

jurnal informal

by Hilman Nuril Hadi

Submission date: 11-May-2022 08:57AM (UTC+0700)

Submission ID: 1833441736

File name: Jurnal_Informal_-_Hilman_published.pdf (468.7K)

Word count: 3153

Character count: 19085

Pengujian Unit Plugin Pemberian Anotasi Efek Secara Semantik pada Proses Bisnis BPMN

Hilman Nuril Hadi*

* Program Studi Teknik Informatika, STIKI Malang

*hilman@stiki.ac.id

ABSTRACT

Business process model was created to make it easier for business process stakeholders to communicate and discuss the structure of the process more effectively and efficiently. Business process models can also be business artifacts and media that can be analyzed further to improve and maintain organizational competitiveness. To analyze business processes in a structured manner, the effect/results of the execution of business processes will be one of the important information. The effect/result of the execution of certain activities or a business process as a whole are useful for managing business processes, including for improvements related to future business processes. This effect annotation approach needs to be supported by business process modeling tools to assist business analysts in managing business processes properly. In previous research, the author has developed a plugin that supports business analysts to describe the effects semantically attached to activities in the Business Process Model and Notation (BPMN) business process model. In this paper, the author describes the unit testing process and its results on the plugin of semantic effect annotation that have been developed. Unit testing was carried out using the basic path testing technique and has obtained three test paths. The results of unit test for plugin are also described in this paper.

Keyword: unit testing, business process, BPMN

1. Pendahuluan

Proses bisnis merupakan serangkaian aktivitas yang dilakukan dalam koordinasi di organisasi dan lingkungan teknis, dimana rangkaian aktivitas tersebut sekaligus mewujudkan tujuan tertentu [1]. Pemodelan proses bisnis menjadi bagian penting di dalam organisasi, dimana dengan pemodelan tersebut dapat mengelola serangkaian aktivitas pada proses bisnis dengan lebih optimal. Hal tersebut dibuktikan dengan sebuah penelitian yang melaporkan sebanyak 93% (520 dari 559) responden bisnis (berasal dari berbagai industri, ukuran organisasi, dan wilayah geografis) telah melakukan pemodelan proses bisnis [2]. Salah satu tujuan pemodelan proses bisnis adalah untuk menciptakan model proses bisnis, dimana model tersebut merupakan bagian yang fundamental dalam manajemen proses bisnis [3]. Selain itu, model tersebut juga dapat digunakan untuk menganalisis, memahami, dan memperbaiki proses yang dimodelkan.

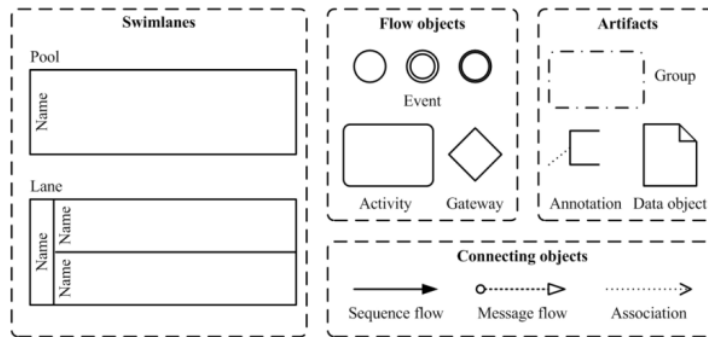
Model proses bisnis secara khusus diciptakan untuk memudahkan para *stakeholders* proses bisnis untuk berkomunikasi dan berdiskusi mengenai struktur dari proses tersebut dengan cara yang lebih efektif dan efisien [4]. Model proses bisnis juga dapat menjadi artefak bisnis dan sebagai sarana yang dapat dianalisis lebih lanjut dalam rangka meningkatkan dan mempertahankan daya saing organisasi. Secara teori, proses bisnis dapat dianalisis dalam tahapan desain. Proses analisis tersebut dapat dilakukan dengan menganalisis struktur atau perilaku model proses bisnisnya, tergantung pada elemen model proses bisnis dan hubungan antar elemen-elemennya [5]. Selain itu, model proses bisnis juga dapat dianalisis secara semantik, misalnya terkait dengan hasil eksekusi yang direpresentasikan dalam bentuk efek dari eksekusi satu atau lebih aktivitas dalam model proses bisnis [6]. Hasil eksekusi (efek) dari proses bisnis akan menjadi salah satu informasi pendukung untuk perbaikan terkait proses bisnisnya. Selain itu, efek dari eksekusi aktivitas tertentu ataupun suatu proses bisnis secara keseluruhan berguna untuk pencarian informasi pada proses bisnis berdasarkan dampaknya. Berdasarkan contoh kasus tersebut, analis bisnis mungkin ingin mengetahui/bertanya "apa efek dari proses jika proses tersebut dieksekusi sampai titik/aktivitas tertentu?" [7]. Akan tetapi, beberapa alat bantu pemodelan proses bisnis belum menyediakan fitur untuk mengelola informasi mengenai efek yang dihasilkan dari masing-masing aktivitas ataupun model proses bisnisnya. Padahal, analis bisnis harus didukung dengan alat untuk secara sistematis memperkaya model proses dengan efeknya. Sehingga efek dari eksekusi proses bisnisnya dapat diketahui dan dianalisis lebih awal sebelum proses bisnisnya diimplementasikan dalam kondisi nyata.

Dari permasalahan tersebut, telah dikembangkan alat yang mendukung analisis bisnis untuk memberikan efek anotasi secara semantik yang melekat pada aktivitas model proses bisnis *Business Process Model and Notation* (BPMN) [7]. Akan tetapi dalam penelitian tersebut tidak menjelaskan cara untuk merepresentasikan efek aktivitas dengan mudah dengan mempertimbangkan struktur kalimat yang sesuai dengan spesifikasi efek. Untuk mengatasi hal itu, peneliti telah mengembangkan plugin yang bisa digunakan dalam alat bantu pemodelan untuk menambahkan anotasi efek aktivitas proses bisnis pada publikasi sebelumnya [8]. Selanjutnya, peneliti akan menjelaskan proses eksekusi dan hasil pengujian unit plugin pemberian anotasi efek secara semantik pada aktivitas proses bisnis BPMN. Dalam penelitian ini, pengujian dilakukan dengan menggunakan teknik pengujian jalur dasar untuk melakukan proses penjaminan mutu terhadap plugin yang telah dikembangkan sebelumnya.

Pembahasan dalam artikel ini terbagi menjadi beberapa bagian. Bagian 2 menjelaskan anotasi efek secara semantik pada model proses bisnis BPMN. Bagian 3 menjelaskan metodologi penelitian dalam pengujian yang akan dilakukan. Bagian 4 menjelaskan langkah-langkah pengujian jalur dasar dan hasil pengujian. Terakhir, bagian 5 menjelaskan kesimpulan dan penelitian lanjutan.

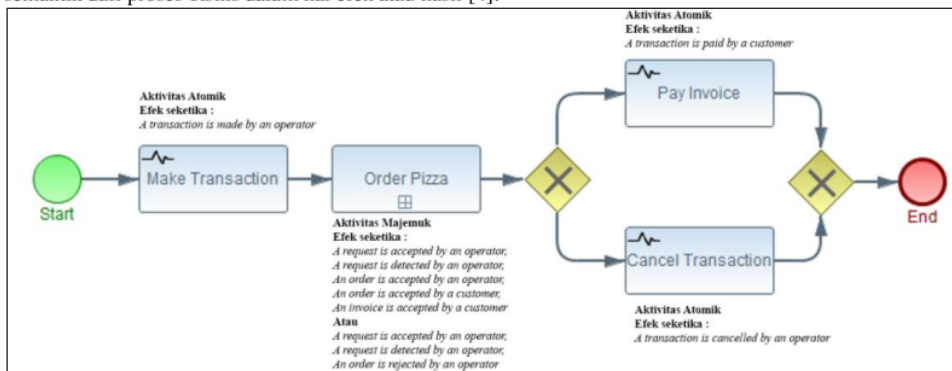
2. Anotasi efek secara semantik pada model proses bisnis BPMN

Selama beberapa tahun terakhir, manajemen proses bisnis telah berkembang di segmen ilmu komputer sebagai sarana untuk mengelola kompleksitas proses bisnis. BPMN menjadi standar untuk melakukan pemodelan proses bisnis yang menyediakan notasi grafis untuk mewakili semua proses bisnis yang ditemukan di dalam organisasi. Pada dasarnya, BPMN memiliki empat kategori elemen, diantaranya adalah flow objects, connecting objects, swimlanes dan artifacts [9], seperti yang ditunjukkan pada Gambar 1.



Gambar 1 Kategori dari Elemen BPMN, sumber: White (2004)

Meskipun popularitasnya lebih tinggi dari kerangka pemodelan proses bisnis lainnya, BPMN masih memiliki beberapa kekurangan seperti yang dilaporkan dalam penelitian terbaru [10]. Selain itu, BPMN dan kerangka pemodelan proses yang tersedia lainnya belum menyediakan fasilitas apapun untuk menggambarkan semantik dari proses bisnis dalam hal efek atau hasil [4].



Gambar 2 Model Proses Bisnis Pembelian Pizza

Pada dasarnya, aktivitas dapat dibagi menjadi dua jenis, yaitu aktivitas atomik dan aktivitas majemuk. Aktivitas atomik menggunakan elemen *task* dan aktivitas majemuk menggunakan elemen *sub-process*. Selanjutnya, aktivitas (atomik ataupun majemuk) diberikan anotasi efek yang merepresentasikan dampak/hasil dari eksekusi aktivitasnya. Sehubungan dengan suatu aktivitas dapat bersifat atomik ataupun majemuk, maka efek seketika dari aktivitasnya juga dapat berupa deterministik dan non-deterministik. Sehingga pada aktivitas majemuk dapat menghasilkan skenario efek lebih dari satu karena memungkinkan terdapat sejumlah jalur eksekusi, yang mana jalur eksekusi disebabkan dengan adanya pilihan jalur yang dapat dilalui (adanya notasi gerbang-OR ataupun gerbang-XOR). Sedangkan aktivitas atomik hanya terdiri dari 1 skenario efek. Oleh karena itu, efek seketika pada aktivitas dapat didefinisikan sebagai sekumpulan dari skenario efek $\{es_1, \dots, es_n\}$, dimana 1 dan n menandakan nomor jalur eksekusinya. Efek seketika pada suatu aktivitas disusun oleh sekumpulan klausa/kalimat (c_n) yang tergabung dengan media gerbang-AND (\wedge) atau gerbang-OR (\vee). Gerbang-AND digunakan untuk menggabungkan beberapa klausa yang didefinisikan. Sedangkan gerbang-OR digunakan untuk memisahkan klausa yang merepresentasikan efek seketika dari jalur eksekusi tertentu dalam satu himpunan efek seketika. Misalnya efek seketika dari sebuah aktivitas berupa $\{c_1 \wedge c_2 \vee c_3 \wedge c_4 \wedge c_5\}$, maka terdapat dua skenario efek yang ditandai dengan adanya satu buah gerbang-OR yang masing-masing skenario efek memiliki sekumpulan klausa yaitu $es_1 = \{c_1, c_2\}$ dan $es_2 = \{c_3, c_4, c_5\}$. Setiap klausa dibentuk sekurang-kurangnya terdiri dari subjek, predikat dan objek. Dalam rangka menyeragamkan struktur kalimat pada setiap klausa, maka penelitian ini mengusulkan struktur *passive voice* sebagai struktur yang digunakan untuk merepresentasikan satu klausa dalam efek seketika. Struktur tersebut dipilih karena struktur *passive voice* menunjukkan bahwa bagian objek pada kalimat menjadi bagian yang paling penting, dimana di kasus ini hasil/dampak berposisi sebagai objek pada suatu kalimat. Struktur *passive voice* pada efek seketika sebagai berikut: *Subject + auxiliary verb + past participle +/-by (doer of action)*, Contohnya 'A message is accepted by an operator'. Untuk memudahkan dalam memahami aktivitas beserta efek seketikanya, Gambar 2 memberikan contoh beberapa elemen *task* dan elemen *sub-process* terkait model bisnis pembelian pizza. Pada Gambar 2, terdapat tiga buah aktivitas atomik dan satu buah aktivitas majemuk. Penjelasan detail terkait efek seketika masing-masing aktivitas dijelaskan dalam Tabel 1.

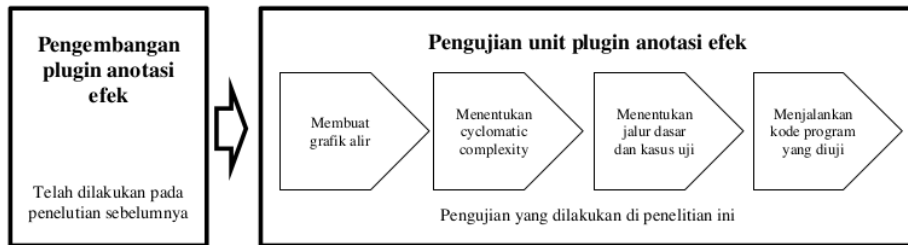
Tabel 1 Representasi Efek Seketika pada Aktivitas Proses Bisnis Pembelian Pizza

No	Nama Aktivitas	Efek Seketika
1	<i>Make Transaction</i> (Aktivitas Atomik)	Efek seketika = c_1 Deklarasi klausa efek: <i>A transaction is made by an operator</i> $\rightarrow c_1$
2	<i>Order Pizza</i> (Aktivitas Majemuk)	Efek seketika = $c_{21} \wedge c_{22} \wedge c_{23} \wedge c_{24} \wedge c_{25} \vee c_{21} \wedge c_{22} \wedge c_{26}$ Deklarasi klausa efek: <i>A request is accepted by an operator</i> $\rightarrow c_{21}$ <i>A request is detected by an operator</i> $\rightarrow c_{22}$ <i>An order is accepted by an operator</i> $\rightarrow c_{23}$ <i>An order is accepted by a customer</i> $\rightarrow c_{24}$ <i>An invoice is accepted by a customer</i> $\rightarrow c_{25}$ <i>An order is rejected by an operator</i> $\rightarrow c_{26}$
3	<i>Pay Invoice</i> (Aktivitas Atomik)	Efek seketika = c_3 Deklarasi klausa efek: <i>A transaction is paid by a customer</i> $\rightarrow c_3$
4	<i>Cancel Transaction</i> (Aktivitas Atomik)	Efek seketika = c_4 Deklarasi klausa efek: <i>A transaction is cancelled by an operator</i> $\rightarrow c_4$

3. Metodologi penelitian

Bagian ini berisi informasi mengenai teknik dan langkah-langkah yang digunakan untuk melakukan pengujian jalur dasar pada plugin yang telah dikembangkan pada penelitian sebelumnya. Pada Gambar 3 merepresentasikan dua bagian inti yang menggambarkan tahapan-tahapan pengembangan plugin pemberian anotasi efek secara semantic pada model proses bisnis BPMN. Bagian pertama adalah pengembangan plugin anotasi efek yang telah dilakukan pada penelitian sebelumnya. Hasil tahapan ini, plugin yang nantinya bisa dijalankan dalam alat bantu pemodelan Eclipse BPMN2 Modeler. Selanjutnya, tahapan pengujian dilakukan

untuk mengetahui hasil pengembangan *plugin* anotasi efek secara semantik pada alat pemodelan Eclipse BPMN2 Modeler. Pada penelitian ini, proses pengujian *plugin* menggunakan teknik jalur dasar / *basis path* untuk membangkitkan jalur uji dan data uji. Setelah itu, klas hasil implementasi dieksekusi sesuai dengan data uji untuk menentukan klas tersebut telah sesuai dengan spesifikasi atau tidak. Hasil dari pengujian ini diharapkan dapat memastikan dan membuktikan bahwa setiap pernyataan (*statement*) dari kode program (khususnya pada bagian representasi efek seketika pada *plugin*) dieksekusi minimal sekali dan tidak menimbulkan pesan kesalahan yang tidak memiliki makna.



Gambar 3 Diagram alur pengujian plugin anotasi efek

4. Pengujian

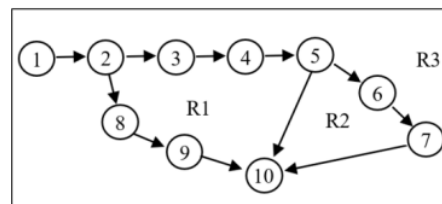
Salah satu fokus pengujian perangkat lunak adalah menjalankan program dengan input tertentu dan memeriksa apakah program berperilaku benar dan tepat sesuai dengan spesifikasi [11]. Untuk pengujian jalur dasar, terdapat beberapa Langkah-langkah yang perlu dikerjakan meliputi: membuat grafik alir, menentukan cyclomatic complexity, menentukan jalur dasar dan kasus uji, dan menjalankan cuplikan kode programnya. Pada jurnal ini, peneliti membatasi dokumentasi pengujian unit hanya pada satu fungsi inti dalam pengembangan *plugin* anotasi efek yakni fungsi pengecekan efek dengan menggunakan layanan eksternal. Pembahasan lebih detail mengenai masing-masing tahapan dijelaskan pada subbab berikutnya.

4.1 Pembuatan Grafik Alir

Grafik alir adalah sebuah notasi sederhana yang merepresentasikan aliran kontrol dari sebuah struktur program. Untuk membuat grafik alir, pada penelitian ini mengacu pada pseudocode pengecekan kalimat efek. Proses pengecekan dilakukan dengan menggunakan bantuan layanan eksternal yakni layanan CNL ACE *Attempto Parsing Engine* (APE) untuk merepresentasikan efek seketiknya. Gambar 4a merupakan representasi pseudocode pengecekan efek yang menjelaskan pernyataan struktur logika dari pengecekan efeknya. Selanjutnya, pada Gambar 4b merepresentasikan hasil grafik alir sesuai dengan pseudocodenya.

```

Nama Algoritme: Pengecekan Kalimat (Check_CNL)
Deklarasi :
  A. String → Effect
  B. String → result
Deskripsi :
  Read(Effect) ①
  If internet_connection = true then ②
    result = service(effect) ③
    if result contain error then ④
      result = getNotificationError ⑤
    else ⑥
      result = "the effect is correct" ⑦
  Else ⑧
    Result = "Please, connect to internet" ⑨
  Return Result ⑩
    
```



(a)

(b)

Gambar 4 Pseudocode operasi pengecekan kalimat (a) dan grafik alir pengecekan kalimat (b)

4.2 Cyclomatic Complexity

Cyclomatic complexity atau biasanya disimbolkan dengan $V(G)$ merupakan nilai yang menyatakan ukuran tingkat kompleksitas dari suatu program. Berdasarkan hasil grafik alir pada gambar 4b, Jumlah *Cyclomatic complexity* $V(G)$ adalah 3. Nilai tersebut didapat melalui beberapa cara. Cara pertama ditentukan dengan berdasarkan jumlah area pada grafik alir pada Gambar 4b yaitu 3 area (R1, R2, dan R3). Beberapa area tersebut dapat diperoleh dengan menentukan wilayah/bagian tertutup maupun wilayah terluar yang dihasilkan dari grafik alirnya, dalam kasus diatas yakni R1, R2, dan R3. Cara kedua ditentukan dengan rumus $V(G) = E - N + 2 = 11 - 10 + 2 = 3$, dimana E =jumlah garis=11 dan N = jumlah titik=10. Selanjutnya, cara ketiga dapat ditentukan dengan rumus $V(G) = P + 1 = 2 + 1 = 3$, dimana P =*predicate node / node* yang memiliki garis output lebih dari satu yakni (*node* 2 dan 5).

4.3 Jalur Dasar dan Kasus Uji

Jalur dasar merupakan sebuah jalur pada program yang mengandung paling sedikit sebuah pernyataan logik. Berdasarkan nilai $V(G)$, maka jumlah jalur dasar yang harus diidentifikasi adalah sebanyak 3 jalur. Setelah menentukan jalur dasar, langkah selanjutnya adalah mendefinisikan kasus-kasus uji berdasarkan jalur dasar yang telah ditentukan. Detil mengenai jalur dasar dan kasus uji masing-masing dijelaskan dibawah ini:

1. Jalur dasar 1 : {1 → 2 → 3 → 4 → 5 → 6 → 7 → 10 }
 Data uji: kalimat efek (*effect*) = *a transaction is made by an operator*
 Hasil yang diharapkan: *“the effect is correct”* atau memberikan informasi bahwa kalimat efek telah sesuai dengan ketentuan.
 Catatan: kalimat efek telah sesuai dengan struktur *passive voice* pada efek seketika → *subject + auxiliary verb + past participle +/-by (doer of action)*
2. Jalur dasar 2 : {1 → 2 → 3 → 4 → 5 → 10}
 Data uji: kalimat efek (*effect*) = *an order is accept by an operator*
 Hasil yang diharapkan: *“error notification”* atau memberikan informasi bahwa kalimat efek tidak sesuai dengan ketentuan dan memberitahukan letak kesalahannya.
 Catatan: memberikan pesan eror misalnya: *an order is accept <> by an operator*. Tanda <> menyatakan letak terjadinya kesalahan ejaan kata.
3. Jalur dasar 3 : {1 → 2 → 8 → 9 → 10}
 Data uji: kalimat efek (*effect*) = *a transaction is made by an operator*
 Hasil yang diharapkan: *“Please, connect to internet”* atau memberikan informasi bahwa komputer harus ada koneksi internet
 Catatan: saat pengujian, komputer harus dalam kondisi tidak terhubung dengan internet.

4.4 Hasil eksekusi kode program

Bagian ini berisi hasil pengujian unit yang dilakukan berdasarkan tiga buah jalur kasus uji yang telah dijelaskan pada bagian sebelumnya. Pengujian unit tersebut dilakukan dengan membuat *class* baru yang berisi fungsi pengecekan efek dan menambahkan fungsi main program agar dapat dieksekusi, cuplikan kode program fungsi pengecekan efek dijabarkan pada Tabel 2.

Tabel 2 kode program fungsi pengecekan efek

LOC	Nama Fungsi : <i>Check_Effect</i>
1	<code>public String Check_effect(String imm_effect){</code>
2	<code>String result = imm_effect;</code>
3	<code>imm_effect = imm_effect.replace(' ','+');</code>
4	<code>String Str_CheckCNL = "";</code>
5	<code>if (Check_Intemet_Connection()==true) {</code>
6	<code>URL url;</code>
7	<code>String str_url =</code>
8	<code>"http://atempto.ifi.uzh.ch/ws/ape/apews.perl?text="+imm_effect+"&solo=fol";</code>
9	<code>url = new URL(str_url);</code>
10	<code>URLConnection con = url.openConnection();</code>
11	<code>InputStream in = con.getInputStream();</code>
12	<code>String encoding = con.getContentEncoding();</code>

```

13 Scanner s = new Scanner(in).useDelimiter("\\A");
14 result = s.hasNext() ? s.next() : "";
15 Boolean found = false;
16 found = result.contains("error");
17 if (found) {
18     DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
19     DocumentBuilder db = dbf.newDocumentBuilder();
20     Document doc = db.parse(url.openStream());
21     NodeList nodes = doc.getElementsByTagName("message");
22     for (int temp = 0; temp < nodes.getLength(); temp++) {
23         Node nNode = nodes.item(temp);
24         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
25             Element eElement = (Element) nNode;
26             Str_CheckCNL = "There's an error on the effect " + eElement.getAttribute("value")+"";
27             Str_CheckCNL = Str_CheckCNL + "\n" + "The sign '<>' indicates the point
28 of failure, make sure the word after / before the mark is correct.";
29         }
30     } else{
31         imm_effect = imm_effect.replace('+',' ');
32         Str_CheckCNL = "The effect "+imm_effect+" are correct.";
33         result = Str_CheckCNL;
34     } else{
35         result = "Please, make sure you're connected to internet.";
36     }
37     return result;
38 }
    
```

Selanjutnya, pengujian dilakukan dengan cara memanggil fungsi pengecekan efek dengan argumen data uji yang telah ditentukan. Hasil pengujian unit yang telah dilakukan dijabarkan pada Tabel 3.

Tabel 3 Hasil Pengujian Unit pada Fungsi Check_CNL

Rute jalur dan Data uji	Hasil Eksekusi Kode Program	Status Pengujian
Rute Jalur Dasar 1 : {1 → 2 → 3 → 4 → 5 → 6 → 7 → 10 } Data Uji: <i>Efek seketika: a transaction is made by an operator</i>	The effect 'transaction is made by an operator' is correct.	Telah sesuai dengan hasil yang diharapkan.
Rute Jalur Dasar 2 : {1 → 2 → 3 → 4 → 5 → 10} Data Uji: <i>Efek seketika: an order is <u>accept</u> by an operator</i>	<i>There's an error.</i> <i>Statement: 'an order is accept <> by an operator.'</i> <i>The sign '<>' indicates the point of failure, make sure the word after/before the mark is correct</i>	Telah sesuai dengan hasil yang diharapkan.
Rute Jalur 3 : {1 → 2 → 8 → 9 → 10} Data Uji: <i>Effect: a transaction is made by an operator</i> Catatan: saat pengujian, komputer harus dalam	<i>Please, connect to internet</i>	Telah sesuai dengan hasil yang diharapkan.

kondisi tidak terhubung dengan internet.		
--	--	--

5. Kesimpulan

Berdasarkan penelitian yang telah dilaksanakan, maka dapat diambil kesimpulan antara lain: pengujian unit menggunakan *basis path* operasi dapat mengacu pada *pseudocode* yang telah dimodelkan pada tahapan perancangan, dalam kasus ini menggunakan operasi/fungsi pengecekan efek seketika pada aktivitas proses bisnis BPMN. Selanjutnya, pengujian dengan menggunakan metode jalur dasar diperoleh 3 buah jalur uji. Jalur uji tersebut digunakan sebagai acuan untuk melakukan eksekusi kode program agar proses pengujian program bisa dilakukan secara efektif (pengujian dapat langsung difokuskan untuk mengecek jalur yang spesifik misalnya jika dalam keadaan tanpa ada koneksi internet) dan efisien (kode program setidaknya dieksekusi minimal 3 kali dengan data uji berbeda). Jalur uji yang diperoleh disimpulkan telah mewakili struktur logika dari rancangan pseudocode yaitu kondisi jika efek seketika telah sesuai spesifikasi, kondisi jika efek seketika tidak sesuai spesifikasi, dan kondisi jika user/komputer tidak memiliki akses internet. Selanjutnya, dari proses eksekusi program (3 jalur uji dan data uji) telah dilakukan menunjukkan bahwa program telah sesuai dengan hasil yang diharapkan.

Daftar Referensi

- [1] M. Weske, *Business Process Management: Concepts, Languages, Architectures*, vol. 54, no. 2, 2010.
- [2] P. Harmon and C. Wolf, "Business Process Modeling Survey," *Bus. Process trends*, no. December, p. 36, 2011, [Online]. Available: www.bptrends.com.
- [3] W. M. P. Van Der Aalst, "Business Process Management: A Comprehensive Survey," *ISRN Softw. Eng.*, vol. 2013, pp. 1–37, 2013, doi: 10.1155/2013/507984.
- [4] T. A. Kurniawan, "Process ecosystem views to managing changes in business process repositories," University of Wollongong, 2013.
- [5] R. Dijkman, M. Dumas, B. Van Dongen, K. Reina, and J. Mendling, "Similarity of business process models: Metrics and evaluation," *Inf. Syst.*, vol. 36, pp. 498–516, 2011, doi: 10.1016/j.is.2010.09.006.
- [6] G. Koliadis and A. Ghose, "Verifying Semantic Business Process Models in Inter-operation," *IEEE Int. Conf. Serv. Comput.*, pp. 731–738, 2007.
- [7] K. Hinge, A. Ghose, and G. Koliadis, "Process SEER: A tool for semantic effect annotation of business process models," *Proc. - 13th IEEE Int. Enterp. Distrib. Object Comput. Conf. EDOC 2009*, pp. 54–63, 2009, doi: 10.1109/EDOC.2009.24.
- [8] H. N. Hadi, T. A. Kurniawan, and I. Aknuranda, "Plug-in for Annotating Semantic Effect on BPMN Business Process Models," *J. Inf. Technol. Comput. Sci.*, vol. 4, no. 2, pp. 116–126, 2019, doi: 10.25126/jitecs.20194255.
- [9] S. A. White, "Introduction to BPMN," *BPTrends*, no. IBM Corporation (c), pp. 1–11, 2004, doi: 10.3727/0000006783982421.
- [10] J. C. Recker, "Opportunities and constraints : the current struggle with BPMN," *Bus. Process Manag. J.*, vol. 16, pp. 181–201, 2010, [Online]. Available: <https://eprints.qut.edu.au/20316/1/c20316.pdf>.
- [11] A. Mili and F. Tchier, *Software Testing Concepts and Operations*. John Wiley & Sons, Inc., 2015.

jurnal informal

ORIGINALITY REPORT

14%

SIMILARITY INDEX

14%

INTERNET SOURCES

7%

PUBLICATIONS

6%

STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

2%

★ Lecture Notes in Business Information Processing,
2012.

Publication

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off