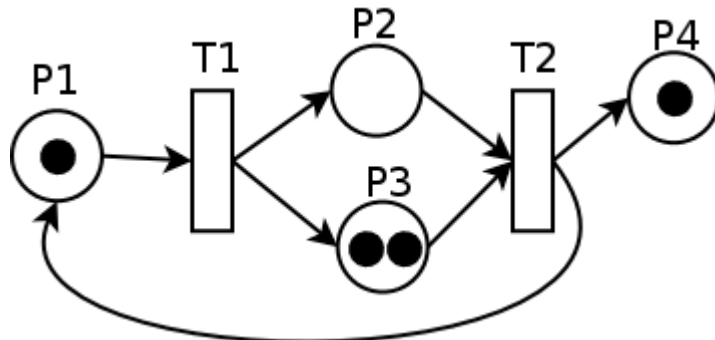
Manfaat yang diharapkan dari tugas akhir ini adalah:

1. Dapat mengubah hasil analisis menjadi model yang sesuai dengan standar pemodelan proses bisnis sehingga dapat merepresentasikan proses bisnis sesungguhnya.
2. Dapat mengimplementasikan model dari hasil analisis sebagai layanan-layanan web.
3. Menjembatani proses analisis dan desain dalam ERP.

8. TINJAUAN PUSTAKA

a. *Petri Net*

Merupakan sebuah bahasa pemodelan matematis yang memberikan gambaran umum untuk mendeskripsikan sebuah perilaku. *Petri Net* terdiri dari elemen *places* dan *transition* yang saling dihubungkan dengan *arcs*. *Petri Net* dideskripsikan sebagai *Petri Net Markup Language* (PNML) sebagai standar untuk pemodelan sebuah *Petri Net*. Gambar 1 merupakan gambar dari *Petri Net* dimana notasi P melambangkan elemen *places* dan notasi T melambangkan elemen *transition*.



Gambar 1. *Petri Net*

b. Business Process Model and Notation (BPMN)

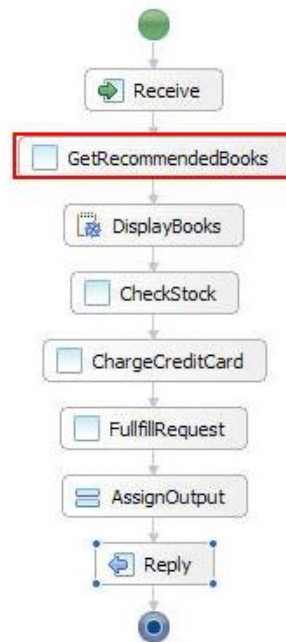
Sebuah representasi grafis untuk menspesifikasi proses bisnis. BPMN memiliki notasi yang dapat mewakili berbagai jenis aplikasi pemodelan proses bisnis sehingga banyak perusahaan yang menggunakan BPMN sebagai standar dalam teknik pemodelan proses bisnis. Gambar 2 menunjukkan notasi-notasi yang terdapat pada BPMN.

Start Intermediate End	Task Process/ Sub-process	XOR OR Complex XOR AND Event-based	Sequence flow Message flow Association	Pool Lane	Data Object Group Text Annotation
Flow Objects			Connectivity Objects	Swimlanes	Artifacts

Gambar 2. Notasi BPMN

c. Web Services Business Process Execution Language (WS-BPEL)

Sebuah standar bahasa yang dapat eksekusi untuk menspesifikasi aktivitas di dalam sebuah proses bisnis dengan layanan web/*web service*. WS-BPEL atau lebih disingkat BPEL merupakan bahasa orkestrasi [8]. BPEL fokus kepada proses bisnis modern yang mengadopsi layanan web sebagai mekanisme komunikasi eksternal sehingga dapat mengirim dan menerima pesan. Gambar 3 merupakan gambar dari BPEL.













Gambar 3. BPEL

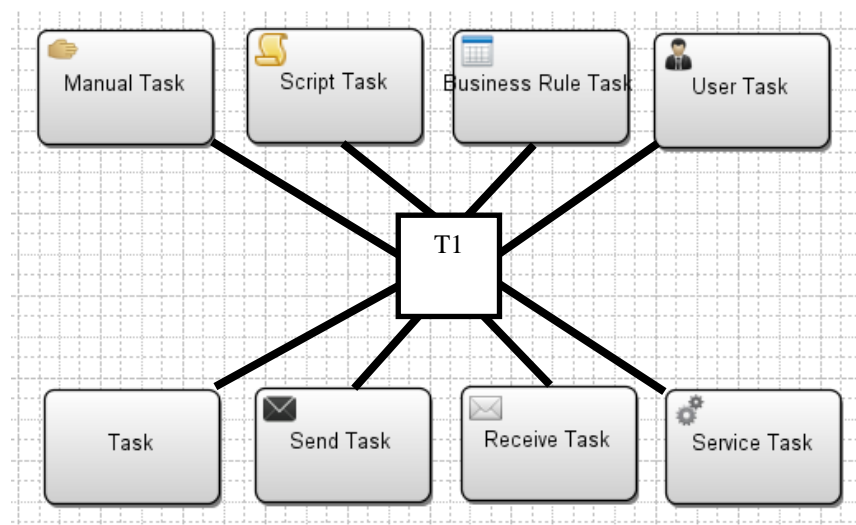
d. Teknik pemetaan elemen pada PNML ke BPMN

Pemetaan model PNML ke BPMN dapat dilakukan dengan metode pemrosesan elemen/*string processing* karena keduanya diserialisasi dengan menggunakan XML. Namun karena BPMN memiliki level yang relatif lebih tinggi dan kompleks jika dibandingkan dengan model PNML, diperlukan beberapa penyesuaian dan konfigurasi untuk menghindari adanya kehilangan informasi/*information loss*.

Tabel 1. Pemetaan elemen-elemen antara BPMN dan Petri Net

BPMN elements	Petri net elements
 Start Event	 Emitter connected to place
 End Event	 Place connected to Collector
 Task	 Transition
 XOR Gateway	 XOR
 AND Gateway	 Transition

Tabel 1 menunjukkan pemetaan setiap elemen *Petri Net* dapat berkorelasi langsung dengan setiap elemen pada BPMN. Hal ini tidak berlaku sebaliknya karena BPMN memiliki elemen-elemen yang tidak dimiliki oleh *Petri Net* seperti *pool* dan *lane* [7] sehingga notasi XML perlu diperluas sesuai dengan elemen yang ditambahkan.



Gambar 4. Pemetaan elemen *transition* menjadi berbagai tipe elemen *task*

Tanpa dilakukan adanya konfigurasi dan penyesuaian, elemen *transition* [5] akan langsung dipetakan menjadi sebuah elemen *task* [7]. Namun kebanyakan proses bisnis menggambarkan secara detail *task* apa yang dilakukan sehingga diperlukan adanya fitur konfigurasi untuk menangani masalah tersebut. Gambar 4 menunjukkan beberapa macam pilihan konfigurasi elemen *task* untuk setiap elemen *transition* yang akan dipetakan.

Setiap elemen *task* memiliki notasi XML yang berbeda, sehingga diperlukan perlakuan khusus dalam proses pengolahan elemen XML sesuai dengan konfigurasi yang telah dilakukan sebelumnya. Setiap elemen *transition* akan diubah menjadi sebuah elemen *task* dengan memperhatikan pilihan konfigurasi. Hal tersebut juga berlaku dalam pemetaan elemen-elemen yang lain.



Gambar 5. Struktur XML dari elemen *transition* dan elemen *task*

e. Transformasi model BPMN ke BPEL

Secara umum terdapat beberapa hal yang harus diperhatikan dalam pemetaan BPMN ke BPEL [9], yaitu:

1. Setiap *task* akan dipetakan langsung.
2. *Start events* ditransformasikan menjadi *receive activities*.
3. *End events* ditransformasikan menjadi *invoke activities*.
4. *Gateways* ditransformasikan menjadi *switch*.
5. *Swimlanes* tidak dipetakan.
6. Data-data bisnis dalam BPMN dipetakan menjadi variabel di BPEL.

Tabel 2. Pemetaan task

BPMN Object/Attribute	BPEL Element/Attribute
Task (TaskType: None)	None, but assignment properties will create a mapping to an assign . Otherwise a BPEL empty element would have been created.
Name = "Data Map"	None.
Assignment	assign . The name attribute is automatically generated by the tool creating the BPEL document.
From = input.carCompany	within a copy element, paired the next row from part="carCompany" variable="input"
To = carReservationRequest. company	within a copy element, paired the previous row to part="company" variable="carReservationRequest"
AssignTime = Start	This doesn't have any direct effect since the Task is of type "None." All Assignments with an AssignTime of "Start" will be combined for one assign element.
There are four other From/To Assignments that are not shown	These will map to additional from and to elements within a copy .

Tabel 3. Pemetaan service task

BPMN Object/Attribute	BPEL Element/Attribute
Task (TaskType: Service)	invoke
Name = "Check Credit Card"	name="checkCreditCardRequest"
InMessage	inputVariable="checkCreditCardRequest"
OutMessage	outputVariable="checkCreditCardResponse"
Implementation = Web service	See next three rows...
Participant	partnerLink="creditCardCheckingService"
Interface	portType="wsdl4:creditCardCheckingServiceImpl"
Operation	operation="doCreditCardChecking"
Assignment	assign . The name attribute is automatically generated by the tool creating the BPEL document.
From = input.cardNumber	within a copy element, paired with the next row from part="cardNumber" variable="input"
To = checkCreditCardRequest. cardNumber	within a copy element, paired with the previous row to part="cardNumber" variable="checkCreditCardRequest"
AssignTime = Start	This means that the assign element will precede the invoke
From = input.cardType	within a copy element, paired the next row: from part="cardType" variable="input"
To = checkCreditCardRequest. cardType	within a copy element, paired the previous row to part="cardType" variable="checkCreditCardRequest"
AssignTime = Start	This means that the assign element will precede the invoke . All Assignments with an AssignTime of "Start" will be combined for one assign element.

Tabel 3 dan Tabel 4 menjelaskan bagaimana proses pemetaan akan ditangani dengan cara yang berbeda-beda untuk setiap elemen *task*. Pada *task* biasa, elemen tersebut akan dipetakan menjadi aktivitas yang tidak memanggil layanan apapun sedangkan pada *service task*, elemen tersebut akan dipetakan sebagai proses *invoke* pada BPEL. Hal tersebut berlaku juga untuk pemetaan elemen-elemen lainnya [10].

Tabel 4. Transformasi *start event* menjadi *receive*

BPMN Object/Attribute	BPEL Element/Attribute
Start Event (EventType: Message)	receive
Name = "Receive"	name="Receive"
Instantiate = "True"	createInstance="yes"
Message = "input"	variable="input"
Implementation = "Web service"	See next three rows...
Participant = "ProcessStarter"	partnerLink="ProcessStarter"
Interface = "travelPort"	portType="wsdl0:travelPort"
Operation = "book"	operation="book"

Tabel 5. Transformasi *end event* menjadi *receive*

BPMN Object/Attribute	BPEL Element/Attribute
End Event (EventType: Message)	reply
Name = "Reply"	name="Reply"
Message = "output"	variable="output"
Implementation = "Web service"	See next three rows...
Participant = "ProcessStarter"	partnerLink="ProcessStarter"
Interface = "travelPort"	portType="wsdl0:travelPort"
Operation = "book"	operation="book"

Tabel 4 dan Tabel 5 menjelaskan bagaimana proses pemetaan akan ditangani dengan cara yang berbeda-beda untuk setiap elemen *task*. Pada *task* biasa, elemen tersebut akan dipetakan menjadi aktivitas yang tidak memanggil layanan apapun sedangkan pada *service task*, elemen tersebut akan dipetakan sebagai proses *invoke* pada BPEL. Hal tersebut berlaku juga untuk pemetaan elemen-elemen lainnya [10].

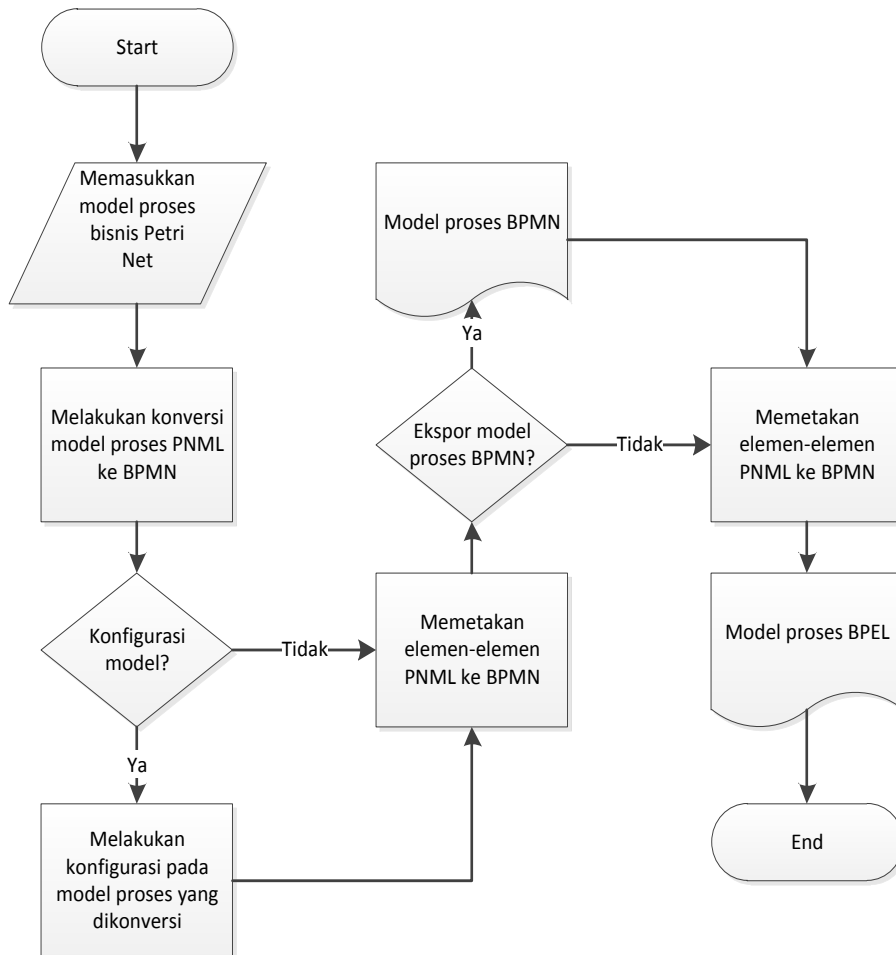
f. ProM Framework

ProM adalah kerangka kerja yang dibuat oleh Eindhoven Technical University untuk mendukung berbagai teknik penggalian proses dalam bentuk *plugin*. Dibangun di atas platform Java dan bersifat indepeden dan terbuka bagi para peneliti maupun pengembang untuk menggunakan kerangka kerja ProM atau ikut berkontribusi membuat *plugin* baru [11].

9. RINGKASAN ISI TUGAS AKHIR

Aplikasi yang dibangun dapat memodelkan Petri Net menjadi BPMN yang merupakan standar dari pemodelan bisnis tanpa mengubah alur proses dengan konfigurasi manual dari user maupun otomatis untuk setiap elemen-elemen yang dipetakan. Aplikasi juga

dapat memetakan model proses BPMN menjadi model proses BPEL yang merupakan standar dari implementasi model proses bisnis. Dari data yang digunakan sebagai masukan dilakukan serangkaian proses hingga menghasilkan keluaran yang diharapkan. Gambar 6 menunjukkan rangkaian proses yang dilakukan oleh aplikasi yang dibangun.

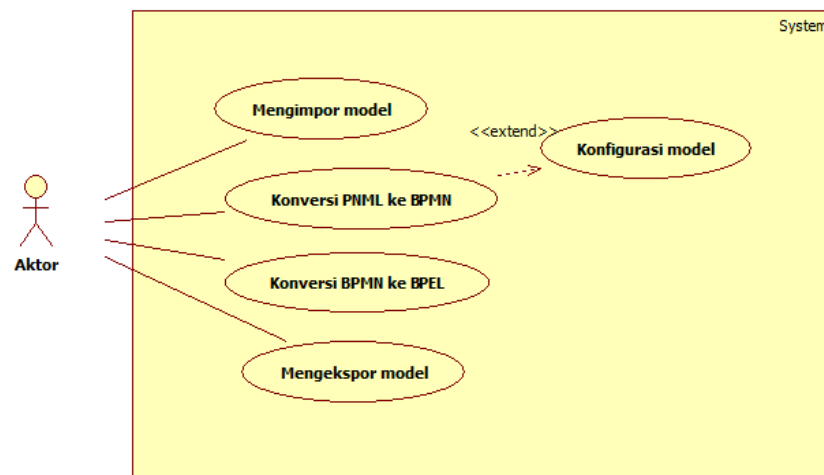


Gambar 6. Alur Sistem

Gambar 7 adalah gambaran diagram kejadian dari aplikasi yang akan dibuat. Penjelasan dari masing-masing kejadian adalah sebagai berikut:

1. Mengimpor model
Pengguna aplikasi memasukkan model yang akan dikonversi berupa file dengan notasi PNML atau BPMN.
2. Konversi PNML ke BPMN
Fitur untuk memetakan model PNML menjadi BPMN. Fitur ini dapat dijalankan setelah pengguna sudah mengimpor model PNML.

3. Konfigurasi model
Pada saat menjalankan proses pemetaan model PNML menjadi BPMN, pengguna melakukan konfigurasi dan penyesuaian dalam proses pemetaan model PNML menjadi BPMN.
4. Konversi BPMN ke BPEL
Fitur untuk memetakan model BPMN menjadi BPEL. Fitur ini dapat dijalankan setelah pengguna sudah mengimpor model BPMN atau sesaat setelah proses pemetaan PNML menjadi BPMN selesai dilakukan.
5. Mengekspor model
Pengguna aplikasi dapat mengekspor model yang telah dikonversi, yaitu model BPMN atau BPEL.



Gambar 7. Diagram kasus penggunaan

10.METODOLOGI

a. Penyusunan proposal tugas akhir

Proposal ini membahas mengenai rencana pengembangan perangkat lunak yang dapat menjembatani lingkup analisis dan desain dengan implementasi dalam ERP. Pada bagian latar belakang dibahas mengenai beberapa hal yang melatarbelakangi dibuatnya aplikasi ini yaitu adalah dengan adanya celah diantara lingkup analisis desain dengan implementasi. Setelah itu adalah rumusan masalah dan batasan masalah yang ingin diselesaikan. Kemudian adalah tujuan dan manfaat yang ingin dicapai dalam pengembangan aplikasi ini.

Pada tinjauan pustaka dibahas mengenai metode yang digunakan sebagai solusi dalam penyelesaian masalah yang telah didefinisikan. Metode yang dibahas adalah pemrosesan elemen dalam setiap pemetaan antar model. Setelah tinjauan pustaka dilanjutkan dengan ringkasan tugas akhir yang menjelaskan arsitektur perangkat lunak serta fitur-fitur yang ingin dikembangkan. Pada bagian metodologi dijelaskan tahap-tahap pengerjaan tugas akhir hingga pembuatan buku. Proposal diakhiri dengan jadwal pengerjaan tugas akhir serta daftar pustaka yang dipakai.

b. Studi literatur

Dalam pembuatan proposal tugas akhir ini telah dipelajari tentang hal-hal yang dibutuhkan sebagai dasar dan penunjang dalam pengembangan aplikasi yang dibuat nantinya. Yang pertama dipelajari adalah teknik pemetaan untuk setiap model yang akan dikonversi. Selanjutnya adalah teknik implementasinya yaitu dengan menggunakan teknik pemrosesan elemen. Kemudian juga dipelajari bagaimana mengimplementasikan aplikasi sebagai *plugin* ProM. Selain itu, juga dibantu beberapa literatur lain yang dapat menunjang proses penyelesaian tugas akhir ini.

c. Analisis dan desain perangkat lunak

Aplikasi ini dikembangkan sebagai *plugin* ProM yang dibangun di atas arsitektur Java. Aktor yang menjadi pelaku adalah pengguna ProM yang memanfaatkan plugin ini. Adapun kebutuhan fungsional dari *plugin* ini adalah:

1. Mengimpor model
2. Konversi PNML ke BPMN
3. Konfigurasi model
4. Konversi BPMN ke BPEL
5. Mengekspor model

d. Implementasi perangkat lunak

Pada tahap ini dilakukan pembuatan perangkat lunak yang merupakan implementasi dari rancangan yang telah dibuat. Beberapa hal yang diperlukan antara lain:

1. IDE menggunakan Eclipse 4.3.0
2. Eclipse BPMN 2.0 Plugin
3. Eclipse BPEL Designer Plugin
4. WoPeD 3.1.0 Workflow Petri Net Editor
5. ProM 6 Framework
6. ProM 5.2 Library
7. Simple 2.1 XML Serilization and Configuration Framework for Java

e. Pengujian dan evaluasi

Pada tahap ini dilakukan pengujian terhadap sistem menggunakan contoh data yang telah disediakan. Uji coba dilakukan untuk menguji fungsionalitas sistem, mencari masalah yang mungkin timbul pada sistem dan melakukan perbaikan bila ada kekurangan.

Uji coba dilakukan dengan menggunakan Eclipse BPMN 2.0 Plugin untuk mencoba membuka dan menjalankan file keluaran berupa BPMN dan Eclipse BPEL Designer Plugin untuk membuka dan menjalankan file keluaran berupa BPEL yang kemudian akan divalidasi dengan menggunakan Eclipse BPEL Plugin Validator. Data uji berupa *Petri Net* diperoleh dari contoh-contoh hasil analisis atau data uji yang dibuat sebelumnya dengan menggunakan WoPeD 3.1.0.

f. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
 - a. Latar Belakang
 - b. Rumusan Masalah
 - c. Batasan Tugas Akhir
 - d. Tujuan
 - e. Metodologi
 - f. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

11.JADWAL KEGIATAN

Tahapan	Bulan (2013-2014)											
	Oktober			Nopember			Desember			Januari		
Studi Literatur												
Perancangan sistem												
Implementasi												
Pengujian dan evaluasi												
Penyusunan buku												

12. DAFTAR PUSTAKA

- [1] W. v. d. Aalst, "Process-Aware Information Systems: Lessons to be Learned from Process Mining," *Department of Mathematics and Computer Science, Eindhoven University of Technology*, p. 27.
- [2] Wikipedia, "Enterprise Resource Planning," Wikimedia, [Online]. Available: http://en.wikipedia.org/wiki/Enterprise_resource_planning. [Accessed 2 October 2013].
- [3] Wikipedia, "Service Oriented Architecture," Wikimedia, [Online]. Available: en.wikipedia.org/wiki/Service-oriented-architecture. [Accessed 2 October 2013].
- [4] OASIS, "Web Services Business Process Execution Language Version 2.0," *OASIS Standard*, p. 264, 2007.
- [5] E. K. Michael Weber, "The Petri Net Markup Language," p. 21.
- [6] W3C, "Extensible Markup Language," W3C, [Online]. Available: <http://www.w3.org/XML>. [Accessed 2 October 2013].
- [7] S. A. White, "Introduction to BPMN," *IBM Corporation*, p. 11.
- [8] M. Rosen, "BPM and SOA," Wilton Consulting Group., 2008.
- [9] Oracle, "Transforming BPMN into BPEL: Why and How," Oracle, September 2008. [Online]. Available: <http://www.oracle.com/technetwork/articles/dikmans-bpm-101437.html>. [Accessed 2 October 2013].
- [10] A. L. Botchkovski, N. V. Mikhaylov and S. S. Pospelov, "GPS/GLONASS Receiver in Land Vehicle: Expectation and Reality," *11th International Conference on ITS Telecommunications*, pp. 287-292, 2011.
- [11] W. v. d. Aalst, "The ProM framework: A new era in process mining tool support," *Department of Technology Management, Eindhoven University of Technology*, p.