Prosiding Seminar Nasional Manajemen Teknologi XIII
Program Studi MMT-ITS, Surabaya 5 Pebruari 2011

MODEL DESAIN BPEL UNTUK KOMPOSISI WEB SERVICE DALAM OPEN ESB

Ricky Eka Putra[1] , Bayu Adhi Nugroho[2]
[1]Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Adhi Tama Surabaya
[2]Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
E-mail: [1]rickyeka25@yahoo.com , [2]nugroho09@mhs.if.its.ac.id

ABSTRAK

Penggunaan *web service* sebagai aplikasi teknologi implementasi dari konsep *Service Oriented Architecture* / SOA memiliki tuntutan mekanisme penggabungan (pembentukan proses *composite*) atau komposisi sehingga kumpulan service tung- gal (*atomic*) dapat terbentuk menjadi proses yang bersifat *composite*. Dalam per- masalahan semacam ini telah umum digunakan perangkat lunak *Enterprise Service Bus* / ESB sebagai lingkungan mediasi komposisi *service* sehingga mampu bersifat *composite*. Akan tetapi perangkat lunak ESB tersebut umumnya memiliki format dokumen dan mekanisme yang bersifat *propertiary* untuk melaksanakan kompo- sisi dari *service*. Manuskrip ini berusaha memberikan solusi masalah umum yang sering muncul dalam komposisi *web service* menggunakan *Business Process Exe- cution Languange* dalam perangkat lunak Open ESB yaitu : komposisi sekuensial *web service*, penanganan data *array / collection / list*, *routing flow* parameter ma- sukan. Metode yang digunakan adalah melakukan pengujian empiris berupa *test case* pada Open ESB

Kata kunci: *BPEL, SOA, Open ESB,* komposisi*, web service.*

PENDAHULUAN

Business Process Execution Languange / BPEL merupakan usaha untuk melakukan standarisasi netralitas *vendor* dari *service* yang dirintis oleh onsorsium bernama OASIS, yang merupakan gabungan dari perusahaan perangkat lunak besar semacam Microsoft, IBM [7] dan kontributor lainnya [6] untuk membentuk standar *de facto* netralitas *web service* / *service*. Penggunaan BPEL untuk melakukan komposisi *web service* merupakan salah satu solusi untuk memenuhi kriteria terbentuknya arsitektur perangkat lunak secara SOA. Solusi masalah umum dari komposisi *ser- vice* menggunakan BPEL yang seharusnya bersifat *trivial* menjadi tidak transparan karena minimalnya metode baku yang teruji dan mampu menyajikan solusi bagi masalah umum komposisi *web service* dalam ESB. Model BPEL untuk memban- gun komposisi *web service* sampai saat ini masih memiliki sedikit data uji lapangan terhadap kasus / permasalahan berikut :

1. komposisi sekuensial *web service*
2. penanganan data *array / collection / list*
3. *routing flow* parameter masukan

Penelitian ini merupakan pengembangan dari penelitian sebelumnya [5], dengan melakukan perubahan kasus data penelitian dan adaptasi perangkat lunak ESB yang digunakan. Tujuan perubahan kasus data adalah karena jenis model kasus data apapun merupakan variabel yang tidak akan mempengaruhi proses penelitian sehingga dapat dilakukan generalisasi, sedangkan perubahan adaptasi ESB bertujuan untuk membuktikan keumuman model dari impelementasi BPEL terhadap varian perangkat lunak ESB yang berbeda.

Banyak pihak berusaha melakukan pengujian terhadap implementasi BPEL di lapangan antara lain dilakukan oleh [1] yang melakukan komparasi BPEL terhadap perangkat lunak ESB yang berbeda, atau oleh [3] yang melakukan *unit testing* ter- hadap BPEL. Model penelitian ini sebagai pengembangan dari [5], akan memiliki *impact* / *side effect* yang kurang lebih sama dengan [1], yaitu memperoleh data adaptasi BPEL pada perangkat lunak ESB yang berbeda. Metode pengujian kom- posisi *web service* dalam BPEL secara struktural dengan *unit testing* yang diperke- nalkan oleh [4] tidak menggunakan model data yang tersimpan dalam basis data, akan tetapi hanya menguji aliran komposisi *web service* dalam BPEL, sehingga penelitian ini yang didasarkan pada penelitian sebelumnya [5] memiliki kelebihan dalam penggunaan basis data sebagai sumber data yang digunakan oleh *web ser- vice* dalam BPEL dibandingkan yang terdapat dalam [4].

DASAR TEORI

Service Oriented Architecture

Menurut [2] arsitektur SOA adalah sistem terdistribusi yang memiliki tiga macam konsep berikut :
1. *Service*
2. Interopabilitas dengan melalui *Enterprise Service Bus*
3. *Loose Coupled*

Menurut [8] interaksi pesan untuk saling berkolaborasi dalam arsitektur SOA dapat dibagi menjadi dua macam yaitu :
1. Koreografi
   Di mana setiap pihak berperan serta dalam proses interaksi pesan.
2. Orkestrasi
   Di mana terdapat sebuah pihak tunggal sebagai pengatur alur pesan.

Sebuah arsitektur SOA yang tersusun dengan *Enterprise Service Bus* / ESB dan *Business Process Execution Language* / BPEL umumnya memiliki bentuk orkestrasi, karena ESB bertindak sebagai pihak tunggal pengatur aliran pesan.

Enterprise Service Bus

Sesuai dengan [2] sebuah arsitektur perangkat lunak dapat disebut SOA jika terhubung melalui *Enterprise Service Bus*. ESB merupakan infrastruktur yang memungkinkan interoperabilitas tingkat tinggi antar *service* dalam sistem terdis- tribusi sehingga semakin mudah untuk mendistribusikan proses bisnis melalui sis- tem (SOA*) dalam *platform* dan teknologi yang berbeda [2].

Business Process Execution Language

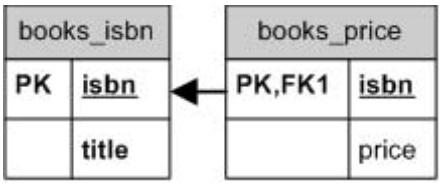
Business Process Execution Language / BPEL merupakan spesifikasi industri saat ini yang memperkenalkan standar orckestrasi dalam konteks *web service* [9]. Orkestrasi dalam BPEL memunculkan orkestrasi dalam dirinya sebagai *service*, yang disusun / dispesifikasikan dalam bahasa tingkat tinggi / *high level language* dan diimplementasikan dalam *engine* (ESB*)[9].

METODE PENELITIAN

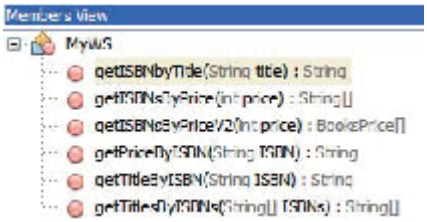
Penelitian ini menggunakan langkah / metode yang telah dikembangkan dari sumber asalnya [5], sebagai berikut :

ISBN : 978-602-97491-2-0

1. Konstruksi model basis data
   Untuk model basis data diimplementasikan dalam basis data relasional Post- greSQL dengan *library* Hibernate yang berfungsi sebagai pemetaan model relasional ke dalam model obyek. Kasus yang dipilih adalah model basis da- ta katalog buku berupa tiga atribut yaitu ISBN, judul buku dan harga buku.
2. Konstruksi *web service*
   Untuk lapisan *web service* diimplementasikan dalam bahasa pemrograman Java dengan menggunakan *library* JAX-WS versi 2.0
3. Konstruksi alur BPEL
   Untuk pembangunan alur BPEL digunakan perangkat lunak Open Source yaitu Open ESB / Glassfish ESB versi 2.2, di mana alur yang dibuat harus mencakup solusi masalah :



Gambar 1: Desain Model Entitas Basis Data



Gambar 2: Desain *Method Web Service*

- Komposisi sekuensial *web service*.
- Penanganan data *array / collection / list*.
- *Routing flow* parameter masukan.

Oleh karenanya dibentuk sebuah permasalahan yang berasal dari *domain* katalog buku berupa :

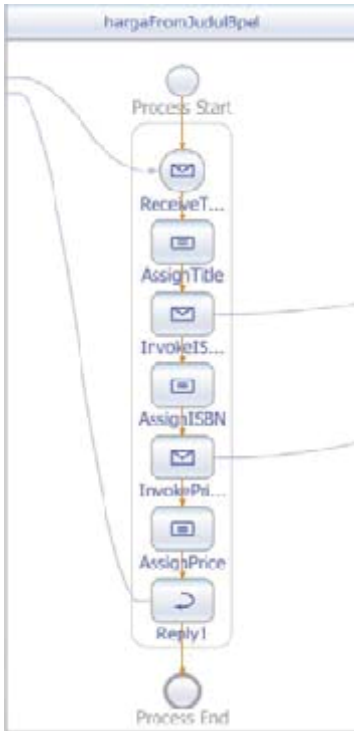
- Memunculkan harga buku melalui pemanggilan judul buku, merupakan permasalahan untuk kasus komposisi sekuensial, dimana untuk memu- nculkan harga buku melalui nama haruslah melewati proses pemanggi- lan nama menjadi ISBN.
- Memunculkan daftar ISBN untuk buku - buku yang memiliki harga sama, merupakan permasalahan untuk kasus *array*, dimana sangat mungkin muncul beberapa ISBN yang memiliki harga jual sama.
- Memunculkan hasil yang tepat berdasar parameter yang dimasukkan jika pemanggilan dilakukan dengan judul buku atau harga buku, sis- tem harus bisa melakukan identifikasi parameter dan mengarahkan ke *service* yang tepat, merupakan permasalahan untuk kasus *routing flow*.

4. Pengujian alur BPEL
   Untuk pengujian alur BPEL dibangun *test case* dengan *tool* yang terdapat pada Open ESB / Glassfish ESB versi 2.2

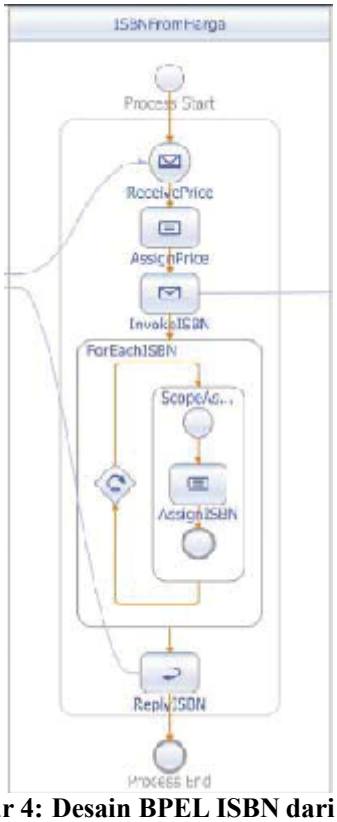
ISBN : 978-602-97491-2-0

DESAIN

Desain yang perlu dibuat dalam proses penelitian ini meliputi :

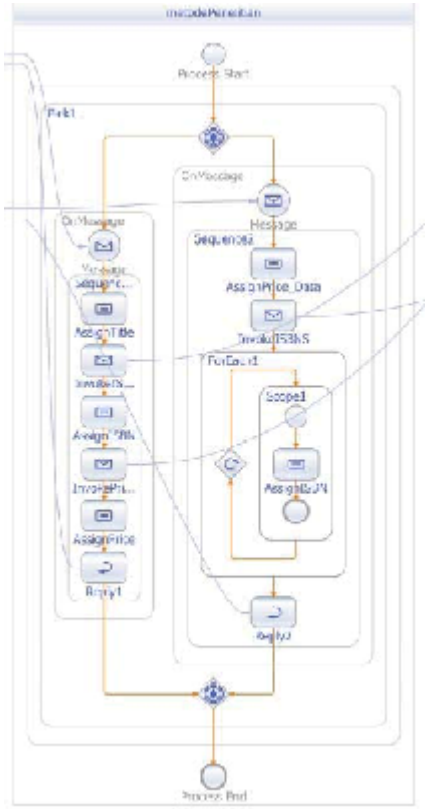


Gambar 3: Desain BPEL Harga dari Judul



Gambar 4: Desain BPEL ISBN dari Harga

ISBN : 978-602-97491-2-0



1. Desain model basis data
   Model entitas basis data yang digunakan dalam penelitian ini adalah sebagaimana terdapat pada gambar 1.
2. Desain *web service*
   Model *web service* yang dikembangkan adalah dengan *method* yang digambarkan sebagai *members* dari *class* seperti terdapat pada gambar 2.
3. Desain alur BPEL
   Model alur BPEL yang dikembangkan adalah seperti terdapat pada gam- bar 3 yang memodelkan memunculkan harga buku melalui pemanggilan judul buku melalui melalui komposisi dalam BPEL, gambar 4 yang mem- odelkan memunculkan daftar ISBN untuk buku - buku yang memiliki harga sama melalui inisialisasi *array* dalam BPEL dan gambar 5 yang memodelkan gabungan di antara keduanya melalui *routing* pesan dengan komponen Pick pada BPEL.

HASIL DAN DISKUSI

Berdasarkan pengujian yang dilakukan, untuk implementasi BPEL dengan de- sain model 3 dan 4 berhasil menjalankan kinerjanya sesuai dengan harapan, akan tetapi untuk desain model 5 masih terdapat pesan kesalahan. Terdapat kesulitan juga untuk melakukan pelacakan mengenai detail kesalahan yang terjadi, karena desain BPEL yang dibuat tidak melibatkan komponen *fault handler*. Dari testing lain yang dilakukan sebelumnya oleh penulis, dengan kasus yang berbeda peng- gunaan komponen Pick pada BPEL seperti terdapat pada desain model 5, terdapat kasus lain yang telah mampu berjalan, sedangkan untuk kasus yang dibangun pada manuskrip ini mengalami kegagalan.

ISBN : 978-602-97491-2-0

KESIMPULAN

Berdasarkan eksperimen yang dilaksanakan, dapat diambil kesimpulan bahwa perlu dikembangan mekanisme atau metodologi untuk melakukan *debugging* pada alur BPEL secara baku, sehingga dapat dilakukan deteksi kegagalan mekanisme kinerja BPEL.

DAFTAR PUSTAKA

[1] HALLWYL, T., HENGLEIN, F., AND HILDEBRANDT, T. A standard-driven implementaion of ws-bpel 2.0. In *Proceedings of the 2010 ACM Symposium on Applied Computing* (New York, NY, SA, 2010), SAC '10, ACM, pp. 2472–2476.

[2] JOSUTTIS, N. M. *SOA in Practice - The Art of Distributed System Design*. OReilly, 2007.

[3] LI, Z., SUN, W., JIANG, Z. B., AND ZHANG, X. Bpel4ws unit testing: Framework and implementation. In *Proceedings of the IEEE International Conference on Web Services* (Washington, DC, USA, 2005), ICWS '05, IEEE Computer Society, pp. 103–110.

[4] LIU, C.-H., CHEN, S.-L., AND LI, X.-Y. A ws-bpel based structural testing approach for web service compositions. In *Proceedings of the 2008 IEEE International Symposium on Service-Oriented System Engineering* (Washington, DC, USA, 2008), IEEE Computer Society, pp. 135–141.

[5] NUGROHO, B. A., AND SARNO, R. Implementing bpel compliant soa orchestration using biztalk server. In *Proceeding of the 6th International Conference on Information Communication Technology and Systems* (Gedung Teknik Informatika ITS - Jl. Raya ITS Keputih Sukolilo Surabaya Indonesia, September 2009), Informatics Department - Faculty of Information Technology – Sepuluh Nopember Institute of Technology Surabaya, pp. VIII–39.

[6] OASIS. Oasis contributors. http://www.oasis-open.org/about/ contributors.php, -.

[7] OASIS. Oasis foundational sponsors. http://www.oasis-open.org/ about/foundational_sponsors.php, -.

[8] PELTZ, C. Web services orchestration and choreography. *Computer 36* (2003), 46–52.

[9] VIROLI, M., DENTI, E., AND RICCI, A. Engineering a bpel orchestration engine as a multi-agent system. *Sci. Comput. Program. 66* (May 2007), 226–245.

ISBN : 978-602-97491-2-0