Abstract: The business process model is a representation of business activities illustrated through diagram notations. This model is composed of repeated specific patterns called basic control structure. Each basic control structure has a level of complexity. The metrics for formulating existing complexity are very diverse, but can only define complexity partially and are less sensitive to small changes in the structure of the business process model. In this paper, we propose a formula of complexity metric that can indicate small changes in structure, type of branching logic, number of branches, loops, and depth. We call it the Yaqin complexity formula. To get the Yaqin complexity formula, we carried out several activities. These activities are identifying the metrics involved using the Goal-Question-Metric (GQM) method, then formulating the metric complexity, the next activity is testing the formula with several business process models and analyzing the test results, and then validating the Yaqin complexity formula using the Weyuker's properties framework. The Yaqin complexity formula, which involves seven parameters, is proven to be more comprehensive than other complexity formulas that involve less than seven parameters. The Yaqin complexity formula also proved to be more sensitive to other complexity formulas, where 7 out of 8 cases affected the Yaqin complexity metric. The validation results state that the Yaqin complexity formula meets 8 of 9 Weyuker's properties. Thus we have succeeded in formulating the Yaqin complexity, which is more comprehensive in involving parameters and more sensitive in detecting small changes in the structure of the business process model.

Keywords: Complexity metrics formula, Business process model, Workflow pattern.

1 Introduction

Business processes are defined as a set of structured and interrelated activities that produce specific products or services [1]. Some business activities are composed in such a way as a business process model. The business process model is a representation of functions related to business activities such as input, control, output, and resources. According to the Business Process Model Notation (BPMN), the business process model consists of several components; event, activity, and control flow [2]. The components are simply composed to form a basic pattern. These patterns are called Workflow Patterns (WP). The WP were proposed by van der Aalst, ter Hofstede, Kiepuszewski, and Barros [3]. They defined eight groups of workflow patterns and detailed them into 43 workflow patterns. Nick Russel, ter Hofstede, van der Aalst, and Nataliya Mulyar completed it by providing a systematic description and formal description in Colored Petri Net (CPN) notation [4].

The business process model is composed of several scalable Basic Control Structure (BCS) [5]. BCS is part of the WP. When BCS are composed to form a business process model, it has complexity. The complexity of the business process model describes the state of the problem that requires resolution [6]. Complexity can be measured in various ways and metrics. The proposed metrics also have many kinds. Jan Mendling, in 2008 in his book, explained about 28 metrics used to measure the size of the business process model [7]. He used the Event-driven Process Chain (EPC) notation. Muketha, Ghani, Selamat, and Atan in 2010 conducted a survey...
of business process complexity metrics and how to validate them. Twenty-five metrics for business process complexity were identified [8]. Polancic and Cegnar, in 2017, surveyed the complexity metrics of the business process model. The survey produced 66 metrics of complexity that had been proposed [9]. In 2012 Carlo Corti created software to measure business process model metrics with BPMN notation [10].

Measurement of the business process model complexity has been carried out by several researchers, namely Cardoso in 2008, proposed the Control Flow Complexity metric (CFC) [11], which is an adaptation of Cyclomatic Complexity proposed by McCabe in 1976 [12]. Gruhn and Laue in 2006, proposed Cognitive Weight (CW) [13], which is a modification of the Cognitive Functional Size (CFS) proposed by Shao and Wang in 2003 [5]. Vanderfeesten, Cardoso, and Mendling in 2007 proposed Cross-Connectivity (CC) based on Cognitive Complexity as a metric to measure complexity [14]. The most comprehensive and sensitive complexity metric than the one proposed above is the Cognitive Activity Depth Arc Control Flow (CADAC) proposed by Emriye Coskun in 2014 [15]. Yaqin also proposed a metric measurement of complexity called Scale in 2017 involving CFC parameters and the number of components of the business process model [16]. The complexity metrics they propose have weaknesses; they can only calculate partial complexity metrics and are less sensitive to small changes that occur in the measured business process model. The papers that we use as references in this study are more than three years old because, in the field of measuring the business process model complexity metric, we do not find new complexity metric findings, be it metric atomic or composite metric. The complexity metric that we make the primary comparison is CADAC, the most comprehensive and sensitive formula among the complexity metrics ever proposed.

In this paper, we examine and discuss how to measure the complexity metric of business process models based on their BCS more comprehensively and are more sensitive to small changes in the structure of business process models. The purpose of this research is to determine the formula for the complexity of business process models that are more comprehensive and sensitive based on the BCS that forms it.

The proposed contribution is the business process model complexity metric that is more comprehensive and sensitive to small changes that occur in the business process model. The more comprehensive complexity metrics are defined as a metric that includes more measurement parameters to get the metric value of the business process model complexity. The more sensitive complexity metrics are defined as metrics that can affect the complexity of the business process model for structural changes in the business process model. We have defined the structural change into two parts. They are significant change and small change. Significant changes are changes that involve adding components and changes of logic to the business process model. In contrast, small changes are changes that occur without the addition of components and changes of logic to the structure of the business process model. The more comprehensive and sensitive complexity metric is essential to improve accuracy in estimating application development costs, increasing accuracy in application configuration that serves the business process.

This research is part of a broader scope of the development of enterprise architecture growth metrics research. One of the stages in the research is the development of the business process model complexity metric. The formula for complexity metric is useful as one of the components needed to simulate the growth of an enterprise architecture that is ultimately used to determine the configuration of information systems based on the business architecture complexity. The business process model complexity metric is also useful for getting the number of resources needed to run a business process [17].

This paper is organized as follows, background, problem formulation, related research, and contributions are presented in section 1. Section 2 explains the theories related to this research, namely about BCS, BCS metrics, and the complexity of business process models. Section 3 describes the methods and stages used to propose the complexity metric for business processes models. Then in section 4, we explain the experimental results and their discussion. Section 5 describes the conclusions of this study.

2 Theory

2.1 Basic control structure

WP are abstractions of real forms of workflow that are repeated in specific non-arbitrary contexts. The WP was proposed by van der Aalst, ter Hofstede, Kiepuszewski, and Barros, in 2003 [3]. The WP uses a Petri-net notation. WP are defined in 8 groups of patterns and broken down into 43 patterns. Then updated by Russell, ter Hofstede, van Der Aalst, Mulyar in 2006 [4]. The update is an improvement by
providing a systematic and formal description in the form of colored Petri-net. In 2004, Stephen A. White specified a WP in BPMN notation [18].

On the other hand, Shao and Wang in 2003 identified BCS as part of the WP based on cognitive patterns [5]. Each cognitive pattern has a weight. This weight is a measure of the complexity of the pattern. In 2006, Gruhn and Laue added cognitive patterns, namely multiple-choice, multiple instances, cancel activity, and cancel case [13]. In this paper, the BCS is defined as in Definition 1.

Definition 1 Basic Control Structure
Business process model (BPmodel) consists of Basic Control Structure (BCS), where

\[ BCS = \{ seq, AND, OR, XOR, cyc \in WP \} \]

BCS is defined as seq iff \( t_0 \xrightarrow{next} t_k \), where \( t \in Ac \)

BCS is defined as AND iff \( t_0 \) has branch \( t_{01}, ..., t_{0s} \), where \( t \in Ac \)

BCS is defined as OR iff \( t_0 \) has branch \( t_{01}, ..., t_{0s} \), where \( t \in Ac \)

BCS is defined as XOR iff \( t_0 \) has branch \( t_{01}, ..., t_{0s} \), where \( t \in Ac \)

BCS is defined as cyc iff \( t_k \) has loop branch, where \( t \in Ac \)

2.2 Metrics for workflow patterns
The business process model metric consists of atomic and composite metrics. Atomic metrics are stand-alone metrics and are not formed by other metrics in a business process model, which in this paper is defined as in Definition 2. While composite metrics are metrics formed by other metrics in a business process model in this paper is defined as in Definition 3.

Definition 2 Atomic metrics
The atomic metrics used in this paper are the arcs size (As), the starts size (Ss), the ends size (Es), the intermediates size (Is), the activities size (Acs), the depth (D), the branches size (Bs), and diameters (Dm), that are part of BPmodel. The atomic metric is defined as:

\[ M_{Atomic} = \{ As, Ss, Es, Is, Acs, D, Bs, Dm \in BPmodel \} \]

Definition 3 Composite metrics
The composite metrics used in this paper are the number of nodes (Ns), number of branching types (Bt), and the complexity of the loop (C_{cyc}) that are part of the BPmodel. Composite metrics are defined as:

\[ M_{Composite} = \{ Ns, Bt, C_{cyc} \in BPmodel \} \]

where

\[ Ns = 5s + Es + Is + Acs + Bt \]  \hspace{1cm} (1)

\[ Bt = ANDs + XORs + ORs \]  \hspace{1cm} (2)

where

\( ANDs : number \ of \ AND \ branching \)

\( XORs : number \ of \ XOR \ branching \)

\( ORs : number \ of \ OR \ branching \)

2.3 The complexity of business process models
Every BCS has a complexity value. Edmond defines complexity as the nature of language expression, which makes it difficult to formulate its overall behavior, even when given almost complete information about the atomic components and the relationships between them [19]. The complexity of the business process model is measured through metrics in diagrammatic representation or formal language. Measurement of business process models complexity can be approached based on software engineering theory, cognitive science, and graph theory. According to Fenton and James Bieman, the definition and validation of new metrics must go through three stages, namely the definition of metrics, theoretical validation of metrics, and empirical validation of metrics. The procedure for defining metrics goes through three stages, namely the identification of the measured entity, the identification of the attributes of the measured entity, and the definition of the metric [20]. The complexity of the business process model is measured by several aspects, namely activities, control flow, data flow, and resources [21].

The cognitive weight (CW) of software is the level of difficulty required to understand software that is modeled with several BCS [5]. The CW value for BCS used in this paper is presented in Table 1.

3 Methods
3.1 Identification of business process model complexity metrics

Business process model metrics are selected and identified using the Goal-Question-Metric (GQM) method. GQM is a framework that uses a top-down approach [22]. The stage of GQM consists of: (1) Specific goals stated; (2) Asking questions whose answers help achieve the goal; (3) Provide a measurement scheme.

The Goal measurement of the business process model complexity is defined as presented in Table 2.
Table 1. Cognitive weight on basic control flow

<table>
<thead>
<tr>
<th>Workflow Pattern</th>
<th>Weight</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>1</td>
<td>[13]</td>
</tr>
<tr>
<td>Exclusive Choice (XOR)</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Parallel Split (AND)</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Multiple Choice (OR)</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Looping (Cyclic)</td>
<td>3</td>
<td>[5]</td>
</tr>
<tr>
<td>Depth</td>
<td>14</td>
<td>[15]</td>
</tr>
</tbody>
</table>

Table 2. Goal measurement

<table>
<thead>
<tr>
<th>Analysis</th>
<th></th>
<th>Issue</th>
<th>To obtain quantitative value for and complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus of measurement</td>
<td></td>
<td>Focus of</td>
<td>exist in the business process model</td>
</tr>
<tr>
<td>Point of view</td>
<td></td>
<td>Context of</td>
<td>A business process model that is built from its workflow pattern</td>
</tr>
</tbody>
</table>

Table 3. Identification of the scale metrics and complexity of the business process model

Q.1 What parameters are used to measure the business process model complexity?
M.1.1 The business process model is represented by graphs using BPMN
M.1.2 The number of nodes
M.1.3 The number of arcs
M.1.4 The number of branches and types
M.1.5 The number of branches in each branch
M.1.6 The number of loops
M.1.7 The average depth of each node

3.2 Proposed formula

The Yaqin complexity formula is more comprehensive and more sensitive to small changes in the structure of the business process model. Our proposed formula accommodates aspects of activities (Acs), branch type (Bt), cyclic (Cyc), depth (D), and cognitive weight (CW). Ns is the sum of the event (Ss, Es, Is), activity (Acs), and branch type (Bt). The branch type aspect is represented by the modified control flow complexity, which consists of branch type AND, XOR, and OR. This modification was inspired by the determination of split-join logic in the process mining [23-26] as defined in Definition 4, Definition 5, and Definition 6.

Definition 4 AND branch complexity
Let the business process model in Fig. 2 (b). Possible paths are ABDEC; ABEDC; ADEBC; ABDC; AEDBC; AEBDC; AEDBC. Three branches produce six possible paths, so 6 = 3!. Weights are given based on Table 1 $CW_{AND} = 4$, so

$$C_{AND} = 4 \cdot \sum_i(n!)_i$$

(3)

where $n$: number of branches in each branch and $i$: index of AND branches in the business process model.

Definition 5 XOR branch complexity
Let the business process model in Fig. 7 (a). Possible paths are ABC, ADC, AEC. Three branches produce three possible paths. Weights are given in Table 1 $CW_{XOR} = 3$, so

$$C_{XOR} = 3 \cdot \sum_i(n!)_i$$

(4)

Definition 6 OR branch complexity
Let the business process model in Fig. 7 (b). Possible paths are ABC; ADC; AEC; ABDC; ADEBC; ABEDC; ABDE; AEBDC; AEDBC. From 3 branches produces 12 possible paths, so 12 = $3!$. Weights are given based on Table 1 $CW_{OR} = 7$ so

$$C_{OR} = 7 \cdot \sum_i \left( \sum_{k=1}^{n} \frac{n!}{(n-k)!} \right)_i$$

(5)

where $n$: number of branches in each branch, $k$: the number of branches that may be passed at a branch, and $i$: index of OR branches in the business process model.

The complexity of cyclic is also taken into account in the formulation of the complexity of business process models. The cyclic complexity is defined as in Definition 7.

Definition 7 Cyclic complexity
The business process model is said to be cyclically complete if the number of activities involved in a loop (Acs$_{cyc}$) equals as the diameter (Dm), so cyclical complexity is defined as the ratio of the number of activities involved in cyclic to its diameter. The intended diameter, as defined by Jan Mendling [7]. Weight given is based on Table 1 $CW_{cyc} = 3$ so

$$C_{cyc} = 3 \cdot \frac{Acs_{cyc}}{Dm}$$

(6)

The depth complexity is defined as the average depth of each activity in the business process model. The complexity of depth is defined in Definition 8.

Definition 8 Depth complexity
The complexity of depth is measured through the average depth of each activity formulated in Eq. (7).

$$D_{avg} = \frac{\sum_{j=Acs}^{} \frac{D_j}{Acs}}{Acs}$$

(7)
where

\( j \): number of activities

\( D \): depth of each activities

Weight given is based on Table 1 \( CW_D = 14 \) so

\[
C_D = 14 \cdot D_{avg} \tag{8}
\]

Finally, the Yaqin complexity metric of the business process model is defined as the sum of the number of vertices, the number of arcs, the sum of the AND, XOR, OR gateway complexities, the sum of cyclic complexities, and the depth complexity as defined in Definition 9.

Definition 9 Yaqin complexity metric

The Yaqin complexity metric of the business process model (YC) is defined based on Eq. (1), Eq. (2), Eq. (3), Eq. (4), Eq. (5), Eq. (6), and Eq. (8), so

\[
YC = Ns + As + C_{AND} + C_{XOR} + C_{OR} + C_{cyc} + C_D \tag{9}
\]

3.3 Experiment design

To test the Yaqin complexity formula, we use a dataset consisting of several business process models.

**Figure 1** Sequence business process models

(a) ![Sequence business process models](image1)

(b) ![Sequence business process models](image2)

(c) ![Sequence business process models](image3)

**Figure 2** Business process models with AND branches

(a) ![Business process models with AND branches](image4)

(b) ![Business process models with AND branches](image5)

**Figure 3** Business process models with the addition of activities in sequence

(a) ![Business process models with the addition of activities in sequence](image6)

(b) ![Business process models with the addition of activities in sequence](image7)

**Figure 4** Business process models with variations in the number of branches

(a) ![Business process models with variations in the number of branches](image8)

(b) ![Business process models with variations in the number of branches](image9)

(c) ![Business process models with variations in the number of branches](image10)

**Figure 5** Business process models with variations of branching position

(a) ![Business process models with variations of branching position](image11)

(b) ![Business process models with variations of branching position](image12)

(c) ![Business process models with variations of branching position](image13)
The business process models presented are required to test the complexity metric measurements, as shown in Table 4.

Then the business process models complexity are calculated with Eq. (9) to be analyzed. the complexity of the business process model is also calculated by formula McCabe’s cyclomatic complexity [12], Jorge Cardoso’s control flow complexity [11], The cognitive complexity formula proposed by Shao and Wang [5] that uses the definitions shown in Table 1, the Coskun’s CADAC complexity formula [15], and the Scale formula [16]. The McCabe’s cyclomatic complexity is:

\[
\text{Cyclomatic Complexity} = N_s + A_s
\]  

(a) Figure 6 Business process models with variations of branching depth

(b) Figure 7 Business process models with variations in branching types

(a) Figure 8 Business process models with looping variations

(a) Figure 9 Business process models with a combination of branching logic
The Jorge Cardoso’s control flow complexity is

\[ CFC = \sum CFC_{XOR} + \sum CFC_{OR} + \sum CFC_{AND} \]  

(11)

where

\[ CFC_{XOR} = fanout \]
\[ CFC_{OR} = 2fanout - 1 \]
\[ CFC_{AND} = 1 \]

The cognitive complexity formula is

\[ CW = \left\{ \sum_j \sum_i CW(j, i) \right\} \]  

(12)

The CADAC complexity formula is

\[ CADAC = (Acs \cdot 1) + (D_{max} \cdot 14) + (XORS \cdot 2) + (ANDs \cdot 4) + (ORS \cdot 7) + (Max(fanin \cdot fanout)^2 \cdot 4) + (As \cdot 1) \]  

(13)

The Scale is

\[ Scale = (Ns + As) \cdot CFC \]  

(14)

The results of calculations with various formulas were analyzed by comparing them. Comparison results will prove that the Yaqin complexity formula is more comprehensive and more sensitive to small changes in the structure of the business process model.

### 3.4 Theoretical validation of business process models complexity metrics

After the complexity metric is defined, the next step is to validate the complexity metric. Theoretical validation applied to the complexity metric is Weyuker’s properties [27]. Then the results of the validation are compared with the results of the validation of the Cyclomatic Complexity, CFC, CW, CADAC, and Scale formula.

### 4 Result and discussion

#### 4.1 Comprehensiveness of the yaqin complexity formula

The comprehensiveness of complexity formula is defined as the number of parameters involved in the formulation: the more parameters involved, the more comprehensive the formula. The Yaqin complexity formula includes more parameters than other complexity formulas. The Yaqin complexity formula consists of seven parameters namely \( Ns, As, ANDs, ORs, XORs, cyc, \) and \( D \). While CADAC consists of six parameters, Cyclomatic Complexity consists of two parameters, CFC consists of three parameters, CW consists of four parameters, and the Scale consists of five parameters. The comparison shows that the Yaqin complexity formula is more comprehensive than other formulas. The summary of the comprehensiveness is shown in Table 5.

#### 4.2 Experiment results

Metrics consist of two groups, namely atomic metrics and composite metrics. Atomic metrics are stand-alone metrics and are not composed of other metrics. Composite metrics are metrics composed of
several other metrics that form a formula.

This section presents the results of the business process model complexity metric calculations, which are the case studies shown in Fig. 1 to Fig. 9. As example, we show the calculation of the Yaqin complexity using Fig. 6 (b).

step 1: calculate $Ns$ using Eq. (1).

Let

\[
S_s = 1 \quad Acs = 8 \\
E_s = 1 \quad Bt = 4 \\
I_s = 0
\]

Table 5. Parameters of the complexity formula

<table>
<thead>
<tr>
<th>Complexity formula</th>
<th>$Cs$</th>
<th>$As$</th>
<th>$ANDs$</th>
<th>$ORs$</th>
<th>$XORs$</th>
<th>cyc</th>
<th>$D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$YC$</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CADAC</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Cyclomatic Complexity</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$CFC$</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$CW$</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

where $Bt = ANDs + ORs + XORs$ such as shown in Eq. (2).

let

\[
\text{ANDs} = 4 \\
\text{ORs} = 0 \\
\text{XORs} = 0
\]

then

\[
Ns = 1 + 1 + 0 + 8 + 4
\]

$Ns = 14$

step 2: calculate $As$

$As = 16$

step 3: calculate $C_{AND}$ using Eq. (3), $C_{XOR}$ using Eq. (4), $C_{OR}$ using Eq. (5), and $C_{cyc}$ using Eq. (6).

let

\[
n_1 = 3 \quad n_2 = 2
\]

$C_{AND} = 4 \cdot (3! + 2!)$

$C_{AND} = 32$

while in Fig. 6 (b) XOR branch, OR branch, and loop are not found, so $C_{XOR} = 0$, $C_{OR} = 0$, $C_{cyc} = 0$.

step 4: calculate $D_{Avg}$ using Eq. (7).

Table 6. Metric measurement results of business process models

<table>
<thead>
<tr>
<th>Business Process Models</th>
<th>$Ns$</th>
<th>$Acs$</th>
<th>$As$</th>
<th>$C_{AND}$</th>
<th>$C_{XOR}$</th>
<th>$C_{OR}$</th>
<th>$Dm$</th>
<th>$C_{cyc}$</th>
<th>$D_{Avg}$</th>
<th>$D_{max}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig. 1 (a)</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1.000</td>
<td>1</td>
</tr>
<tr>
<td>Fig. 1 (b)</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>1.000</td>
<td>1</td>
</tr>
<tr>
<td>Fig. 1 (c)</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>1.000</td>
<td>1</td>
</tr>
<tr>
<td>Fig. 2 (a)</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>1.500</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Fig. 2 (b)</td>
<td>9</td>
<td>5</td>
<td>10</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>1.600</td>
<td>2</td>
</tr>
<tr>
<td>Fig. 3 (a)</td>
<td>10</td>
<td>6</td>
<td>11</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>1.500</td>
<td>2</td>
</tr>
<tr>
<td>Fig. 3 (b)</td>
<td>11</td>
<td>7</td>
<td>12</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>1.429</td>
<td>2</td>
</tr>
<tr>
<td>Fig. 3 (c)</td>
<td>12</td>
<td>8</td>
<td>13</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>1.375</td>
<td>2</td>
</tr>
<tr>
<td>Fig. 4 (a)</td>
<td>12</td>
<td>8</td>
<td>14</td>
<td>96</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>1.500</td>
<td>2</td>
</tr>
<tr>
<td>Fig. 4 (b)</td>
<td>12</td>
<td>8</td>
<td>15</td>
<td>480</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>1.625</td>
<td>2</td>
</tr>
<tr>
<td>Fig. 5 (a)</td>
<td>12</td>
<td>8</td>
<td>13</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>1.375</td>
<td>2</td>
</tr>
<tr>
<td>Fig. 5 (b)</td>
<td>12</td>
<td>8</td>
<td>13</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>1.375</td>
<td>2</td>
</tr>
<tr>
<td>Fig. 5 (c)</td>
<td>12</td>
<td>8</td>
<td>13</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>1.375</td>
<td>2</td>
</tr>
<tr>
<td>Fig. 6 (a)</td>
<td>14</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>1.625</td>
<td>2</td>
</tr>
<tr>
<td>Fig. 6 (b)</td>
<td>14</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>1.875</td>
<td>3</td>
</tr>
<tr>
<td>Fig. 6 (c)</td>
<td>14</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>2.000</td>
<td>3</td>
</tr>
<tr>
<td>Fig. 7 (a)</td>
<td>9</td>
<td>5</td>
<td>10</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>1.600</td>
<td>2</td>
</tr>
<tr>
<td>Fig. 7 (b)</td>
<td>9</td>
<td>5</td>
<td>10</td>
<td>0</td>
<td>42</td>
<td>7</td>
<td>0</td>
<td>1.600</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Fig. 8 (a)</td>
<td>12</td>
<td>8</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>0.273</td>
<td>1.125</td>
<td>2</td>
</tr>
<tr>
<td>Fig. 8 (b)</td>
<td>12</td>
<td>8</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>0.500</td>
<td>1.250</td>
<td>2</td>
</tr>
<tr>
<td>Fig. 8 (c)</td>
<td>12</td>
<td>8</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>0.692</td>
<td>1.375</td>
<td>2</td>
</tr>
<tr>
<td>Fig. 8 (d)</td>
<td>12</td>
<td>8</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>14</td>
<td>0.857</td>
<td>1.500</td>
<td>2</td>
</tr>
<tr>
<td>Fig. 8 (e)</td>
<td>12</td>
<td>8</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>0.545</td>
<td>1.250</td>
<td>2</td>
</tr>
<tr>
<td>Fig. 8 (f)</td>
<td>12</td>
<td>8</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>0.750</td>
<td>1.375</td>
<td>2</td>
</tr>
<tr>
<td>Fig. 8 (g)</td>
<td>12</td>
<td>8</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>1.000</td>
<td>1.500</td>
<td>2</td>
</tr>
<tr>
<td>Fig. 8 (h)</td>
<td>14</td>
<td>8</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>1.000</td>
<td>1.500</td>
<td>2</td>
</tr>
<tr>
<td>Fig. 9 (a)</td>
<td>14</td>
<td>8</td>
<td>16</td>
<td>24</td>
<td>6</td>
<td>0</td>
<td>9</td>
<td>1.875</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Fig. 9 (b)</td>
<td>14</td>
<td>8</td>
<td>16</td>
<td>8</td>
<td>9</td>
<td>0</td>
<td>8</td>
<td>1.875</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Fig. 9 (c)</td>
<td>14</td>
<td>8</td>
<td>16</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0.667</td>
<td>1.875</td>
<td>2</td>
</tr>
</tbody>
</table>

The results of the business process model metric calculations are presented in Table 6. Then the metrics are used to calculate complexity metrics with CADAC as in Eq. (13), Cyclomatic Complexity as in Eq. (10), CFC as in Eq. (11), CW as in Eq. (12), and Scale as in Eq. (14). The results of calculating the complexity metrics are presented in Table 7.

**Table 7. Comparison of complexity metrics among business process models**

<table>
<thead>
<tr>
<th>Business Process Models</th>
<th>YC</th>
<th>CADAC</th>
<th>Cyclomatic Complexity</th>
<th>CFC</th>
<th>CW</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig. 1 (a)</td>
<td>19.000</td>
<td>21</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Fig. 1 (b)</td>
<td>21.000</td>
<td>23</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>Fig. 1 (c)</td>
<td>23.000</td>
<td>25</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>Fig. 2 (a)</td>
<td>45.000</td>
<td>48</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>Fig. 2 (b)</td>
<td>65.400</td>
<td>51</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>19</td>
</tr>
<tr>
<td>Fig. 3 (a)</td>
<td>66.000</td>
<td>53</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>21</td>
</tr>
<tr>
<td>Fig. 3 (b)</td>
<td>67.000</td>
<td>55</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>23</td>
</tr>
<tr>
<td>Fig. 3 (c)</td>
<td>68.250</td>
<td>57</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>25</td>
</tr>
<tr>
<td>Fig. 4 (a)</td>
<td>143.000</td>
<td>58</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>26</td>
</tr>
<tr>
<td>Fig. 4 (b)</td>
<td>529.750</td>
<td>59</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>27</td>
</tr>
<tr>
<td>Fig. 5 (a)</td>
<td>68.250</td>
<td>57</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>25</td>
</tr>
<tr>
<td>Fig. 5 (b)</td>
<td>68.250</td>
<td>57</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>25</td>
</tr>
<tr>
<td>Fig. 5 (c)</td>
<td>68.250</td>
<td>57</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>25</td>
</tr>
<tr>
<td>Fig. 6 (a)</td>
<td>84.750</td>
<td>64</td>
<td>4</td>
<td>2</td>
<td>8</td>
<td>60</td>
</tr>
<tr>
<td>Fig. 6 (b)</td>
<td>88.250</td>
<td>78</td>
<td>4</td>
<td>2</td>
<td>8</td>
<td>60</td>
</tr>
<tr>
<td>Fig. 6 (c)</td>
<td>90.000</td>
<td>78</td>
<td>4</td>
<td>2</td>
<td>8</td>
<td>60</td>
</tr>
<tr>
<td>Fig. 7 (a)</td>
<td>50.400</td>
<td>49</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>57</td>
</tr>
<tr>
<td>Fig. 7 (b)</td>
<td>83.400</td>
<td>54</td>
<td>3</td>
<td>7</td>
<td>7</td>
<td>133</td>
</tr>
<tr>
<td>Fig. 8 (a)</td>
<td>40.023</td>
<td>52</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>48</td>
</tr>
<tr>
<td>Fig. 8 (b)</td>
<td>42.000</td>
<td>52</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>48</td>
</tr>
<tr>
<td>Fig. 8 (c)</td>
<td>43.942</td>
<td>52</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>48</td>
</tr>
<tr>
<td>Fig. 8 (d)</td>
<td>45.857</td>
<td>52</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>48</td>
</tr>
<tr>
<td>Fig. 8 (e)</td>
<td>42.045</td>
<td>52</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>48</td>
</tr>
<tr>
<td>Fig. 8 (f)</td>
<td>44.000</td>
<td>52</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>48</td>
</tr>
<tr>
<td>Fig. 8 (g)</td>
<td>46.000</td>
<td>52</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>48</td>
</tr>
<tr>
<td>Fig. 8 (h)</td>
<td>51.000</td>
<td>55</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>116</td>
</tr>
<tr>
<td>Fig. 9 (a)</td>
<td>86.250</td>
<td>62</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>120</td>
</tr>
<tr>
<td>Fig. 9 (b)</td>
<td>73.250</td>
<td>62</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>120</td>
</tr>
<tr>
<td>Fig. 9 (c)</td>
<td>80.917</td>
<td>60</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>120</td>
</tr>
</tbody>
</table>

**4.3 Discussion of experimental results**

In this section, we show the effect of several cases on the Yaqin complexity formula, namely.

**Case 1 Adding elements to the business process model**

The Yaqin complexity metric increases with the increasing number of elements in the business process model. Fig. 1 shows the increase in the \( A_{sc} \) and \( A_{nd} \) in the sequential business process model. Table 6 shows that \( N_s \) and \( A_{sc} \) are different so that it affects the business process model complexity. The difference occurs because the Yaqin complexity formula involves \( N_s \) and \( A_{sc} \). Fig. 2 shows the addition of \( A_{cs} \) and \( n \) in a business process model with AND branching. Table 6 shows that \( N_s \), \( A_{cs} \), \( C_{AND} \), and \( D_{avg} \) are different. The addition of \( A_{cs} \) and \( n \) affects \( D_{avg} \) and \( C_{AND} \), which also affects the complexity of the business process model. The difference occurs because the Yaqin complexity formula also involves \( C_{AND} \) and \( D \). Fig. 3 shows the addition of \( A_{cs} \) in sequence to a business process model that has AND branches. Table 6 shows that \( N_s \), \( A_{sc} \), and \( D_{avg} \) are different, but the \( C_{AND} \) is the same. The addition of \( A_{cs} \) in sequence to a business process model that has AND branches affects the complexity of the business process model. The results of calculating the business process model complexity in Table 7 show that the Yaqin complexity metric is different in Fig. 1, Fig. 2, and Fig. 3. This case proves that the addition of \( A_{cs} \) affects \( N_s \) and \( A_{sc} \), and in some cases, also affects \( D_{avg} \) and \( C_{AND} \), so it affects the business process model complexity. Table 7 shows that CADAC and Scale are different, so it can be said that this case affects the CADAC and Scale. This case does not affect Cyclomatic Complexity as evidenced by Fig. 1 and Fig. 3, each having the same complexity metric and only in Fig. 2, which has a different complexity metric. This case also does not affect CFC, where Fig. 1, Fig. 2, and Fig. 3 have the same CFC. This case also does not affect CW, wherein Fig. 1, the CW is the same, and in Fig. 2, and Fig. 3, the CW is the same too. Thus it can be concluded that in this case, the Yaqin complexity metric is more sensitive than Cyclomatic Complexity, CFC, and CW, but equally sensitive to CADAC and Scale.

**Case 2 Add branches to a branch**

The Yaqin complexity metric increases with the number of branches in a branch. Fig. 3 (c) and Fig. 4 show the addition of branches in an AND branch with the same \( N_s \). This difference affects the complexity of
the business process model because \( C_{\text{AND}} \) involves the number of branches in its formula in Eq. (3). In this case, the AND branch is used as an example. The XOR and OR branches also have the same effect according to the formulas in Eq. (4) and Eq. (5). The calculation results of the business process model complexity in Table 7 show that the Yaqin complexity metric in Fig. 3 (c) and Fig. 4 are different. This case proves that increasing \( n \) in a branch affects \( C_{\text{AND}} \) and \( D_{\text{Avg}} \), thus also affecting the Yaqin complexity metric of the business process model. Table 6 also shows that CADAC, Scale, and Cyclomatic Complexity differ due to differences in As, so it can be said that this case affects the CADAC, Scale, and Cyclomatic Complexity. Adding branches to a branch will add As. In CADAC and Scale, branching logic will affect the complexity metrics, but the Cyclomatic Complexity does not affect. This case does not affect CFC and CW because they do not involve the number of branches in its complexity calculation. The results of the CFC and Cognitive Complexity calculations, in this case, are the same. Thus it can be concluded that the Yaqin complexity metrics more sensitive than CFC and CW, but it is equally sensitive to CADAC, Scale, and Cyclomatic Complexity.

Case 3 Shifting the location of the branches
The Yaqin complexity metric is not affected by shifting branching locations. In Fig. 3 (c) and Fig. 5, each shows an AND branching with three branches with different branching locations. Fig. 3 (c) and Fig. 5 do not show differences in \( N_s, A_s, C_{\text{AND}}, \) and \( D_{\text{Avg}} \) have shown in Table 6. The calculation of complexity in Table 7 also does not show differences. This case proves that the branching location shift does not affect the complexity of the business process model. In this case, Table 7 shows that CADAC, Cognitive Complexity, CFC, CW, and Scale are also not affected by shifting branching locations. Thus, the branching location shift does not affect the complexity of the business process model.

Case 4 Depth changing in the business process model
The Yaqin complexity metric changes when \( D \) changes in the business process model. Fig. 6 shows business process models with \( Acs \) equal to various depths. Table 6 shows that \( D_{\text{max}} \) for Fig. 6 (a) and Fig. 6 (b) is different, as shown in Fig. 6 (d) and Fig. 6 (e). While Fig. 6 (b), Fig. 6 (c), and Fig. 6 (d) \( D_{\text{max}} \) are the same. Fig. 6 (a) and Fig. 6 (b) show business process models with 2 AND branches that differ in \( D \). Table 7 shows that the results of the Yaqin complexity metric calculation in Fig. 6 (a) and Fig. 6 (b), are different even though \( N_s, A_s, \) and \( C_{\text{AND}} \) are the same. \( D_{\text{max}} \) causes different Yaqin complexity metrics. The Yaqin complexity metric in Fig. 6 (b) and Fig. 6 (c) also differs even though \( N_s, A_s, C_{\text{AND}}, \) and \( D_{\text{max}} \) are the same. \( D_{\text{Avg}} \) is causing different Yaqin complexity metrics. Table 7 shows that CADAC in Fig. 6 (a) and Fig. 6 (b) are different, whereas CADAC in Fig. 6 (b) and Fig. 6 (c) are the same. These cases show that CADAC only accommodates \( D_{\text{max}} \) changes. Cyclomatic Complexity, CFC, CW, and Scale do not provide differences in depth in the business process model. Thus, the Yaqin complexity metric is the most sensitive in accommodating depth to the business process model. CADAC is less sensitive in providing depth to business process models. Whereas Cyclomatic Complexity, CFC, CW, and Scale are not sensitive in accommodating depth to business process models.

Case 5 Changes in branching logic in the business process model
The Yaqin complexity metric changes when the branching logic changes in the business process model. This change in branching logic is shown in Fig. 2 (b) and Fig. 7. Fig. 2 (b) shows a business process model with two AND branching logic. In contrast, Fig. 7 (a) shows a business process model with XOR branching logic, and Fig. 7 (b) shows a business process model with OR branching logic. All three business process models have the same \( N_s, A_s, \) and \( D_{\text{Avg}} \), but differ in terms of \( C_{\text{AND}}, C_{\text{XOR}}, \) and \( C_{\text{OR}} \) shown in Table 6. The Yaqin complexity metric shows the differences in all three, as shown in Table 7. Table 7 also shows that CADAC, CFC, CW, and Scale differ in the three business process models, while in Cyclomatic Complexity, the same. Cyclomatic complexity only pays attention to \( N_s \) and \( A_s \) without paying attention to the logic of branching. Thus it can be concluded that the Yaqin complexity metric is more sensitive than Cyclomatic Complexity and equally sensitive to CADAC, CFC, CW, and Scale.

Case 6 Number of nodes involved in a loop, and the location of the nodes in that loop
The Yaqin complexity metric is affected by \( Acs \) involved in a loop, and the location of activities in that loop. Fig. 8 represents this case with the same \( N_s \), \( A_s \) and one loop with a variety of loop patterns, except for Fig. 8 (h), which contains two loops. Fig. 8 (b) and Fig. 8 (e) show a business process model
that involves two activities in one loop with different locations that produce different Yaqin complexity metrics. Likewise, Fig. 8 (c) and Fig. 8 (f) include three activities in a loop with different positions. Fig. 8 (d) and Fig. 8 (g) involve four activities in one loop with different locations. In Table 6, the business process models differ in their cyclical complexity involving diameters as defined in Definition 7. Table 7 also shows that CADAC, Cyclomatic Complexity, CFC, Cognitive Complexity, and Scale are all the same except in Fig. 8 (h) because CADAC, Cyclomatic Complexity, CFC, CW, and Scale do not accommodate loops. It can be concluded that the Yaqin complexity metric is more sensitive than CADAC, Cyclomatic Complexity, CFC, CW, and Scale.

Case 7 Number of the loop in the business process model

The Yaqin complexity metric is affected by the number of loops in the business process model. Fig. 8 (d) and Fig. 8 (g) show a business process model with one loop involving four activities, while Fig. 8 (h) shows a business process model with two loops involving four activities. Table 6 shows As differences in the business process models. As the number of loops increases, As also increases. The Yaqin complexity metrics for the business process models are shown in Table 7. Table 7 also shows that CADAC, Cyclomatic Complexity, CFC, CW, and Scale are all different because of the difference in As. Thus it can be concluded that the metric complexity of Yaqin is as sensitive as CADAC, Cyclomatic Complexity, CFC, CW, and Scale.

Case 8 Branching logic type exchange

The Yaqin complexity metric is affected by the exchange of branching types in the business process model. Fig. 6 (b) and Fig. 9 show business process models with the same structure with variations in the location of branching types. Fig. 6 (b) shows a business process model with two AND branches. Fig. 9 (a) and Fig. 9 (b) show business process models with AND and XOR branching variations. Fig. 9 (c) shows a business process model with a loop. The structure of the business process models is the same, but the branching logic is different. Table 6 shows that \( N_s \), \( A_s \), and \( D_{avg} \) for Fig. 6 (b) and Fig. 9 are the same. \( C_{AND} \) and \( C_{XOR} \) for Fig. 6 (b) and Fig. 9 differ because the number of branches in each branch is different, so the Yaqin complexity metric shown in Table 7 is also different. Table 7 shows that CADAC is not affected by the branching type exchange described in Fig. 9 (a) and Fig. 9 (b) because CADAC does not involve the number of branches in its formula. Still, in Fig. 9 (c), CADAC is affected by loops in the business process model. Meanwhile, Cyclomatic Complexity is not affected at all in this case, because Cyclomatic Complexity only involves \( N_s \) and \( A_s \). CFC, CW, and Scale in Fig. 6 (b), and Fig. 9 are different because the combination of branching types is also different. Thus it can be concluded that the order of sensitivity of the most sensitive is the metric complexity of Yaqin. The next order is CADAC, then CFC, CW and Scale together are almost insensitive. Cyclomatic complexity is not sensitive.

The analysis shows that the Yaqin complexity proved to be more sensitive to small changes in the business process model, because the different composition of elements can produce different complexity metric values. The third case produces the same complexity metric because the Yaqin complexity formula only accommodates changes in structure and does not accommodate changes in behavior in the business process model. The summary of the cases above can be seen in Table 8.

### 4.4 Theoretical validation

The Yaqin complexity metric is then validated with Weyuker's properties. Weyuker's properties have nine properties that are used to test the validity of the Yaqin complexity metric. The nine properties are:

**Property 1: Non-coarseness :** \((3P)(3Q)\); \(|P| \neq |Q|\). Complexity metrics can distinguish two different process models, so they produce different metrics. The Yaqin complexity metric can distinguish two different business process models; this can be seen in Table 7, where each business process model discussed has a different complexity metric. Thus property 1 has been fulfilled by Yaqin complexity metrics.

**Property 2: Granularity:** Complexity metrics must be non-negative. Table 7 shows Yaqin complexity

<table>
<thead>
<tr>
<th>Case</th>
<th>YC</th>
<th>CADAC</th>
<th>Cyclomatic Complexity</th>
<th>CFC</th>
<th>CW</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>5</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>6</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>7</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>8</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>
metric values, which are always positive, and if we
look at the Eq. (7), the complexity metric values
never to be negative. Thus property 2 has been
fulfilled by Yaqin complexity metrics.

Property 3: Non-uniqueness : (P ≠ Q); (|P| = |Q|). In some cases, two different business process
models can have the same complexity metrics. The
Yaqin complexity also fulfills this property. This
property is proven in Fig. 3 (c), Fig. 5 (a), (b), and (c),
where they have different business process model
structures but have the same complexity metric.

Property 4: Design Implication : (∃P)(∃Q); (P ≡ Q & |P| ≠ |Q|). Complexity metrics must be able to
distinguish between two business process models that
have the same function but have different
implementation details. The Yaqin complexity metric
can distinguish business process models that have
different structures but have the same functions as
those in property 1. Thus the Yaqin complexity
metric fulfills this property.

Property 5: Monotonicity : (∀P)(∀Q); (|P| ≤ |P;| & |Q| ≤ |P;Q|). Complexity metrics must be able
to show the unification of several new elements in
a business process model. The Yaqin complexity
metric can indicate the unification of new elements in
a business process model. The business process
model in Fig. 1 (a) and Fig. 1 (b) connected sequentially
to produce a complexity metric of 23. This
figure has more significant complexity than the
complexity of Fig. 1 (a) of 19 and Fig. 1 (b) of 21.
Thus the Yaqin complexity metric fulfills property 5.

Property 6: Non-equivalence of Interaction:
6a. (∃P)(∃Q)(∃R); (|P| = |Q| & |P; R| ≠ |Q; R| & ∗ ∈ {−, ∇, ; , }).
6b. (∃P)(∃Q)(∃R); (|P| = |Q| & |R; P| ≠ |R; Q| & ∗ ∈ {−, ∇, ; , }).
Complexity metrics must be able to distinguish two
identical processes. When there are three business
process models P, Q, and R with the same metric
value, then the business process model P is connected
with R or Q is connected with R, then the two metric
values should not be the same. The simple cases used
for validation on this property are Fig. 5 (a) as the
business process model P, Fig. 5 (b) as the business
process model Q, and Fig. 1 (a) as the business
process model R. When the business process model
P is connected with the business process model R,
and the business process model Q is connected with
business process models R, both of which produce
the same complexity metric. Thus the Yaqin
complexity metric does not fulfill property 6.

<table>
<thead>
<tr>
<th>Properties</th>
<th>YC</th>
<th>CADAC</th>
<th>Cyclomatic Complexity</th>
<th>CFC</th>
<th>CW</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>5</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>6</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>7</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>8</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>9</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Property 7: Significance of Permutation:
(∃P)(∃Q); (|P| ≠ |Q|) → If Q is formed by the
sequence of permutations of activity P. Workflow
patterns have been identified using a combination
formula and permutation so that the logic difference
in the branching and the number of branches in each
branching significantly affect this complexity metric.
The prove of this property can be seen in Fig. 6 (b)
and Fig. 9 and the results of the metric calculations in
Table 7. Thus the Yaqin complexity metric fulfills
this property 7.

Property 8: No change in renaming. Changing names
or labels on the business process model elements does
not change complexity metrics. The Yaqin
complexity metric bases the calculation on the
structure of the business process model so that
changes to the label do not affect the value of the
complexity metric. Thus the Yaqin complexity metric
fulfills property 8.

Property 9: Interaction Complexity :
(∃P)(∃Q); (|P| + |Q| < |P;Q|). The combined
complexity metrics of several business process
models must be greater than the sum of the
complexity metrics of each. This property has already
been discussed in property 5. Thus the Yaqin
complexity metric fulfills property 9.

Based on this analysis, the Yaqin complexity
metric meets 8 of 9 Weyuker’s properties. As a
comparison, we also present the Weyuker’s
properties evaluation of CADAC [15], Cyclomatic
Complexity [28], CFC [29], CW [28], and Scale. The
summary can be seen in Table 9.

5 Conclusions

Based on the previously elaborated discussion,
we conclude that:

1. The Yaqin complexity metric has advantages in
detecting differences in the number of branches,
differences in the structure of nested branches,
and loops in the business process model.
2. The Yaqin complexity formula proved to be more comprehensive by involving seven parameters than the other complexity formulas that involve fewer parameters.
3. The Yaqin complexity metric proved to be more sensitive by affecting 7 of the 8 cases provided than the other complexity metric.
4. The Yaqin complexity metric has been validated with Weyuker’s properties, which state that 8 out of 9 properties have been met. In addition to some of the results above, this research also still needs some improvements, namely:
   1. The Yaqin complexity formula is only tested to measure the complexity of structured business process models, so in the next research, it is necessary to examine the character of the Yaqin complexity formula in measuring the complexity of unstructured business process models.
   2. Only split logic branching is measured by the Yaqin complexity formula, while the join logic is ignored. Therefore in the next research, the complexity of split and join logic branching must be considered.
   3. The results of this research can also be used to measure the growth of enterprise architecture, explicitly determining the configuration of information systems based on the complexity of the business architecture.

Conflicts of Interest

The authors (Muhammad Ainul Yaqin, Riyanarto Sarno, and Siti Rochimah) declare no conflict of interest.

Author Contributions

Conceptualization, Muhammad Ainul Yaqin; methodology, Muhammad Ainul Yaqin; validation, Muhammad Ainul Yaqin; formal analysis, Muhammad Ainul Yaqin; investigation, Muhammad Ainul Yaqin; resources, Muhammad Ainul Yaqin; data curation, Muhammad Ainul Yaqin; writing—original draft preparation, Muhammad Ainul Yaqin; writing—review and editing, Riyanarto Sarno and Siti Rochimah; visualization, Muhammad Ainul Yaqin; supervision, Riyanarto Sarno and Siti Rochimah.

References


