

BAB III ANALISIS

3.1 Model Penerapan BPM pada SOA

Penerapan proses BPM pada sebuah organisasi akan mengakibatkan sistem yang digunakan terus berubah untuk mencapai proses bisnis yang lebih efisien dan efektif. Perubahan sistem tentunya akan mengakibatkan perubahan dari penggunaan aplikasi yang digunakan, baik dari segi proses bisnis yang harus dijalankan oleh aplikasi tersebut, ataupun pemilihan aplikasi apa saja yang digunakan oleh sebuah pekerjaan.

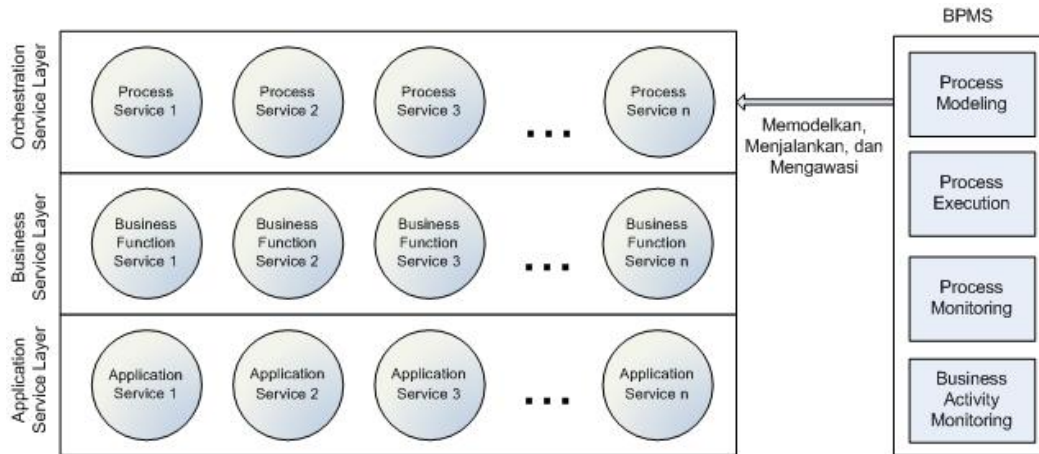
Seperti dijelaskan pada Bab 2.2.3, sebuah aplikasi yang tidak berbasis *service-oriented* akan mendefinisikan proses bisnisnya bersamaan dengan perintah yang berhubungan dengan sisi teknis. Hal ini menyebabkan proses perubahan proses bisnis harus turun ke kode.

Pengubahan yang langsung turun ke kode ini seharusnya bisa disederhanakan dengan memisahkan antara proses bisnis pada bagian tersendiri, sebab pada umumnya, yang berubah pada sistem adalah proses bisnisnya saja. Sementara fungsi-fungsi yang dijalankan tidak.

Penggunaan arsitektur SOA akan menjadi jawaban terhadap masalah yang dialami tersebut. Arsitektur SOA menjamin modularitas fungsi-fungsi dari aplikasi yang ada di dalamnya, dan memisahkan proses bisnis menjadi sebuah bagian tersendiri yang disebut lapisan orkestrasi. Perubahan proses bisnis cukup berlangsung pada lapisan orkestrasi tersebut dengan memanfaatkan fungsi-fungsi yang ada tersebut.

Namun, proses BPM tidak terkait hanya pada mengubah proses bisnis saja, tetapi juga menganalisis proses bisnis yang telah berjalan sehingga menjadi masukan untuk ke depannya.

Untuk itu, diperlukan lebih dari sekedar arsitektur SOA. Penggunaan BPMS merupakan jawabannya. BPMS merupakan sebuah alat bantu yang akan mengatur lapisan orkestrasi yang terdapat di dalam arsitektur SOA. Seperti dijelaskan pada Bab 2.1.3, BPMS ini terdiri atas *process modeling* untuk memodelkan proses, *process execution* untuk mengeksekusi, *process monitoring* untuk mengawasi, dan *business activity monitoring* untuk mengambil hasil analisis. Keempat kakas tersebut akan melakukan manajemen lapisan orkestrasi SOA. Penggunaan BPMS untuk arsitektur SOA inilah yang akan menjadi model penerapan BPM pada SOA. Model ini diilustrasikan pada Gambar III-1.



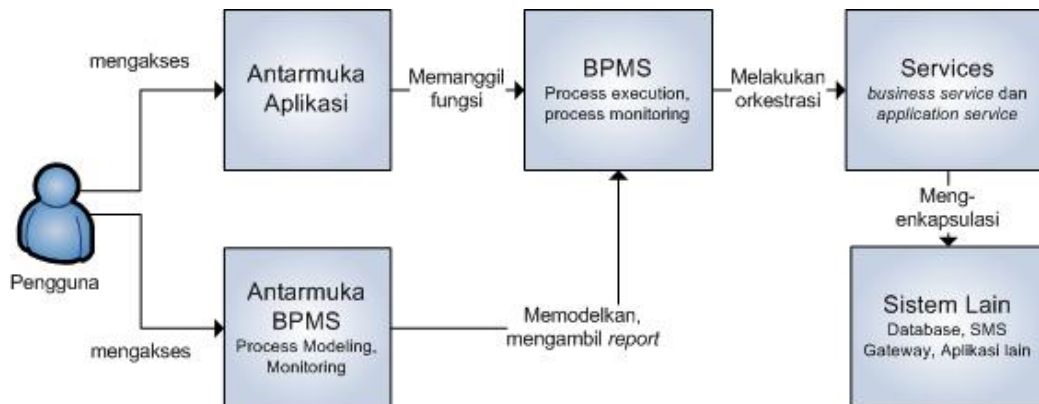
Gambar III-1 Model Penerapan BPM pada SOA

Pada dasarnya keempat komponen BPM tersebut tidak selalu dibutuhkan. Menurut [GHA06] sebuah BPMS cukup hanya memiliki *process modeling* dan *process execution* saja. BPMS seperti ini dapat disebut sebagai BPMS yang minimal. Jika selanjutnya dibutuhkan sebuah mekanisme administrasi dan pengawasan khusus, maka komponen lain dapat ditambahkan. Hal ini berarti kelengkapan BPMS yang digunakan bergantung pada kompleksitas dan kebutuhan aplikasi yang dikembangkan.

3.2 Implementasi Model

3.2.1 Gambaran Umum

Model penerapan BPMS pada SOA seperti telah dijelaskan pada Bab 3.1 memberi deskripsi pada *service interface layer* mengenai bagaimana *service* dan BPMS diposisikan. Namun deskripsi tersebut belum menggambarkan bagaimana perangkat lunak maupun sistem secara keseluruhan akan diimplementasi. Ilustrasi dari implementasi model di atas dapat dilihat pada Gambar III-2 Gambaran Umum Implementasi.



Gambar III-2 Gambaran Umum Implementasi

Sebuah perangkat lunak pada dasarnya adalah sebuah bagian dari sistem. Perangkat lunak tentunya perlu berinteraksi dengan pengguna (*user*) melalui antarmuka (*interface*) dalam menjalankan proses bisnis. Untuk memungkinkan proses interaksi diimplementasi dalam model ini, maka akan ada aplikasi pengguna *service*. Aplikasi tersebut memiliki antarmuka untuk digunakan oleh pengguna. Aplikasi pengguna *service* ini memiliki dua buah alternatif pemanggilan fungsi, yaitu ke komponen *process execution*, yang diimplementasikan dengan teknologi WS-BPEL, atau langsung menggunakan *service* pada *business service layer* maupun *application service layer*. Pemanggilan fungsi langsung ke *service* tanpa melalui komponen *process execution* dilakukan untuk fungsi-fungsi yang tidak terkait proses bisnis, yakni tidak terdiri atas sejumlah aktivitas.

Pada implementasi model ini, BPMS berperan untuk melakukan orkestrasi terhadap seluruh operasi yang dimiliki sejumlah *service* yang ada di dalam arsitektur ini. Orkestrasi ini dilakukan untuk proses-proses yang terkait dengan proses bisnis, yaitu yang terdiri atas satu aktivitas atau lebih. Orkestrasi ini dimungkinkan dengan menggunakan teknologi WS-BPEL. Deskripsi mengenai orkestrasi proses bisnis tersebut dilakukan oleh komponen *process modeling* yang ada pada Antarmuka BPMS. Selain itu, BPMS juga berperan dalam melakukan *monitoring* terhadap proses eksekusi yang dilakukan oleh dirinya sendiri. Hasil dari proses monitoring tersebut akan diambil oleh Antarmuka BPMS.

Sebuah kelengkapan lain yang perlu dimiliki oleh sebuah sistem adalah antarmuka dengan sistem atau aplikasi lain. Misalnya sistem basis data yang digunakan, SMS Gateway, dan lain-lain. Seluruh sistem tersebut akan dienkapsulasi oleh *application service layer* pada model, sehingga akan menjadi sekumpulan *service* dengan yang mengabstraksikan aplikasi-aplikasi tersebut.

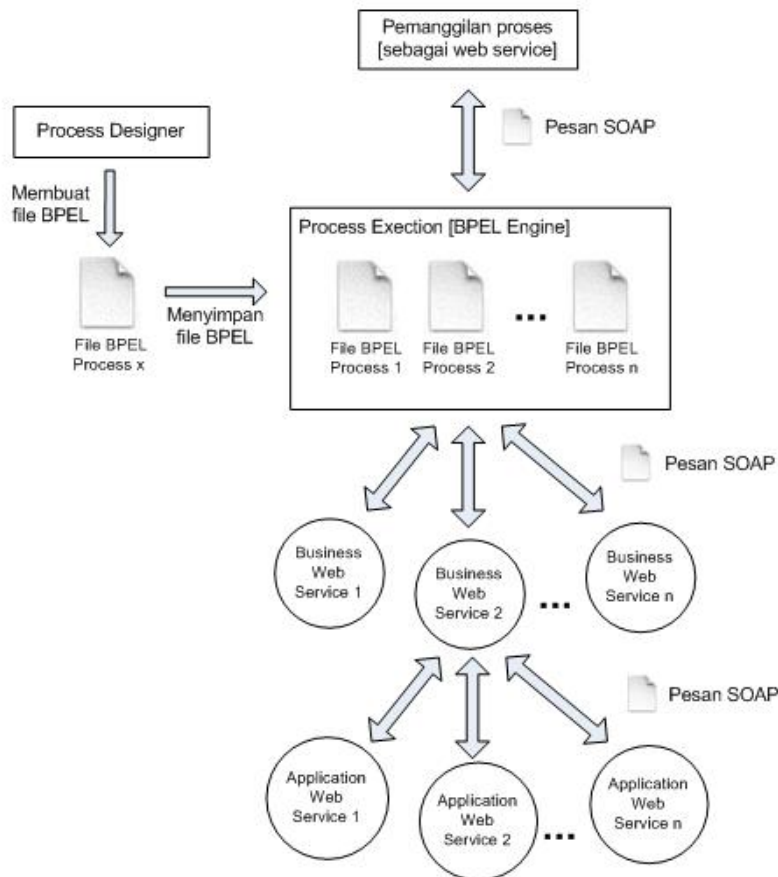
3.2.2 Pemilihan Teknologi

Seperti dijelaskan pada Bab 2.2.4, penggunaan SOA yang paling baik adalah dengan menggunakan *web service*. Oleh karena itu, pada implementasi model ini digunakan prinsip *contemporary SOA*, di mana setiap *service* adalah sebuah aplikasi *web service*, *operation* merupakan operasi atau layanan yang disediakan oleh *service* tersebut, *message* dikirimkan dengan format SOAP, dan setiap *service* dapat mengetahui operasi yang dimiliki *service* lain dengan WSDL.

Satu komponen SOA yang tersisa, yaitu proses bisnis, akan didefinisikan dengan WS-BPEL. WS-BPEL dipilih karena teknologi ini merupakan teknologi yang paling luas diterima dan digunakan untuk proses bisnis dan satu-satunya yang bisa menerapkan proses bisnis pada *web service*. Penggunaan WS-BPEL inilah yang mengakibatkan setiap proses didefinisikan sebagai sebuah proses, karena teknologi WS-BPEL membuat sebuah proses menjadi sebuah

web service yang hanya memiliki satu operasi, yakni operasi untuk menjalankan proses tersebut.

Peran BPMS dalam teknologi ini adalah untuk menunjang semua teknologi tersebut. Komponen *process designer* pada BPMS akan melakukan pembuatan kode BPEL sesuai proses yang didefinisikan, kemudian menyimpannya untuk dieksekusi oleh *process execution*. Jika terdapat panggilan kepada komponen *process execution*, komponen tersebut akan menjalankan proses BPEL terkait. Proses BPEL terkait akan memanggil seluruh *web service* yang dibutuhkannya, yang berada pada *business service layer*. Sementara itu, jika sebuah *web service* pada *business service layer* membutuhkan sebuah *web service* dari *application service layer*, dia akan memanggilnya. Seluruh pemanggilan yang terjadi, baik ke *process execution* maupun ke *web service* akan menggunakan pesan berformat SOAP. Seluruh pemakaian teknologi pada model yang didefinisikan pada Bab 3.1 ini dapat dilihat pada Gambar III-3.



Gambar III-3 Ilustrasi Penggunaan Teknologi

3.2.3 Pemilihan BPMS

Untuk mengimplementasikan model penerapan BPM pada SOA seperti dijelaskan sebelumnya, terdapat berbagai BPMS yang dapat digunakan. Seperti telah diuraikan pada Bab 2.1.3.2, ada tiga BPMS yang telah dieksplorasi, yaitu Intalio BPMS Community Edition (CE), JBoss jBPM, dan OpenESB. Ketiga aplikasi ini dipilih disebabkan bersifat *open source* sehingga bebas untuk digunakan dan dimodifikasi sesuai kebutuhan. Dari ketiga BPMS ini akan dipilih salah satu sebagai kakas yang tepat untuk mengimplementasikan model.

Dari uraian sebelumnya, ketiga alternatif implementasi tersebut dapat dibandingkan satu sama lain. Intalio BPMS CE dapat mengimplementasikan model pada Gambar III-1 meski tidak memiliki semua komponen BPMS. Namun seperti telah dijelaskan pada Bab 3.1 tersebut, sebuah BPMS tidak perlu memiliki keempatnya. Intalio dapat memerankan BPM dari segi pemodelan, eksekusi dan monitoring. Sebab *process monitoring* pada Intalio BPMS CE dapat dilakukan oleh Intalio BPMS Server, meski terbatas hanya pada melihat proses apa saja yang sedang berjalan.

JBoss jBPM juga dapat mengimplementasikan model yang ditunjukkan pada Gambar III-1, namun jika menggunakan bahasa BPEL dalam implementasinya, maka tidak akan tersedia *process modeler*, hal ini berarti BPMS ini hanya akan mencakup proses monitor dan eksekusi saja. Pemodelan proses dengan BPEL dengan visualisasi dapat saja dilakukan namun dengan menggunakan kakas dari pihak ketiga.

Implementasi model menggunakan OpenESB dapat dilakukan, namun memerlukan implementasi sendiri untuk aspek monitoring, dengan menggunakan teknologi JMX. Jadi OpenESB hanya menyediakan *process modeling* dan *executing*. Hal ini disebabkan OpenESB bukan merupakan BPMS, tapi lebih ke teknologi ESB yang memungkinkan dibentuknya arsitektur SOA.

Seluruh perbandingan tersebut dapat dilihat pada Tabel III-1 Perbandingan Kelengkapan Ketiga BPMS.

Tabel III-1 Perbandingan Kelengkapan Ketiga BPMS

Alternatif	Modeling	Execution	Monitoring	Activity Monitoring
Intalio	Ada	Ada	Ada	Tidak Ada
jBoss BPM	Tidak ada	Ada	Ada	Tidak Ada
Open ESB	Ada	Ada	Tidak Ada	Tidak Ada

Setelah melihat perbandingan di atas, pada tugas akhir ini akan digunakan Intalio BPMS CE sebagai BPMS dalam studi kasus, disebabkan karena dukungannya terhadap BPEL dan fiturnya yang relatif lengkap untuk sebuah BPMS.

3.3 Metodologi Pembangunan

Seperti dijelaskan pada subbab I.5, pembangunan perangkat lunak yang menjadi studi kasus untuk tugas akhir ini akan menggunakan metodologi *unified process*. Ada empat fase dalam *unified process* yang dijalankan dengan sejumlah iterasi dalam masing-masing fase tersebut sesuai kebutuhan.

Namun, *unified process* adalah sebuah metodologi yang belum dipersiapkan untuk menangani komposisi *web services*, seperti telah dijelaskan pada Bab II.2.5. *Unified process* umumnya digunakan untuk mendekomposisi sebuah perangkat lunak dalam bentuk objek. Untuk arsitektur berbasis SOA, proses ini membutuhkan beberapa aktivitas tambahan, khususnya untuk mendefinisikan *service*. Karena secara umum perangkat lunak ini akan terbagi menjadi dua bagian, yaitu sekumpulan *service* dan aplikasi *client*.

Oleh karena itu, untuk metodologi pembangunan studi kasus, akan dilakukan penambahan dua aktivitas, yaitu analisis *service* dan perancangan *service*. Aktivitas dasar untuk *unified process* diambil berdasarkan referensi yang diperoleh dari [DES06]. Tambahan aktivitas diambil dari metode *SOA Delivery Lifecycle* pada Subbab II.2.5 yang sebelumnya didefinisikan oleh Thomas Erl di [ERL05]. Seperti telah dijelaskan pada bab tersebut, *SOA Delivery Lifecycle* mendeskripsikan langkah-langkah generik. Langkah-langkah ini perlu diadaptasi dengan metodologi pembangunan perangkat lunak yang telah biasa digunakan dalam organisasi yang membangun perangkat lunak tersebut.

Adapun aktivitas yang akan dilakukan pada pembangunan perangkat lunak berbasis SOA pada *unified process* dijelaskan pada Tabel III-2.

Tabel III-2 Keterhubungan Fase, Aktivitas dan Deliverables

Fase	Aktivitas	Deliverables
<i>Inception</i>	Analisis domain masalah	- Hasil analisis domain.
	Analisis sistem	- <i>Use case</i> sistem - Identifikasi proses bisnis sistem
<i>Elaboration</i>	Analisis domain masalah	- Revisi hasil analisis domain.
	Analisis <i>service</i>	- Revisi identifikasi proses bisnis - Identifikasi kandidat <i>service</i> - Identifikasi kandidat operasi <i>service</i>
	Analisis perangkat lunak <i>client</i>	- Diagram <i>use case client</i> - Skenario <i>use case</i> .
	Perancangan <i>service</i>	- Identifikasi <i>service</i> dan teknologinya - Rancangan proses bisnis dalam BPMN - Identifikasi operasi <i>service</i> - Diagram kelas perancangan - Diagram <i>deployment</i>

Fase	Aktivitas	Deliverables
	Perancangan perangkat lunak <i>client</i>	<ul style="list-style-type: none"> - Diagram <i>sequence</i> - Identifikasi kelas perancangan. - Perancangan antarmuka. - Diagram <i>deployment</i>
	Implementasi	<ul style="list-style-type: none"> - Kode program.
<i>Construction</i>	Perancangan <i>service</i>	<ul style="list-style-type: none"> - Revisi identifikasi <i>service</i> dan teknologi - Revisi rancangan proses bisnis dalam BPMN - Revisi identifikasi operasi <i>service</i> - Revisi <i>diagram deployment</i>
	Perancangan perangkat lunak <i>client</i>	<ul style="list-style-type: none"> - Revisi <i>sequence diagram</i> - Revisi identifikasi kelas perancangan. - Revisi perancangan antarmuka. - Revisi <i>diagram deployment</i>.
	Implementasi perangkat lunak	<ul style="list-style-type: none"> - Revisi kode program.
	Pengujian terintegrasi	<ul style="list-style-type: none"> - Evaluasi pengujian.
<i>Transition</i>	Persiapan instalasi	<ul style="list-style-type: none"> - <i>Installer</i>/panduan instalasi dan aplikasi