**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# IDENTITAS PENGUSUL

**NAMA : ENDANG WAHYU PAMUNGKAS**

**NRP : 5110100106**

**DOSEN WALI : Anny Yuniarti S.Kom, M.Comp.**

**DOSEN PEMBIMBING : 1. Prof. Ir. Riyanarto Sarno, SE, MSc, PhD**

**2. Abdul Munif, S.Kom, M.Sc**

# JUDUL TUGAS AKHIR

“Rancang Bangun Aplikasi untuk Menghasilkan Model Proses Bisnis yang Fleksibel pada Sistem Enterprise Resource Planning”

# LATAR BELAKANG

Saat ini teknologi *Business Process Management* (BPM) telah berkembang dan banyak diaplikasikan baik dalam skala kecil maupun skala besar. Bentuk riilnya banyak perusahaan atau organisasi yang memanfaatkan sistem *Enterprise Resource Planning* (ERP), dan sistem lainnya yang berorientasi pada proses bisnis. Sebagai penyedia jasa yang membuat layanan ERP, harus dapat menyediakan model proses bisnis yang merupakan *best practice* dalam suatu bagian industri [9]. Kemudian konsumen dapat menyesuaikan proses bisnis yang ada dengan kebutuhannya. Sehingga dibutuhkan model yang fleksibel dari penyedia jasa sehingga memudahkan kerja dari konsumen. Dilain sisi fleksibilitas merupakan salah satu tantangan dalam *Process Aware Information System* (PAIS). PAIS merupakan sistem perangkat lunak yang mengatur serta menjalankan proses operasional berkaitan dengan manusia, perangkat lunak dan atau sumber informasi yang berbasis sistem model proses [10]. Saat ini semakin banyak perusahaan yang mengadopsi PAIS yang menawarkan perspektif menjanjikan dengan *enterprise computing* lebih fleksibel [8]. Konsep proses bisnis fleksibel dalam PAIS merupakan proses bisnis yang dapat mengakomodasi setiap perubahan yang terjadi di lingkungannya.

Dalam beberapa tahun terakhir teknologi manajemen proses bisnis semakin banyak digunakan sejalan dengan banyaknya proses bisnis yang ada. Dengan teknologi ini, suatu perusahaan dapat membangun dan memperbaharui setiap informasi yang ada dalam proses bisnisnya secara cepat termasuk pada repositori model prosesnya. Sehingga setiap pelayanan yang diberikan oleh perusahaan tersebut dapat berubah sesuai perubahan dengan cepat. Perubahan yang terjadi ini dapat dikarenakan oleh perubahan kebijakan atau dapat juga dikarenakan perubahan kondisi pasar [1]. Banyaknya penggunaan teknologi manajemen proses bisnis dapat dilihat dari banyaknya perusahaan-perusahaan yang menggunakan otomasi kinerja perusahaannya. Saat ini ERP merupakan contoh nyata dari penggunaan teknologi manajemen proses bisnis. Tercatat di perusahaan China Haier memiliki lebih dari 3000 model proses bisnis dan juga 600 EPC di dalam SAP *reference model*. Teknologi ini juga diterapkan oleh salah satu perusahaan terbesar di China, CNR Corporation Limited, yang merupakan gabungan dari 20 perusahaan di China. Sebelum bergabung setiap perusahaan tersebut memiliki masing-masing hampir 200000 model proses. Sehingga dapat dihitung sendiri berapa total model proses yang harus diintegrasikan [1].

Dalam dunia industri, misalnya seperti permintaan konsumen, inovasi teknologi, dan peraturan dapat berubah setiap waktu. Perubahan yang terjadi ini harus diikuti dengan perubahan proses bisnis yang telah berjalan. Sehingga proses bisnis yang ada harus mampu menyesuaikan terhadap setiap perubahan yang mungkin terjadi. Secara umum *Business Process Management Suite* (BPMS) akan menyediakan fasilitas untuk melakukan perubahan [11]. Namun, reference model yang ada sendiri harus mudah dilakukan perubahan. Sehingga usaha dan biaya yang dikeluarkan konsumen dapat ditekan seminimal mungkin.

Saat ini proses desain selalu menjadi bagian yang ditinggalkan sebagai fokus dalam pembuatan proses bisnis. Ketika disibukkan pada bagian implementasi, biaya yang dikeluarkan cukup besar saat terjadi perubahan. Hal ini disebabkan rendahnya analisis pada bagian desain. Sehingga sangat dibutuhkan suatu metode yang dapat melakukan analisis pada bagian desain yang nantinya akan menghasilkan desain model yang diharapkan. Sebenarnya dengan adanya proses bisnis yang fleksibel ini tidak hanya menguntungkan bagi konsumen. Namun, sebagai penyedia layanan juga akan mereduksi waktu desain yang terbuang ketika harus merubah proses bisnis yang ada menyesuaikan permintaan konsumen. Pada tugas akhir ini akan diimplementasikan suatu perangkat lunak berupa *plugin* yang dapat menghasilkan proses bisnis yang fleksibel. Dalam pembentukan proses bisnis baru yang fleksibel tersebut dibutuhkan data berupa proses bisnis yang sudah ada dan berjalan. Dengan metode pengelompokan berdasar kesamaan serta penggabungan diharapkan menghasilkan proses bisnis fleksibel seperti yang diinginkan.

# RUMUSAN MASALAH

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

1. Bagaimana mencari nilai kesamaan antara model proses bisnis dari segi struktur dan *behavioral*.
2. Bagaimana cara melakukan pengelompokan pada model proses bisnis sesuai dengan nilai kesamaannya.
3. Bagaimana menarik *common fragment* model proses bisnis pada setiap kelompok yang terbentuk.
4. Bagaimana melakukan konversi dari satu notasi model proses bisnis ke notasi lainnya.

# BATASAN MASALAH

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

1. Bahasa pemrograman menggunakan bahasa Java.
2. Perangkat lunak yang dikembangkan merupakan *plugin* dari *Prom 6*.
3. Data uji menggunakan *Petri Net Markup Language* (PNML) dan menghasilkan keluaran berupa EPC.

# TUJUAN PEMBUATAN TUGAS AKHIR

Tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Membuat program yang dapat mendapatkan nilai kesamaan antar model proses bisnis.
2. Membuat program yang dapat melakukan proses pengelompokan pada sekumpulan model proses bisnis berdasarkan kesamaannya.
3. Membuat program yang dapat mengambil *common fragment* dari model proses bisnis pada setiap kelompok yang terbentuk.
4. Membuat program yang dapat melakukan proses konversi dari notasi petri net ke notasi EPC.

# MANFAAT TUGAS AKHIR

Manfaat yang diharapkan dari tugas akhir ini adalah :

1. Mempermudah dalam mencari nilai kesamaan dari beberapa model proses bisnis.
2. Mempermudah untuk mengelompokkan beberapa proses bisnis yang mirip sehingga mempermudah analisa.
3. Mempermudah dalam proses efisiensi proses bisnis dengan menghasilkan *common fragment* dari beberapa proses bisnis yang mirip.
4. Membantu dalam proses konversi dari notasi proses bisnis petri net ke notasi EPC.
5. Mempermudah desain proses bisnis yang fleksibel.

# TINJAUAN PUSTAKA

* 1. **Plugin Prom**

Sebuah *plug-in* (atau *plugin* ekstensi) adalah komponen perangkat lunak yang menambahkan fitur khusus untuk aplikasi perangkat lunak yang ada. Bila aplikasi yang mendukung *plug-in*, memungkinkan melakukan perubahan. Banyak sekali aplikasi saat ini yang memberikan kelonggaran bagi pengembang untuk dapat menambahkan fitur di dalamnya. Dalam tugas akhir ini, akan dibuat *plugin* khusus untuk *software process mining* bernama *Prom* [12]. Saat ini *Prom* telah memiliki banyak sekali *plugin* yang tersedia secara gratis.

* 1. **Model Proses Bisnis**

Proses bisnis merupakan sekumpulan aktifitas yang dibuat untuk menghasilkan keluaran spesifik dengan tujuan tertentu [5]. Dari model tersebut juga dapat diketahui informasi dimana dan kapan suatu aktifitas dilakukan, kondisi awal sebelum aktifitas dilakukan, kondisi akhir setelah aktifitas dilakukan, serta masukan dan keluaran yang jelas. Adapun ciri-ciri dari proses bisnis itu sendiri adalah sebagai berikut :

1. Mempunyai tujuan tertentu
2. Mempunyai masukan yang spesifik.
3. Mempunyai keluaran yang spesifik.
4. Memanfaatkan *resource*.
5. Memiliki aktifitas yang dapat dieksekusi dengan urutan tertentu.
6. Dapat melibatkan lebih dari satu organisasi.

Model proses bisnis atau business process model merupakan representasi dari proses bisnis. Sehingga sebuah model proses bisnis harus secara jelas mendefinisikan setiap ciri-ciri yang harus dimiliki oleh suatu proses bisnis. *Unified Modeling Language* (UML) merupakan salah satu representasi dasar dari proses bisnis. Saat ini representasi dari model proses bisnis itu sendiri sudah banyak berkembang dan banyak jenisnya. Mulai dari UML, BPEL, *Business Process Modeling Language* (BPMN), *Event-driven Process Chain* (EPC), PNML, dan masih banyak lagi. Tetapi, masing-masing jenis tersebut juga memiliki keuntungan sendiri-sendiri apabila ingin dimanfaatkan.

* 1. ***Structural Similarity***

*Structural similarity* dihitung dengan pemikiran *dependency graph* [3]. Dalam *paper* tersebut dijelaskan bahwa *insidence matric* dari *petrinet* digunakan sebagai *process matrix* oleh sebuah *dependency graph*. Kemudian dihitung perbedaan *dependency matrix* dari kedua model sebagai *distance* (d). Kemudian nilai kesamaan didapatkan dengan rumus 1/(1+d). Berikut langkah dalam mencari nilai *distance* [4] :

* Pembentukan dependency graph. Dalam [4] *dependency graph* (DG) didefinisikan dengan *binary tuple* <DN, DE>. Dimana DN merupakan kumpulan dari *activity nodes* dan DE merupakan kumpulan dari *edge* yang menghubungkan *activity nodes*.
* *Comparability filter*. Pada bagian ini dilakukan pembandingan antar *dependency graph* dengan *δ-Comparability*. Pada proses ini dibutuhkan nilai *threshold* δ. Dua *dependency graph* misal dan dikatakan *δ-Comparable* jika memenuhi kondisi

(1)

* Membentuk *process matrix*. *Process matrix* sama dengan *adjacency matrix* M yaitu *matrix* n x n yang berisi informasi apakah satu *node* dengan *node* lain saling berhubungan.

(2)

* Melakukan normalisasi pada *process matrix*. Normalisasi dimaksudkan untuk menyamakan jumlah *row* dan *column* pada kedua *process matrix*. Misal pada model 1 terdapat *node* yang tidak ada di model lain, maka ditambahkan *row* dan *column* di *process matrix* dengan nilai 0. Begitu juga pada model yang lainnya. Misal terdapat 2 *dependency graph* dan . Kemudian dan merupakan *matrix* yang telah dinosrmalisasi. Kemudian didapatkan maka :

(3)

(4)

* Mendapatkan nilai *distance*. Nilai *distance* dapat dicari dengan melakukan *matrix* *substraction* . Setelah itu, akan didapat *matrix* yang didalamnya terdapat nilai 1 atau -1. *Distance* didapatkan dengan menjumlahkan nilai tersebut. Namun sebelumnya nilai -1 harus dipositifkan terlebih dahulu.
  1. ***Behavioural Similarity***

*Behavioural similarity* dihitung dengan *Principle Transition Sequence* (PTS) *metric* [5]. Dalam mencari nilai *behaviour similarity* prinsip tentang *coverability tree* sangatlah penting [7]. PTS dianggap bagus dalam perhitungan *behaviour similarity* karena dapat menangani beberapa kasus pada petrinet. *Complex structure, non-free choice, arbritrary loop*, dan lainnya dapat ditangani oleh PTS. Sehingga diharapkan didapatkan nilai *behaviour similarity* dengan aproksimasi yang bagus [3]. Berikut formula dalam mencari nilai kesamaan dengan PTS :

Misalkan dan merupakan 2 *labeled petrinet*, maka dan merupakan *transition sequence* dari kedua petrinet tersebut. Tingkat kemiripan dari 2 PTS didapat berdasar nilai *Longest Common Subsequence* (LCS) seperti formula di bawah ini :

(5)

Setelah didapatkan formula mengenai cara mencari nilai kesamaan dari 2 *seqeuence*, selanjutnya dikembangkan dengan mencari kesamaan dari 2 set *transition sequence*. Misal P dan Q adalah 2 sets yang berisi kumpulan *transition*, maka :

(6)

Kemudian baru didapatkan formula untuk mencari kesamaan antara 2 *labeled petrinet*. Berikut formulanya :

(7)

Dimana dan mengikuti definisi dari *Principal Transition Sequence* (PTS) sebagai berikut :

Misal : merupakan *labeled petrinet*, kemudian merupakan *insidence matrix* dari *petrinet* tersebut, kemudian merupakan *coverable tree* dari *labeled petrinet* dengan *root* , merupakan kumpulan *node dead-end*, merupakan *old node*. Maka dapat diketahui dengan ketentuan berikut :

* If
* If
* If

Maksud dari berarti ada transisi dari node dan pada *coverability tree* . Dari definisi diatas maka didapatkan. Sedangkan didapatkan sesuai model petrinet tersebut masuk ke kategori PTS mana. Berikut kategori PTS yang dimaksud :

* = PTS yang tidak mengandung transition berulang.
* = PTS yang berulang namun bersifat *finite*.
* = PTS yang berulang dan bersifat *infinite*.
  1. **Klustering dengan *Graph Partition Approaches***

Metode pengelompokan yang dipakai berdasarkan jarak antara 2 entitas yang mirip dengan graph partition. Metode ini biasa dipakai ketika objek yang dikelompokan sulit untuk direpresentasikan ke dalam bentuk matematika [6]. *Distance* yang digunakan dalam proses pengelompokan berdasarkan kesamaan dari masing-masing model. Kumpulan proses bisnis yang memiliki tingkat kesamaan tinggi tentunya akan menjadi 1 kelompok.

* Tentukan *threshold*
* Hitung semua nilai kesamaan antar model
* Ulangi untuk setiap model bandingkan nilai kesamaan dengan *threshold*
* Untuk 2 model yang nilai kesamaannya diatas *threshold* diberi *edge* yang menghubungkan keduanya.
* Untuk model yang tidak tehubung dengan model manapun berarti tidak masuk ke kelompok manapun.
* Satu *graph* yang terbentuk menunjukkan satu kelompok. Jumlah kelompok sama dengan jumlah *graph* yang terbentuk.
  1. ***Common Fragment***

*Common fragment* yang dimaksud disini merupakan rangkaian *node* dan *edge* dimana *node* dan *edge* tersebut selalu ada dalam beberapa *graph*. Jadi *common fragment* didapatkan dari penggabungan beberapa *graph* dan menghilangkan *node* dan *edge* yang frekuensinya kurang. Jadi proses dalam mendapatkan *common fragment* yaitu dengan menggabungkan beberapa *graph* menjadi satu *graph*. Dari *graph* gabungan tersebut dilakukan proses *digesting* hingga didapatkan 1 *graph* yang merupakan *fragment* utama [11]. Berikut contoh penarikan *common fragment* :

A

A

A

C

E

C

C

B

D

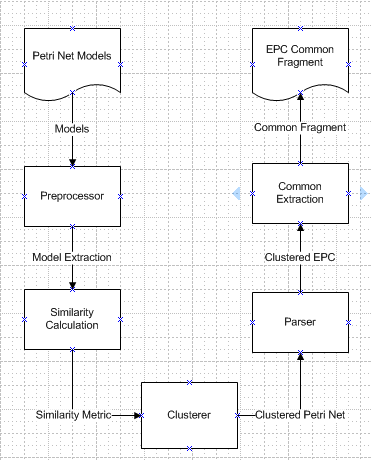
D

D

Gambar 1. Proses *digestion* untuk *common fragment*

# RINGKASAN ISI TUGAS AKHIR

Perangkat lunak yang akan dibangun dapat menghasilkan proses bisnis fleksibel. Proses pembentukan proses bisnis tersebut membutuhkan data proses bisnis yang sudah ada dan berjalan. Dari data yang digunakan sebagai masukan dilakukan serangkaian proses hingga menghasilkan keluaran yang diharapkan. Berikut alur pemrosesan dan bentuk arsitektur perangkat secara sederhana :



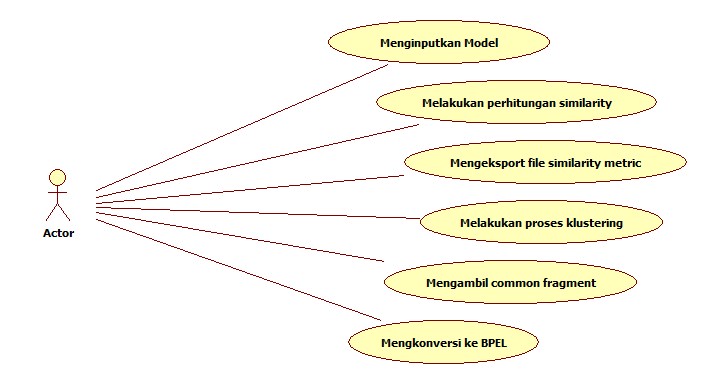
Gambar 2. Arsitektur Sistem

Tahap *preprocessor* merupakan tahap mengekstrasi informasi dari kumpulan model yang menjadi masukan. Informasi tersebut berupa *edge-edge* dan *node-node* pada masing-masing model yang nantinya digunakan dalam perhitungan. Informasi *edge* dan *node* yang berupa bahasa xml akan dienkapsulasi ke objek untuk memudahkan pemrosesan data lebih lanjut.

Setelah tahap *preprocessing*, selanjutnya informasi mengenai model yang telah dienkapsulasi akan digunakan untuk perhitungan kemiripan antar model. Setiap model akan dibandingkan satu per satu hingga dihasilkan *similarity metric* dari semua model. Dari *similarity metric* yang dihasilkan akan digunakan sebagai dasar dalam melakukan pengelompokan. Proses pengelompokan menggunakan metode *graph partition approaches*.

*Threshold* disini maksudnya adalah nilai kesamaan minimal yang memungkinkan pembentukan *subgraph*. Proses selanjutnya adalah pengambilan *common fragment* dari setiap kelompok yang terbentuk. Sebelumnya diperlukan proses konversi dari model bentuk *Petri Net* ke bentuk EPC. Hal ini diperlukan karena struktur xml EPC yang lebih mudah untuk direpresentasikan sebagai sebuah *graph*. Proses konversinya sendiri dengan cara melakukan mapping dari setiap *tag-tag* xml.

Setelah proses konversi selesai, selanjutnya adalah pengambilan *common fragment* pada setiap kelompok yang terbentuk. Dalam pembentukan *common fragment* langkah pertama adalah melakukan merging pada semua model yang berada dalam 1 kelompok. Kemudian baru dilakukan proses *digestion* pada *merging* model yang dihasilkan [12]. Berikut disertakan diagram kasus penggunaan perangkat lunak yang akan dibangun :



Gambar 3. Diagram Kasus Penggunaan

Penjelasan fitur aplikasi yang ada adalah sebagai berikut :

1. Menginputkan Model

Merupakan fitur yang digunakan untuk memasukkan model proses bisnis yang akan digunakan sebagai dasar pembentukan model baru. Model yang digunakan harus dalam notasi PNML.

1. Melakukan Perhitungan Kesamaan antar Model

Pada fitur perhitungan nilai kesamaan ini akan menghasilkan *similarity metric*. *Similarity metric* ini berisi nilai tingkat kesamaan antar model. Terdapat 2 *metric* yaitu *behavioural* dan juga *structural similarity*. 2 *metric* ini digabungkan dengan prosentase tertentu. Nilai tersebut yang akan digunakan sebagai dasar dalam melakukan pengelompokan.

1. Mengeksport *File Similarity Metric*

Fitur ini merupakan tambahan dari fitur perhitungan *similarity metric*. Jadi *metric* yang dihasilkan dapat dieksport ke file *excel* yang mungkin dibutuhkan untuk analisa.

1. Melakukan Proses Pengelompokan

Fitur ini menjalankan proses pembentukan kelompok berdasarkan *similarity metric*. Jadi fitur ini akan menghasilkan beberapa kelompok model yang di dalamnya berisi model-model yang memiliki nilai kesamaan tinggi.

1. Mengambil *Common Fragment*

Dari beberapa kelompok yang dihasilkan pada fitur pengelompokan akan diambil *common fragment*-nya di fitur ini. *Commont fragment* ini yang merupakan model terakhir dan menjadi keluaran sebagai model yang fleksibel.

# METODOLOGI

## Penyusunan proposal tugas akhir

Pada proposal ini akan dibahas mengenai rencana pembangunan perangkat lunak yang akan membantu desain proses bisnis yang fleksibel. Pada bagian latar belakang dibahas mengenai beberapa alasan yang mendasari pengerjaan tugas akhir. Kondisi perkembangan PAIS saat ini merupakan latar belakang utama dalam pembuatan tugas akhir ini. Dimana salah satu tantangan dari PAIS adalah pengefisienan *design time* proses bisnis serta pembentukan proses bisnis yang fleksibel. Kemudian dilanjutkan dengan beberapa rumusan serta batasan masalah yang ingin diselesaikan. Setelah itu tujuan dan manfaat yang ingin dicapai dalam tugas akhir ini.

Pada tinjauan pustaka dibahas mengenai beberapa metode serta algoritma yang akan digunakan dalam penyelesaian masalah yang sebelumnya telah didefinisikan. Metode yang dibahas merupakan proses perhitungan kesamaan proses bisnis secara struktur dan secara *behavior*. Selain itu juga dibahas mengenai algoritma pengelompokan proses bisnis yang digunakan. 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.

## Studi literatur

Dalam pembuatan tugas akhir ini telah dipelajari tentang hal-hal yang dibutuhkan sebagai ilmu penunjang dalam penyelesaiannya. Pertama adalah tentang perhitungan kemiripan antar model proses bisnis. Kemudian juga dipelajari tentang cara pengelompokan beberapa proses bisnis sesuai kemiripannya. Literatur yang penting lainnya adalah tentang bagaimana cara mengambil *common fragment* dari beberapa model proses bisnis. Selain itu, juga dibantu beberapa literatur lain yang dapat menunjang proses penyelesaian tugas akhir ini.

## Analisis dan desain perangkat lunak

Aktor yang menjadi pelaku adalah pengguna *Prom* yang memanfaatkan *plugin* ini. Kemudian beberapa kebutuhan fungsional dari *plugin Prom* ini adalah sebagai berikut :

1. Melakukan proses perhitungan kesamaan antar model proses bisnis.
2. Melakukan proses pengelompokan model proses bisnis berdasarkan kemiripannya.
3. Melakukan konversi dari notasi petrinet ke EPC.
4. Melakukan proses pengambilan common fragment dari beberapa model proses bisnis.

## Implementasi perangkat lunak

Berikut beberapa hal yang diperlukan dalam implementasi :

1. IDE menggunakan *Eclipse Kepler*.

2. Perangkat yang dikembangkan merupakan *plugin Prom*.

3. *Java Development Kit* 7.0.

4. *Java Runtime Environment*.

5. *Library Prom*.

## Pengujian dan evaluasi

Proses pengujian dilakukan secara *blackbox*. Data uji diperoleh dari proses bisnis ERP yang telah berjalan dan dimodelkan dengan notasi *petrinet*. Dari data *petrinet* yang dimasukkan nantinya akan dicocokan apakah keluaran dari masing-masing fitur yang ada sudah benar. Hingga dihasilkan keluaran terakhir berupa *common fragment* yang merupakan beberapa proses bisnis yang fleksibel.

## 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
   1. Latar Belakang
   2. Rumusan Masalah
   3. Batasan Tugas Akhir
   4. Tujuan
   5. Metodologi
   6. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka
7. **JADWAL KEGIATAN**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tahapan | Bulan (Tahun 2013) | | | | | | | | | | | | |
| Oktober | | | | Nopember | | | | Desember | | | |
| Analisa kebutuhan dan studi literature |  |  |  |  |  |  |  |  |  |  |  |  |
| Perancangan system |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementasi |  |  |  |  |  |  |  |  |  |  |  |  |
| Uji coba dan evaluasi |  |  |  |  |  |  |  |  |  |  |  |  |
| Penyusunan buku |  |  |  |  |  |  |  |  |  |  |  |  |

1. **DAFTAR PUSTAKA**
2. Jin T., Wang J., and Wen L., Querying Business Process Model based on Semantic, *Proceedings of the 16th international conference on Database systems for advanced applications: Part II*, pp.164-178, 2011.
3. Sparx System, UML Tutorial, White Paper, “The Business Process Model”, page 1-4, 2004.
4. Wang, J., Wong, R. K., Ding, J., Guo, Q., and Wen, L., On Recommendation of Process Mining Algorithms, *Proceedings of the 2012 IEEE 19th International Conference on Web Services*, pp. 311-318.
5. Bae, J., Liu, L., Caverlee, J., and Rouse, W. B., Process mining, Discovery, and Intregation using Distance Measure, *Proceedings of the IEEE International Conference on Web Services,* pp. 479-488, 2006.
6. J. Wang, T. He, L. Wen, N. Wu, A.H.M. ter Hofstede, and J. Su. A behavioral similarity measure between labeled Petri nets based on principal transition sequences, *Proceedings of the 2010 international conference on On the move to meaningful internet systems - Volume Part I*, pp. 394–401.
7. Qiao M., Akkiraju R., and Aubrey J. Rembert, Towards Efficient Business Process Clustering and Retrieval: Combining Language Modelingand Structure Matching, *Proceedings of the 9th international conference on Business process management*, pp. 199-214, 2011.
8. T. Murata. Petri nets: properties, analysis and applications*. Proceedings of the IEEE, 77,* pp.541–580, 1989.
9. Weber B., Reichert M., and Rindele-Ma S., Change patterns and change support features – Enhancing flexibility in process-aware information systems, Data & Knowledge Engineering, Volume 66 Issue 3, September, 2008, Pages 438-466
10. [Balko](http://www.researchgate.net/researcher/69768150_Soeren_Balko/) S., [ter Hofstede](http://www.researchgate.net/researcher/14910465_Arthur_H_M_ter_Hofstede/) A. H. M., [Barros](http://www.researchgate.net/researcher/18677231_Alistair_P_Barros/) A. P., and [La Rosa](http://www.researchgate.net/researcher/14910467_Marcello_La_Rosa/) M., Controlled Flexibility and Lifecycle Management of Business Processes through Extensibility, *Proceedings of the 3rd International Workshop on Enterprise Modelling and Information Systems Architectures*, 2009, pp. 1-14
11. Aalst. W. M., Process-Aware Information Systems: Lessons tobe Learned from Process Mining, Book Transaction on Petri Nets and Other Models of Concurrency II, pp. 1-26.
12. La Rosa M., Dumas M., Uba R., and Dijkman R, Business Process Model Merging: An Approach to Business Process Consolidation, *ACM Transactions on Software Engineering and Methodology (TOSEM)*, Volume 22, Issue 2, March 2013, Article No. 11
13. van der Aalst W. M. P, van Dongen B. F., Gunther C. W., Mans R. S., A.K.A. de Medeiros, Rozinat A., Rubin V., Song M., Weijters A. J. M. M., and Verbeek H. M. W. ProM 4.0 : comprehensive support for real process analysis. In M. van Eekelen P. Groot, A. Serebrenik, editor, *26th International Conference on Applications and Theory of Petri Nets (ICA TPN 2007)*. Springer, 2007