Johnson, R., A., and Wichren, D., W., 2002, *Applied Multivariate Statistical Analysis*, 5th Edition, Prentice Hall International Inc., New Jersey

Karson, M., J., 1982, *Multivariate Statistical Methods*, The Iowa State University Press, USA

Laporan Fakta Analisis, Penyusunan Rencana Tata Ruang Wilayah Provinsi Maluku, 2005, Bappeda Provinsi Maluku

Lebart, L., Morineau, A., & Warwick, K., M., 1984, *Multivariate Descriptive Statistical Analysis*, John Wiley & Sons, New York

Maluku Dalam Angka 2005/2006, 2006, Badan Pusat Statistika Provinsi Maluku

Morrison, 1990, *Multrivariate Statistical Methods*, McGraw Hill, Tokyo

Rencana Strategi Provinsi Maluku 2003 – 2008, 2003, Bappeda Provinsi Maluku

Sharma, S., 1996, *Applied Multivariate Techniques*, John Wiley & Sons, New York

Siswadi dan Suharjo, B., 1997, *Analisis Eksplorasi Data Peubah Ganda*, Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Institut Pertanian Bogor, Bogor

Stockburger, W., 1998, *Multivariate Statistics*: *Concepts*, *Models, and Applications*, http://www.psychstat.smsu.edu/multibook/mlt00.htm [16 Pebruari 2007]

Wattimanela, H.J., dkk, 2008, Analisis Potensi Berbasis Gugus Pulau di Provinsi Maluku, Lemlit Unpatti, Ambon

Wulder, M., 2002, *Multivariate Statistics*: A Practical Guide, http://www.pfc.forestry.ca/profiles/wulder/mvstats/-18k [16 Pebruari 2007]

T – 32

Pemodelan Komposisi Web Service Dengan Menggunakan Petri Net

Bandung Arry Sanjoyo[1)], Subiono[2)], dan Riyanarto Sarno[3)]
1) bandung@matematika.its.ac.id;bandungarry@yahoo.com;
Jurusan Matematika FMIPA ITS
2) subiono@matematika.its.ac.id; Jurusan Matematika FMIPA ITS
3) riyanarto@gmail.com; Jurusan T. Informatika FTIf ITS

ABSTRAK

Web engineering dapat dipandang sebagai penerapan software engineering pada lingkungan web yang memiliki aspek utama orientasi menuju sistem terdistribusi. Saat ini, aplikasi lingkungan web menjadi andalan beberapa instansi/perusahaan dalam mengelola proses bisnis.

Dalam automasi berbasis Internet, proses binis merupakan sekumpulan / penggabungan dari web service. Komposisi web service membutuhkan suatu mekanisme penggabungan agar bisa membentuk sebuah workflow untuk merepresentasikan bisnis proses. Dibahas suatu metoda untuk mengkaji semantic web service agar sesuai dengan kebutuhan pengguna. Model matematika graf bentuk Petri Net akan dipakai untuk memodelkan komposisi web service dan dipakai untuk melakukan verifikasi dan evaluasi kinerja sistem melalui matriks incidence.

Kata kunci : web service, komposisi web service, Petri net, matriks incidence

PENDAHULUAN

Pesatnya perkembangan pasar BPM (Business Process Management) telah menjadi tren baru di industri perangkat lunak. BPM merupakan perangkat lunak untuk membantu perusahaan dalam mengelola proses bisnis yang dimilikinya mulai dari tahap perancangan, otomasi (komputerisasi), eksekusi, hingga tahap monitoring. Dengan BPM, seorang business analyst dapat membuat berbagai skenario simulasi untuk menguji performansi sebuah proses.

Manajemen website dan kemampuan desain merupakan dua unsur kuat dalam bidang ilmu informasi. Untuk mencapai manajemen website yang lebih baik dan kemampuan desain yang handal, para business analyst memeriksa frekuensi, durasi dan bahkan melalui profil penggunaan Web. Oleh karena itu, topik penelitian web service menjadi sangat dibutuhkan.

Kumpulan web service yang mendukung sebuah bisnis proses, membutuhkan mekanisme penggabungan agar bisa membentuk sebuah workflow untuk merepresentasikan suatu bisnis proses. Mekanisme penggabungan web service tersebut dibedakan menjadi dua yaitu orkestrasi dan koreografi (Viroli et al, 2007). Metode

semantik digunakan untuk mendapatkan web service yang sesuai dengan kebutuhan pengguna.

Model Petri net akan digunakan untuk menemukan komposisi yang handal dari rangkaian web service. Dan analisis kinerja sistem dapat dilakukan melalui matrix incidence untuk model Petri net tersebut.

Perumusan Masalah

Permasalahan yang diangkat pada penelitian ini adalah: bagaimana menentukan rangkaian komposisi web service yang handal dengan menggunakan operasi aljabar pada Petri Net untuk web service.

Tujuan Penelitian

Aktivitas penelitian ini mempunyai tujuan melakukan pemodelan komposisi semantik web service pada sistem bisnis dengan menggunakan Petri Net, sehingga proses dapat dianalisis lebih jauh melalui model Petri net.

METODE PENELITIAN

Langkah-langkah yang dilakukan dalam penelitian ini adalah:

1. Pemodelan Komposisi Web Service ke dalam Model Petri Net
2. Analisis komposisi Web service melalui aljabar Web service.

HASIL DAN PEMBAHASAN

Petri Net

Petri net merupakan salah satu alat untuk memodelkan sistem event diskrit dimana harga variabel-variabel sistem hanya berada pada dua keadaan seperti aktif atau tidak aktif. Petri net memuat *event* berkaitan dengan transisi. Agar suatu *event* dapat terjadi, beberapa keadaan harus terpenuhi. Informasi mengenai *event* dan keadaan ini masing-masing dinyatakan dengan transisi dan *place*. *Place* dapat berfungsi sebagai input atau output suatu transisi. *Place* sebagai input menyatakan keadaan yang harus dipenuhi agar transisi dapat terjadi. Setelah transisi terjadi maka keadaan akan berubah. *Place* yang menyatakan keadaan tersebut adalah output dari transisi.

Definisi 1 (Cassandras**,** 2008)**:** Petri net adalah pasangan 4-tuple $(\boldsymbol{P}, \boldsymbol{T}, \boldsymbol{A}, \boldsymbol{w})$ dengan

- P adalah himpunan berhingga *place*, $P = \{p_1, p_2, \dots, p_n\}$,
- T adalah himpunan berhingga transisi, $T = \{t_1, t_2, \dots, t_n\}$,

- A adalah himpunan *arc*, $A \subseteq (P \times T) \cup (T \times P)$,
- w adalah fungsi bobot, $w: A \to \{1, 2, 3, \dots\}$.

Petri net dapat direpresentsikan dalam bentuk graf dengan aturan sebagai berikut:

- $\forall p_i \in P$ digambarkan dalam graf menjadi node lingkaran berlabel p_i.
- $\forall t_i \in T$ digambarkan dalam graf menjadi node segiempat berlabel t_i.
- $\forall (p_i, t_j) \in A$ digambarkan dalam graf menjadi garis berarah dari node p_i ke node t_j.
- Bobot pada arc merupakan label dari garis untuk arc tersebut.
- Petri net dapat digambarkan sebagai graph berarah.

Transisi pada Petri net menyatakan ***event*** pada sistem event diskrit dan *place* merepresentasikan **kondisi** agar *event* dapat terjadi.

Input dan output untuk node transisi disimbolkan sebagai berikut:

- $I(t_j) = \{\mathrm{p_i} | (\mathrm{p_i}, \mathrm{t_j}) \in \mathrm{A}\}$ (1)
- $O(t_j) = \{\mathrm{p_i} | (\mathrm{t_j}, \mathrm{p_i}) \in \mathrm{A}\}$ (2)

Token adalah sesuatu yang diletakkan di place yang menyatakan terpenuhi tidaknya suatu kondisi. Pada graf, token digambarkan dengan *dot* dan diletakkan di dalam node untuk place. Jika jumlah token besar maka dituliskan dengan angka.

Definisi 2: Penanda (*marking*) x pada Petri net adalah fungsi $x: P \to \{0, 1, 2, \dots\}$. Penanda dinyatakan dengan vektor yang berisi bilangan bulat tak negatif yang menyatakan jumlah token yaitu $\mathbf{x} = [x(p_1), x(p_2), \dots, x(p_n)]^T$. Banyaknya elemen $\mathbf{x}$ sama dengan banyak place di Petri net. Elemen ke-i pada vektor $\mathbf{x}$ merupakan jumlah token pada place p_i, $x(p_i) \in \{0, 1, 2, \dots\}$.

Definisi 3: Petri net bertanda (*marked*) adalah pasangan 5-tuple $(P, T, A, w, \mathbf{x}_0)$ dimana (P, T, A, w) adalah Petri net dan $\mathbf{x}_0$ adalah penanda awal.

Sedangkan keadaan (*state*) pada Petri net didefinisikan sebagai berikut.

Definisi 4: Keadaan (*state*) pada Petri net bertanda adalah fungsi $\mathbf{x} = [x(p_1), x(p_2), \dots, x(p_n)]^T$. Ruang keadaan (*state space*) X pada Petri net bertanda dengan *n* place didefinisikan oleh semua vektor berdimensi *n* dengan elemen bilangan bulat tak negatif, sehingga $\mathbf{X} = \{0, 1, 2, \dots\}^n$.

Jika semua keadaan yang diperlukan sudah terpenuhi maka transisi dapat terjadi. Dalam hal ini keadaan merupakan place input dari transisi. Bobot arc dari place input ke

transisi menunjukkan jumlah token minimum di place agar transisi *enabled*. Jika semua place input mempunyai token lebih dari atau sama dengan jumlah token minimum yang dibutuhkan maka *transisi enabled*.

Definisi 5: Transisi $t_j \in T$ pada Petri net bertanda dikatakan *enabled* jika $x(p_i) \geq w(p_i, t_j),\ \forall p_i \in I(t_j)$.

Dalam pemodelan sistem dinamik event diskrit, Petri net dilengkapi dengan mekanisme untuk menjalankan token melewati jaringan (net) ketika transisi menjadi enabled dan proses ini akan mengubah keadaan Petri net.

Hanya transisi enabled yang dapat dilakukan *fire*. Transisi *difire* saat event yang dinyatakan oleh transisi terjadi. Berikut ini adalah proses yang terjadi pada *firing* transisi. Semua token di place input berkurang sebanyak bobot arc yang menghubungkannya. Dan token pada place output bertambah bobot arc yang menghubungkannya.

Definisi 6: Fungsi perubahan keadaan pada Petri net bertanda $(P, T, A, w, \mathbf{x}_0)$ yaitu $f : \{0, 1,\ 2,\ \ldots\}^n \times T \rightarrow \{0, 1,\ 2,\ \ldots\}^n$ terdefinisi untuk transisi $t_j \in T$ jika dan hanya jika

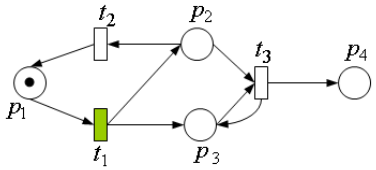
$$x(p_i)\ \geq\ w(p_i, t_j), \forall p_i \in I(t_j) \tag{3}$$

Jika $f(\mathbf{x}, t_j)$ terdefinisi maka ditulis $\mathbf{x}' = f(\mathbf{x}, t_j)$, dimana

$$x'(p_i) = x(p_i) - w(p_i, t_j) + w(t_j, p_i),\ \ i = 1, 2, \ldots\ , n \tag{4}$$

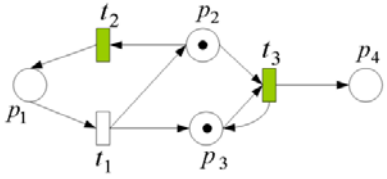
Berdasarkan (4), jika p_i adalah place input untuk transisi t_j, maka token pada place p_i berkurang sebanyak bobot arc dari p_i ke t_j . Jika p_i adalah place output dari transisi t_j, maka token pada place p_i bertambah sebesar bobot arc dari t_j ke p_i. Dan Jika p_i adalah place input dan output dari transisi t_j, maka token pada place p_i berkurang sebanyak $w(p_i, t_j)$ dan bertambah sebanyak $w(t_j, p_i)$.

Proses *firing* diilustrasikan menggunakan Petri net Gambar 1. Transisi t_1 merupakan transisi yang *enabled* karena $x(p_1) \geq w(p_1, t_1)$, transisi yang *enabled* digambarkan dengan warna hijau. Penanda awal dari Petri net ini adalah $\mathbf{x_0} = [1,0,0,0]^T$.



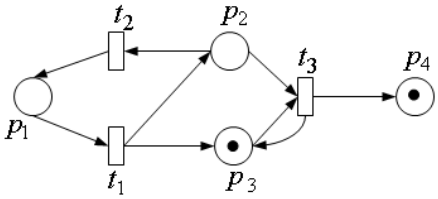
Gambar 1. Bentuk Petri net dengan transisi t_1 *enabled* dan t_2, t_3 tidak *enabled*.

Jika dilakukan proses *firing* pada transisi t_1, maka penanda dari Petri net berubah menjadi $\mathbf{x}' = f(\mathbf{x_0}, t_1) = [0,1,1,0]^T$, dan Petri net berubah menjadi Gambar 2.



Gambar 2. Bentuk Petri net setelah proses *firing* transisi t_1.

Selanjutnya, jika dilakukan proses *firing* pada transisi t_3, maka penanda dari Petri net berubah menjadi $\mathbf{x}'' = f(\mathbf{x}', t_3) = [0,0,1,1]^T$, dan Petri net berubah menjadi Gambar 3.

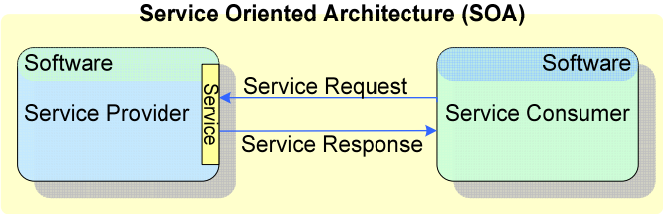


Gambar 3. Bentuk Petri net setelah proses firing transisi t_1 dan kemudian t_3.

Petri net pada Gambar 3, tidak memuat node transisi yang *enabled*. Keadaan seperti ini dikatakan Petri net mengalami ***deadlock*** disebut keadaan ***terminal*** dari Petri net.

Web Service

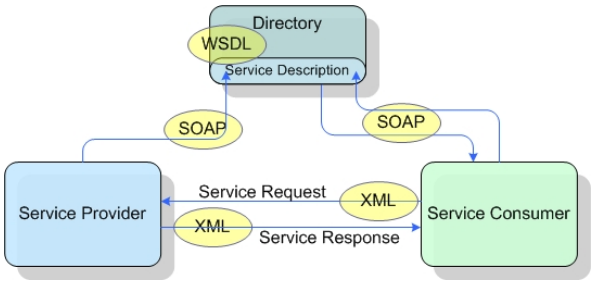
Misal sebuah *software* yang bernama *service consumer* ingin berkomunikasi dengan *software* lain yang bernama *service provider*. *Service consumer* sedang mengirim pesan permintaan layanan (*sevice request*) ke *service provider*, dan *service provider* menjawab dengan mengirimkan respon layanan (*service response*) ke *service consumer*. Mekanisme komunikasi seperti ini digambarkan pada Gambar 4.



Gambar 4. Skema SOA

Pada saat perangkat lunak *service provider* menerima permintaan layanan, permintaan layanan ini diproses oleh sebuah *service*. *Service* adalah suatu fungsi yang terdefinisi dengan baik dan tidak tergantung pada keadaan *service* lainnya. Perangkat lunak *service consumer* perlu tahu cara untuk memanggil *service* tersebut dan apa harapan dari respon *service provider*. Kerangka mekanisme komunikasi antar software seperti di atas dinamakan *Service Oriented Architecture* (SOA). Implementasi dari SOA dinamakan *web services*.

W3C mendefinisikan *web service* sebagai sebuah sistem software yang dirancang untuk mendukung interaksi (*interoperable*) mesin-ke-mesin melalui jaringan. *Web service* memiliki antarmuka dalam format *machine-processable* (khususnya Web Services Description Language WSDL). Sistem lain berinteraksi dengan *web service* dengan pesan SOAP (*Simple Object Acess Protocol*), biasanya disampaikan menggunakan HTTP dengan serialisasi XML (*Extensible Markup Languages*) dalam hubungannya dengan standar Web.



Gambar 5 Web Services Achitecture

Definisi formal tentang Web service dapat dinyatakan sebagai berikut, [Hamadi, 2003].

Definisi 7: (Service Net)
Service net adalah pasangan tuple $SN = (P, T, W, i, o, l)$, dimana:
- P merupakan himpunan berhingga *place*.
- T merupakan himpunan berhingga transisi yang merupakan operasi dari servis.
- $W \subseteq (P \times T) \cup (T \times P)$ yang merupakan himpunan busur berarah.
- i merupakan place input dengan, $\bullet\, i = \{x \in P \cup T | (x, i) \in W\} = \emptyset$
- o merupakan place output dengan, $o \bullet = \{x \in P \cup T | (o, x) \in W\} = \emptyset$
- $l: T \to \mathcal{A} \cup \{\tau\}$ merupakan fungsi pelabelan dengan $\mathcal{A}$ suatu himpunan nama-nama operasi dan $\tau \notin \mathcal{A}$ adalah operasi diam.

Definisi 8: (Web Service)

Web service adalah sepasang tuple $S = (NameS, Desc, Loc, URL, CS,\ SN)$, dimana:

- $NameS$ merupakan nama dari layanan (*service*), mempunyai identifier tunggal.
- $Desc$ merupakan deskripsi dari layanan yang disediakan. Berupa ringkasan tentang layanan yang disediakan.
- Loc adalah server tempat layanan disimpan.
- URL merupakan nama alamat pemanggilan Web service,
- CS merupakan himpunan dari komponen service. Jika $CS = NameS$ maka S merupakan service dasar. Selain itu S merupakan *composite service*.
- $SN = (P, T, W, i, o)$ merupakan pemodelan *service net. Place i* merupakan sebuah *place* sebagai tanda awal dari service S. Eksekusi S dimulai ketika sebuah token berada di *place* i dan berakhir ketika token mencapai *place* o.

Komposisi Web Service

Proses pemodelan Petri net dari komposisi *Web service* dilakukan dengan cara melakukan operasi Sequence, Alternatif, Arbitrary Sequence, Paralel dengan komunikasi, dan Diskriminator dari beberapa bentuk Petri net dari *Web service* dasar

Himpunan services dapat didefinisikan dalam bentuk notasi grammer berikut ini.

$S ::= \epsilon \mid X \mid S \odot S \mid S \oplus S \mid S \diamondsuit S \mid \mu S \mid S \parallel_c S \mid (S|S) \rightsquigarrow S \mid [S(p,q) : S(p,q)]$

dengan

- ϵ merupakan service kosong atau sebuah service yang tidak memuat operasi.
- X merupakan konstanta service yang digunakan sebagai atomik atau service dasar.
- $\odot, \oplus, \diamondsuit, \mu, \parallel_c, \rightsquigarrow$ merupakan operator-operator yang bekerja pada Web service untuk membentuk komposisi Web Service.

Dari definisi Web service, n buah service $S_1, S_2, \dots, S_n$ dapat dituliskan sebagai berikut.

$S_i = (NameS_i, Desc_i, Loc_i, URL_i, CS_i, SN_i)$ dengan

- $SN_i = (P_i, T_i, W_i, i_i, o_i)$ untuk $i = 1, \dots, n$
- $P_i \cap P_j = \emptyset$ dan $T_i \cap T_j = \emptyset$ untuk $i \neq j$.

Sebuah Web service dapat berupa service kosong ataupun service atomik.

Web service Kosong

Web service kosong berbentuk $\varepsilon = (NameS, Desc, Loc, URL, CS,\ SN)$
dimana:

- $NameS$ = Empty, $Desc$ = "*Empty Web Service*", Loc adalah server tempat layanan disimpan, URL =Null, yang menyatakan bahwa ε tidak memiliki URL, CS = {Empty}, $SN = (\{p\}, \emptyset, \emptyset, p, p, \emptyset)$.

Secara grafis, Web service kosong dimodelkan dalam Petri net pada Gambar 6.

Gambar 6. Web Service Kosong, $\boldsymbol{\varepsilon}$

Komposisi Web Service dengan Operator Sequence

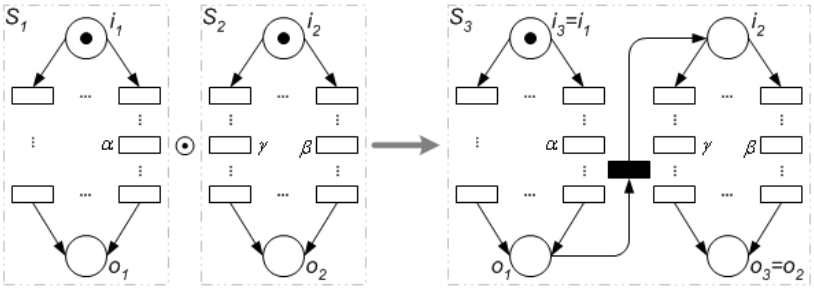
Operator *sequence* pada Web service S_1 dan S_2 disimbolkan dengan $S_1 \odot S_2$ dan menghasilkan suatu Web service S_3 yang melakukan eksekusi Web service S_1 dan dilanjutkan dengan eksekusi Web service S_2. Eksekusi S_2 dilaksanakan setelah eksekusi S_1 selesai. Secara definisi Web service

$S_3 = S_1 \odot S_2 = (NameS_3, Desc_3, Loc_3, URL_3, CS_3, SN_3 = (P_3, T_3, W_3, i_3, o_3, l_3))$

dimana:

- $NameS_3$ = nama service S_3, $Desc_3$ = Diskripsi dari service S_3, Loc_3 adalah server tempat service S_3 disimpan (bisa tidak sama dengan Loc_1 atau Loc_2).
- URL_3 = permintaan service S_3, $CS_3 = CS_1 \cup CS_2$, $P_3 = P_1 \cup P_2$, $\mathrm{T_3 = T_1 \cup T_2 \cup \{t\}}$
- $\mathrm{W_3 = W_1 \cup W_2 \cup \{(o_1, t), (t, i_2)\}}$, $\mathrm{i_3 = i_1}$, $\mathrm{o_3 = o_2}$, $\mathrm{l_3 = l_1 \cup l_2 \cup \{t, \tau\}}$

Model Petri net untuk komposisi Web service menggunakan operator *sequence* digambarkan pada Gambar 7.



Gambar 7 Komposisi Web Service Menggunakan Operastor *Sequence*

Segiempat warna hitam adalah transisi τ yang merupakan operasi diam.

Komposisi Web Service dengan Operator Sequence

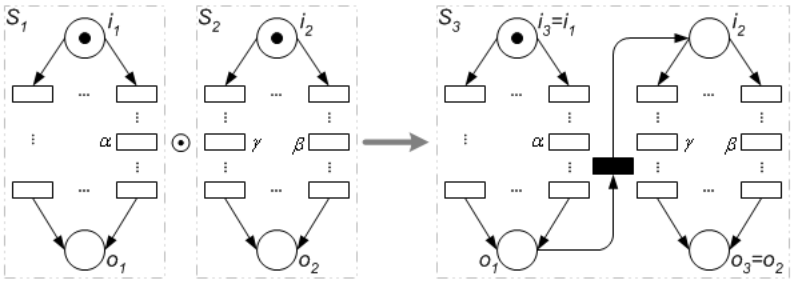
Operator *sequence* pada Web service S_1 dan S_2 disimbolkan dengan $S_1 \odot S_2$ dan menghasilkan suatu Web service S_3 yang melakukan eksekusi Web service S_1 dan dilanjutkan dengan eksekusi Web service S_2. Eksekusi S_2 dilaksanakan setelah eksekusi S_1 selesai. Secara definisi Web service dituliskan sebagai berikut.

$S_3 = S_1 \odot S_2 = (NameS_3, Desc_3, Loc_3, URL_3, CS_3, SN_3 = (P_3, T_3, W_3, i_3, o_3, l_3))$

dimana:

- $P_3 = P_1 \cup P_2$, $\mathrm{T_3 = T_1 \cup T_2 \cup \{t\}}$, $\mathrm{W_3 = W_1 \cup W_2 \cup \{(o_1, t), (t, i_2)\}}$, $\mathrm{i_3 = i_1}$, $\mathrm{o_3 = o_2}$, dan $\mathrm{l_3 = l_1 \cup l_2 \cup \{t, \tau\}}$.

Model Petri net untuk komposisi Web service menggunakan operator *sequence* digambarkan pada Gambar 8.



Gambar 8 Komposisi Web Service Menggunakan Operastor *Sequence*

Komposisi Web Service dengan Operator Alternatif

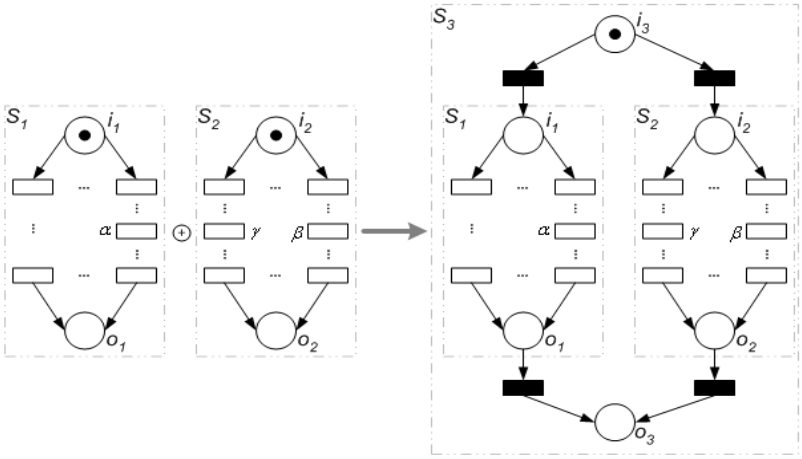
Operator alternatif pada Web service S_1 dan S_2 disimbolkan dengan $S_1 \oplus S_2$ dan menghasilkan suatu Web service S_3 yang melakukan eksekusi Web service S_1 atau eksekusi Web service S_2.

$S_3 = S_1 \oplus S_2 = (NameS_3, Desc_3, Loc_3, URL_3, CS_3, SN_3 = (P_3, T_3, W_3, i_3, o_3, l_3))$

dimana:

- $P_3 = P_1 \cup P_2 \cup \{i_3, o_3\}$, $\mathrm{T_3} = \mathrm{T_1} \cup \mathrm{T_2} \cup \{t_{i1},\ t_{i2}, t_{o1}, t_{o2}\}$,
- $\mathrm{W_3} = \mathrm{W_1} \cup \mathrm{W_2} \cup \{(i_3, t_{i1}), (i_3, t_{i2}), (t_{i1}, i_1), (t_{i2}, i_2), (o_1, t_{o1}), (o_2, t_{o2}), (t_{o1}, \mathrm{o_3}), (t_{o2}, \mathrm{o_3})\}$
- $\mathrm{l_3} = \mathrm{l_1} \cup \mathrm{l_2} \cup \{(\mathrm{t}_{i1}, \tau), (\mathrm{t}_{i2}, \tau), (\mathrm{t}_{o1}, \tau), (\mathrm{t}_{o2}, \tau)\}$

Model Petri net untuk komposisi Web service menggunakan operator alternatif digambarkan pada Gambar 9.



Gambar 9. Komposisi Web Service Menggunakan Operator Alternatif

Komposisi Web Service dengan Operator Arbitrary Sequence

Operator arbitrary sequence pada Web service S_1 dan S_2 disimbolkan dengan $S_1 \diamondsuit S_2$ dan menghasilkan suatu Web service S_3 yang melakukan eksekusi Web service S_1 diikuti service S_2 atau eksekusi Web service S_2 diikuti service S_1.

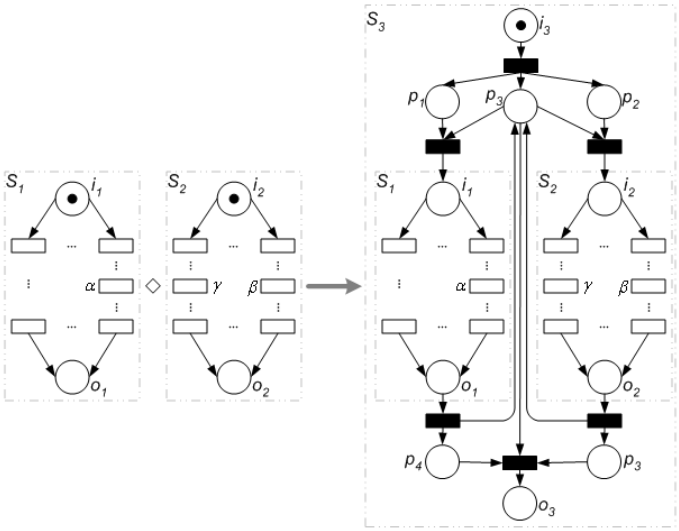
$S_3 = S_1 \diamondsuit S_2 = (NameS_3, Desc_3, Loc_3, URL_3, CS_3, SN_3 = (P_3, T_3, W_3, i_3, o_3, l_3))$

dimana:

- $P_3 = P_1 \cup P_2 \cup \{i_3, o_3, p_1, p_2, p_3, p_4, p_5\}$, $\mathrm{T}_3 = \mathrm{T}_1 \cup \mathrm{T}_2 \cup \{t_i, t_1, t_2 t_3 t_4, t_o\}$,
- $\mathrm{W}_3 = \mathrm{W}_1 \cup \mathrm{W}_2 \cup$

  $\{(i_3, t_i), (t_i, p_1), (t_i, p_2), (t_i, p_3), (p_1, t_1), (p_2, t_2), (p_3, t_1), (p_3, t_2),$
  $(p_3, t_o), (t_1, i_1), (t_2, i_2), (o_1, t_3), (o_2, t_4), (t_3, \mathrm{p}_3), (t_4, p_3), (t_3, p_4),$
  $(t_4, p_5), (p_4, t_\mathrm{o}), (p_5, t_\mathrm{o}), (t_o, o)\}$
- $\mathrm{l}_3 = \mathrm{l}_1 \cup \mathrm{l}_2 \cup \{(\mathrm{t}_i, \tau), (\mathrm{t}_1, \tau), (\mathrm{t}_2, \tau), (\mathrm{t}_3, \tau), (\mathrm{t}_4, \tau), (\mathrm{t}_o, \tau)\}$

Model Petri net untuk komposisi Web service menggunakan operator abitrary sequence digambarkan pada Gambar 10.



Gambar 10. Komposisi Web Service Menggunakan Operator Arbitrary Sequence

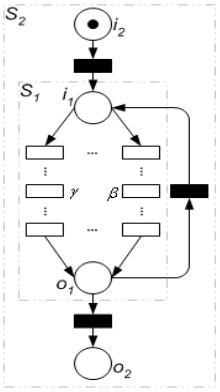
Komposisi Web Service dengan Operator Iterasi

Operator iterasi pada Web service S_1 disimbolkan dengan μS_1 dan menghasilkan suatu Web service S_2 yang melakukan eksekusi Web service S_1 berkali-kali.

$S_2 = \mu S_1 = (NameS_2, Desc_2, Loc_2, URL_2, CS_2, SN_2 = (P_2, T_2, W_2, i_2, o_2, l_2))$, dimana:

- $P_2 = P_1 \cup \{i_2, o_2\}$, $T_2 = T_1 \cup \{t_i, t_o, t_2\}$, $l_2 = l_1 \cup \{(t_i, \tau), (t_O, \tau), (t_2, \tau)\}$
- $W_2 = W_1 \cup \{(i_2, t_i), (t_i, i_1), (o_1, t_o), (t_o, o), (o_1, t_2), (t_2, i_1)\}$

Model Petri net untuk komposisi Web service menggunakan operator iterasi digambarkan pada Gambar 11.



Gambar 11. Komposisi Web Service Menggunakan Operator Iterasi

Komposisi Web Service dengan Operator Paralel dengan Komunikasi

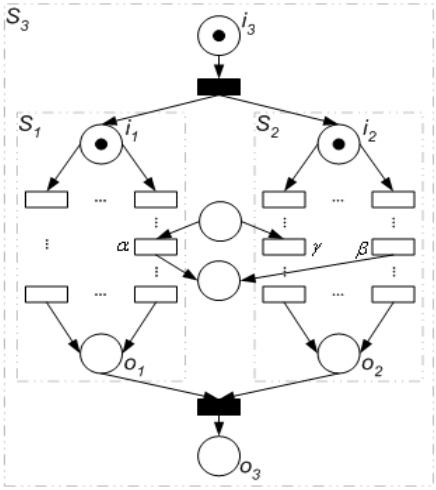
Operator paralel dengan komunikasi pada Web service S_1 dan Web service S_2 disimbolkan dengan $S_1 \|_c S_2$ atau S_3 merupakan proses eksekusi Web service S_1 dan Web service S_2 secara bersamaan. Pada saat eksekusi bersamaan ini, dimungkinkan adanya sinkronisasi dan pertukaran informasi.

$S_3 = S_1 \|_c S_2 = (NameS_3, Desc_3, Loc_3, URL_3, CS_3, SN_3 = (P_3, T_3, W_3, i_3, o_3, l_3))$

dimana:

- $P_3 = P_1 \cup P_2 \cup \{i_3, o_3\} \cup \{p_i | (\alpha_i, \beta_i) \in \mathcal{C}\}, T_3 = T_1 \cup T_2 \cup \{t_i, t_o\}$
- $W_3 = W_1 \cup W_2 \cup \{(i_3, t_i), (t_i, i_1), (t_i, i_2), (o_1, t_o), (o_2, t_o), (t_o, o_3)\}$
  $\cup \{(\alpha_i, p_i), (p_i, \beta_i) | (\alpha_i, \beta_i) \in \mathcal{C}\}$
- $l_3 = l_1 \cup l_2 \cup \{(t_{i1}, \tau), (t_{i2}, \tau), (t_{o1}, \tau), (t_{o2}, \tau)\}$
- $\mathcal{C} = \{(\alpha, \beta) | (\alpha, \beta) \in T_1 \times T_2 \cup T_1 \times T_2\}$ merupakan himpunan elemen komunikasi.

Model Petri net untuk komposisi Web service menggunakan operator paralel dengan komunikasi digambarkan pada Gambar 11.



Gambar 11. Komposisi Web Service: Paralel dengan Komunikasi

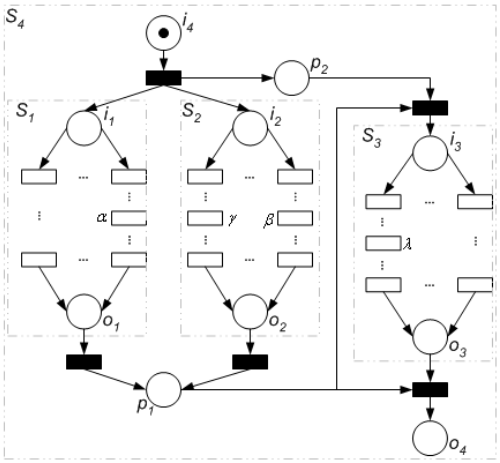
Komposisi Web Service dengan Operator Diskriminator

Operator diskriminator pada Web service S_1, Web service S_2, dan Web service S_3 disimbolkan dengan $(S_1|S_2) \rightsquigarrow S_3$ atau S_4 merupakan proses eksekusi Web service S_1 dan Web service S_2 secara paralel tanpa komunikasi, dan bisa dilanjutkan proses Web service S_3.

$S_4 = (S_1|S_2) \rightsquigarrow S_3 = (NameS_4, Desc_4, Loc_4, URL_4, CS_4, SN_4)$, dimana:

- $SN_4 = (P_4, T_4, W_4, i_4, o_4, l_4)$
- $P_4 = P_1 \cup P_2 \cup P_3 \cup \{i_4, o_4, p_1, p_2\}$
- $T_4 = T_1 \cup T_2 \cup T_3 \cup \{t_i, t_1, t_2, t_3, t_o\}$
- $W_4 = W_1 \cup W_2 \cup W_3 \cup \{(i_4, t_i), (t_i, i_1), (t_i, i_2), (t_i, p_2), (o_1, t_1), (o_2, \mathrm{t}_2),$
  $(t_1, p_1), (t_2, p_1), (p_1, t_3), (p_1, t_o), (p_2, t_3), (t_3, \mathrm{i}_3), (o_3, t_o), (t_o, \mathrm{o}_4)$
- $l_4 = l_1 \cup l_2 \cup l_3 \cup \{(t_i, \tau), (t_1, \tau), (t_2, \tau), (t_3, \tau), (t_o, \tau)\}$

Model Petri net untuk komposisi Web service menggunakan operator diskriminator digambarkan pada Gambar 12.



Gambar 12. Komposisi Web Service: Operator Diskriminasi

Liveness Web Service Dasar

Sebuah Web service dapat dinyatakan dalam bentuk sebuah model pasangan 6 buah tuple, dengan komponen tuplenya adalah: Nama dari layanan (service), mempunyai identifier tunggal, Deskripsi dari layanan yang disediakan. Berupa ringkasan tentang layanan yang disediakan, Nama server tempat layanan disimpan, Nama alamat URL untuk pemanggilan Web service, Himpunan dari komponen service yang dipakai untuk membedakan service dasar dan servise komposisi, Sebuah Service net yang merupakan bentuk khusus dari model Petri net, yaitu sebuah Petri net yang mempunyai place awal dan place akhir.

Dengan menggunakan Service net ini, setiap Web service dasar (Web service kosong dan konstan) dapat dimodelkan dengan mudah/sederhana dalam bentuk Petri net yang setiap arc-nya berbobot 1. Vektor keadaan awal dari Service net adalah $\mathbf{x_o} = [1,0,0,\ldots,0]$ dengan banyaknya komponen $\mathbf{x_o} = |P|$**.** Oleh karena itu, liveness dari setiap transisi pada Service net tidak *dead*. Atau dari keadaan awal selalu dapat dilakukan proses *firing* paling tidak sekali sampai dengan keadaan terminal. Sehingga setiap transisi pada Service net adalah *L1-live*.

Petri net dari Web service kosong tidak mempunyai transisi, oleh karena itu live.

Liveness Web Service Komposisi

Jika Web service S_1 dan S_2 mempuyai bentuk Petri net yang *L1-live,* maka komposisi Web service yang memuat operator Sequence, Alternatif, Arbitrary Sequence, Paralel dengan komunikasi, dan Diskriminator juga mempuyai bentuk Petri net yang *L1-live.*

SIMPULAN DAN SARAN

Komposisi Web service dapat dibentuk dari mengoperasikan beberapa Web service dasar dengan menggunakan operator Sequence, Alternatif, Arbitrary Sequence, Paralel dengan komunikasi. Makna semantik dari komposisi Web service dinyatakan dalam bentuk Petri net. Dengan demikian, layanan komposisi dinyatakan dengan menggunakan konstruksi Petri net.

Jika Web service S_1 dan S_2 mempuyai bentuk Petri net yang *L1-live,* maka komposisi Web service yang memuat operator Sequence, Alternatif, Arbitrary Sequence, Paralel dengan komunikasi juga mempuyai bentuk Petri net yang *L1-live.*

Perlu dilakukan kajian lanjutan mengenai sifat-sifat aljabar dari operator dan implementasi operasi komposisi yang dilanjutkan dengan uji coba secara empiris.

DAFTAR PUSTAKA

Adzkiya, Dieky, 2008, *Membangun Model Petri net Lampu Lalulintas dan Simulasinya,* Thesis, Jurusan Matematika FMIPA ITS.

Cassandras, C.G., S. Lafortune, 2008, *Introduction to Discrete Event Systems*, Second Ed., Springer Science and Business Media.

Chen, P.Z., C.H. Sun, S.Y. Yang, 2008, *Modeling and Analysis the Web Structure Using Stochastic Timed Petri Nets*, Journal Of Software, Vol. 3, No. 8, Academy Publisher.

Hamadi, R., Boualem Benatallah, 2003, A Petri Net-based Model for Web Service Composition, Fourteenth Australian Database Conference (ADC2003).

Viroli, M., Denti, E., Ricci, A., 2007. *Engineering a BPEL Orcheatration Engine as a Multi-Agent System*. Elsevier Science of Computer Programming. 66 (2007). 226-245.

Wang, J., 1998, Timed Petri Net : Theory and Application. Norwell, Kluwer Academic Publishers.

Yang, SY., 2007, *Using Petri Nets to Enhance Web Usage Mining,* Acta Polytechnica Hungarica, Vol. 4., No 3.