International Journal of Electrical and Computer Engineering (IJECE)
Vol. 15, No. 3, June 2025, pp. 2938~2950
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i3.pp2938-2950

Generation of business process modeling notation diagrams from textual functional requirements in Indonesian

Sholiq Sholiq[1], Muhammad Ainul Yaqin[2], Apol Pribadi Subriadi[1], Bambang Setiawan[1]
[1]Department of Information Systems, Institut Teknologi Sepuluh Nopember Surabaya, Surabaya, Indonesia
[2]Department of Informatics, Universitas Islam Negeri Maulana Malik Ibrahim, Malang, Indonesia

Article Info

Article history:

Received Jul 23, 2024
Revised Mar 7, 2025
Accepted Mar 20, 2025

Keywords:

Business process modeling notation diagrams
Functional requirement
Indonesian
Natural language processing
Text

ABSTRACT

This study proposes a method for converting textual functional requirements in Indonesian to Business Process Modeling Notation (BPMN) diagrams has not been found in previous studies. The use of BPMN diagrams to present software functional requirements has the advantage of being better in terms of presenting sequential activities than using use case diagrams. On the other hand, the requirements obtained from clients in the requirements collection session are more in the form of user stories, namely text in natural language. The method used in this study is to integrate natural language processing and a set of mapping rules and BPMN diagram generation rules. The proposed method is tested with 15 functional requirement cases from three applications, namely mini hospital software, employee cooperatives, and stores. Then, the results are compared with diagrams made by experts for the same cases. The test results show an accurate level of the proposed method of 94.4%.

Corresponding Author:

Sholiq Sholiq
Department of Information Systems, Institut Teknologi Sepuluh Nopember Surabaya
Kampus ITS Sukolilo Surabaya, Indonesia
Email: sholiq@is.its.ac.id

1. INTRODUCTION

In the early stages of software project development, the requirements obtained from the client in the functional requirements collection session are more in the form of user stories, namely text in natural language [1]–[3]. To get a better understanding of the requirements, the user stories need to be changed to visual diagrams including Business Process Modelling and Notation (BPMN) diagrams. BPMN is a graphical standard for modelling business processes in easy-to-understand diagrammatic form, facilitating communication between technical and non-technical stakeholders [4]. BPMN enables detailed process visualization, helps identify inefficiencies and areas for improvement, and ensures uniform understanding among team members [5]. BPMN also supports workflow automation, increases productivity, reduces operational costs, and facilitates "what-if" analysis for more informed decision-making [6], [7]. At first, BPMN diagrams were employed for modelling processes of business within organizations, however next time, these diagrams are initialized for requirements visualization [8], [9]. The study [10] found no significant distinction between BPMN diagrams and use case diagrams for understanding software requirements. Therefore, BPMN diagrams are slightly preferable rather than Use case diagrams for depicting series activities. BPMN diagrams can also be transformed into other artefacts in software development, for example, class diagrams [11] and the transformation of BPMN into a use case diagram [12]. Meanwhile, business processes in the form of data flow diagrams are generated in the study [13].

Conversion from text to BPMN diagrams has been done by researchers. The studies [14], [15] translated text input into BPMN diagrams by utilizing Cooperative Requirements Engineering with Scenarios

as an intermediary. However, they failed to consider complex sentence structures in the text, and the method in [11] requires sequential sentences in input, which prevents iterative processing. Likewise, this study [16] extracted the input text into a BPMN diagram, but not all BPMN elements were generated from the text. The study [17] employed a spreadsheet-based description as an intermediary to generate BPMN diagrams. This approach offers the benefit of simplicity in producing the diagrams, but it has three major drawbacks. First, it can only manage a limited range of BPMN elements, such as tasks and gateways, and is incapable of handling intermediate events, pools, lanes, data stores, or data objects. Second, the activity extraction process was not validated. Third, the approach was not tested with complex or compound sentence structures in text. The study [18] improves the above studies by obtaining an accuracy rate of 92.83%. The above studies were conducted using English input text, which in Indonesian has not been found until now.

Generating BPMN diagrams from Indonesian language functional requirements presents various challenges. First, Indonesian has different sentence structures and grammar to English. Indonesian has variations in the use of words, phrases and conjunctions which can cause ambiguity and difficulty in understanding the context by machines. Second, functional requirements written in Indonesian require the identification of different keywords such as activities, decisions, and workflows required for creating BPMN diagrams [19]. Third, there is a lack of resources and tools that specifically support natural language processing (NLP) for Indonesians. Most existing NLP tools and datasets focus more on English, so developing tools for Indonesian requires additional effort in building models and training on appropriate data [20]. Therefore, this paper proposes the conversion of textual functional requirements in Indonesian to BPMN diagrams to address the above three challenges. Also to fill the gap in knowledge about the conversion of text for Indonesian functional requirements to BPMN diagrams.

The paper structure begins with an introduction that explains the background of research, problems, objectives, and contributions. Related work reviews previous research related to text conversion to BPMN diagrams. The method is divided into several sub-sections: textual analysis using NLP, Fact Type extraction, diagram generation, and evaluation. Results and discussion discuss the experimental results and performance analysis. The conclusion summarizes the main findings and potential further research.

2. BPMN DIAGRAM FROM NATURAL LANGUAGE

The generation of BPMN based on natural language input has attracted the attention of many people to research it. Sholiq *et al.* [18] proposed an approach to generate BPMN from the English natural language using semantics of business vocabulary and business rules (SBVR) definitions, rule sets, and spreadsheet-based descriptions. The method was tested with ten case studies with an accuracy of 92.83%. Mößlang *et al.* [21] presented a method for generating BPMN from audio recordings converted into text. They analysed process descriptions in depth to generate process models in extensible markup language (XML) format, achieving an accuracy rate of 100% and increasing generation speed by 48%. Indahyanti and Siahaan [22] conducted a survey on techniques for automatically generating business process models from heterogeneous documents using NLP approaches, semantic knowledge engineering, and ontology bases. Schüler and Alpers [23] identified various techniques for generating business process models, including those based on source code, business rules (such as SBVR), data tables, models (such as Use cases), knowledge bases (such as ontologies) [24], event logs, and natural text.

Various approaches have been proposed, including a machine translation-like approach that achieves more than 81% similarity with manual models [25] and spreadsheet-based methods with semantic analysis [17]. The use of NLP is also prominent, as in the systematic literature review by Bordignon *et al.* [26] and the identification of NLP tools by Maqbool *et al.* [27], which simplified the creation of BPMN models from text. Tools such as BPMN Sketch Miner [28] combined natural language logging with process mining to produce real-time BPMN diagrams, emphasizing usability and immediate feedback. Methodological evaluations show high-accuracy results; for example, Fatimah *et al.* [29] reported achieving an average F-measure of 96.68%.

The studies above used English text input, while this study uses Indonesian text for functional requirements. As explained in the previous section, creating BPMN diagrams from Indonesian texts has its challenges due to differences in grammatical rules, the need for more understanding to find keywords, and the lack of tool support. In addition, this study also proposes the use of a different approach from previous studies. This study proposes using an intermediate representation, which is a combination of diagram generation rules and substitution of input text with identified Fact Types.

3. METHOD

There are five main activities carried out in this study as presented in Figure 1. Each activity and sub-activity are explained sequentially. The final goal is to obtain a BPMN diagram for which the level of accuracy of the resulting diagram is measured.

3.1. Setup

The setup contains data collection activities that are set in the form of functional requirements in Indonesian text. Data is collected from software requirement specification (SRS) documents of the mini-hospital, employee cooperative, store application. The functional requirements in text form collected from user stories in the SRS document of each application.

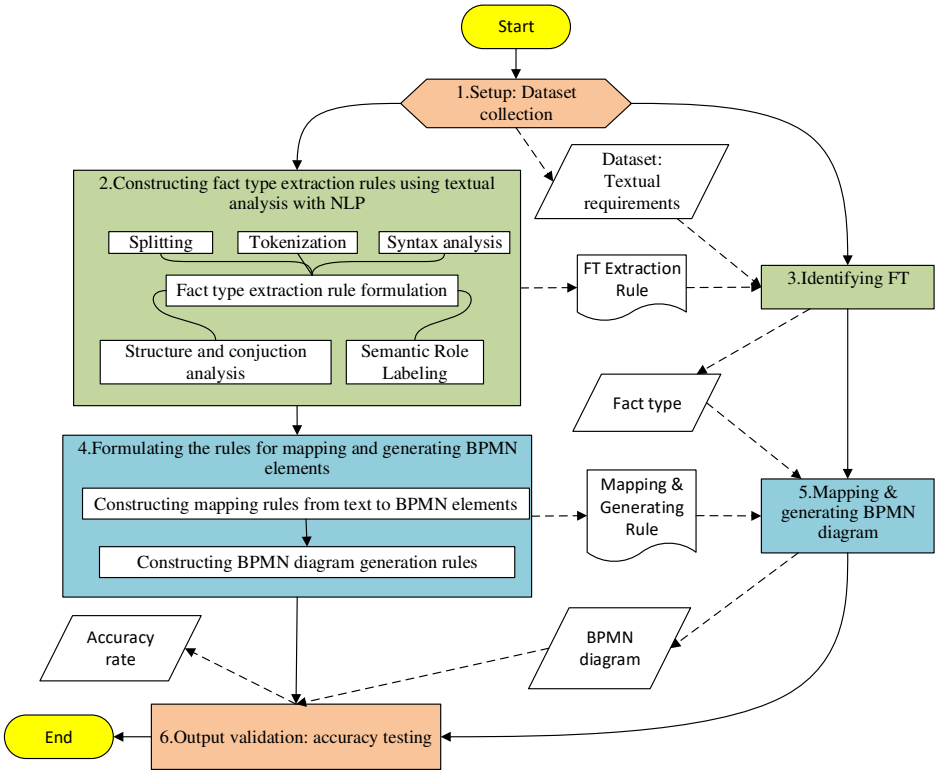

Figure 1. Methods of this research

3.2. Constructing the Fact Type extraction rules using textual analysis with NLP

Input data in the form of Indonesian text is processed using the rules of Fact Type extraction. This rule is composed of textual analysis using NLP and analysis of sentence structure and conjunctions. We are using Stanza version 1.6.0 which already supports the use of Indonesian [30]. The following is an explanation of each of the processes needed in this phase.

3.2.1. Splitting and tokenization

Splitting is the process of separating an input text into compilation sentences. Then, tokenization is breaking down a sentence into its compiler words [31]. For example, a sentence in Indonesian below.

'Bagian IGD ingin membuat rujukan untuk pasien.'

The sentence is broken down into '[Bagian][IGD] [ingin] [membuat] [rujukan] [untuk] [pasien] [.]'.

3.2.2. Syntax analysis

The analysis of syntax involves two steps: parts of speech (POS) tagging and generating a dependency tree. POS tagging entails detecting every token as a verb, noun, adjective, preposition, and so on. For instance, in the sentence provided, the token is:

{'Bagian/NSD IGD/X-- ingin/VSA membuat/VSA rujukan/NSD untuk/R-- pasien/NSD ./Z--}.

Dependency trees can help detect the connections between diverse syntactic structures in textual requirement inputs. For example, the dependency breakdown for the example sentence above is displayed in Figure 2.

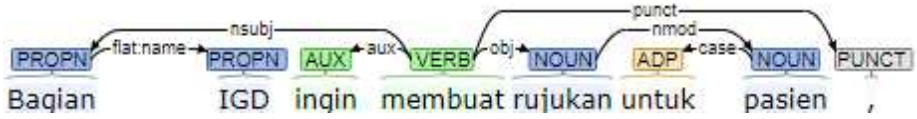

Figure 1. Dependency tree of the first clause of the first sentence

3.2.3. Analysis of semantic

Semantic analysis is carried out by assigning semantic role labels or semantic role labeling (SRL). It assigns a sticker to every mark/token in a sentence automatically. We follow [32] and [33], where roles are derived from SBVR elements, such as noun concepts/general concepts, concepts of individual nouns, concepts of verbs, attributes, quantification, and others [34].

3.2.4. Identification of sentence structure and conjunctions

A sentence is a collection of words that at least consist of a subject and a predicate and have a complete meaning. In a sentence, at least there must be an element of Subject (S) and an element of Predicate (P). S is a topic that is being discussed, it can be in the form of a person, thing, place, or situation. P is the thing that explains the subject. According to the number of clauses that form it, sentences can be formed into three types: i) Single sentences, ii) Parallel compound sentences, and iii) multi-layered or complex compound sentences [35].

a. Single sentence

A single sentence is a sentence that has one independent clause. That means there is only one P in a single sentence. The P element is a marker of the clause. The S and P elements are always mandatory in every sentence. The elements of Object (O), Complement (C), and Description (D) are not required to be present in the sentence, including in a single sentence. If P still needs to be completed, then the complementary element is presented.

Based on the type of word/phrase that fills P, a single sentence can be grouped into six types that are given according to the type of word or phrase [36]: i) A verb sentence is a sentence whose predicate is a verb or verbal phrase, ii) An adjective sentence, is a sentence whose predicate is an adjective or an adjective phrase, iii) A nominal sentence, is sentence with predicates in the form of nominals or nominal phrases, iv) A prepositional sentence, is a sentence with its predicate form in the form of a prepositional phrase, v) A numeral sentence, is a sentence with a predicate in the form of numeric, and vi) An adverbial sentence is a sentence whose sentence form is in the form of an adverbial.

Verb sentences are divided into 2 groups, namely transitive verb sentences and intransitive verb sentences. Transitive verb sentences require an object or thing. Meanwhile, intransitive sentences do not require objects. The author in this reference says [37] that verbs that use the prefix ‘me (N)’ tend to be transitive, while verbs starting with 'ber-' are intransitive.

b. Parallel/coordinative compound sentences

A compound sentence is a sentence that is a combination of two or more single sentences. A parallel/coordinating compound sentence is a combination of two or more main ideas that are in equal position. The sentence structure in which there are at least, two basic sentences and each can stand alone as a single sentence. The conjunctions that connect clauses in parallel compound sentences are quite numerous. These conjunctions designate several types of relationships and have several functions. Table 1. indicates the connecting clause in a parallel compound sentence.

Table 1. Conjunctions in parallel compound sentences

Types of relationships	Function	Conjunction
Joining	Expresses the sum or combination of events, activities, or processes	dan, serta, baik, maupun
Contradicting	States that the statement in the first clause contradicts the second clause	tetapi, lainya, kalau tidak, sedangkan, bukannya, melainkan
Choosing	Expresses a choice between two possibilities	atau
Sequencing	Declares sequential events	lalu, kemudian

c. Multilevel/complex/subordinating compound sentences

This compound sentence is a single sentence in which one of the positions is expanded to form a new sentence. In a multi-level compound sentence, there is a parent clause and a child clause. The parent clause is fixed or unchanged, while the child clause is an expanded clause position forming a new clause. The child clause is marked with the use of a connecting word and if it precedes the parent clause it is separated by a

comma punctuation. Table 2 shows the types of relationships between clauses, conjunction, and their functions in a multi-layered compound sentence.

Table 2. Conjunctions in complex compound sentences

Types of Relationships	Conjunction
Time	sejak, sedari, sewaktu, sementara, seraya, setelah, sambil, sehabis, sebelum, ketika, tatkala, hingga, sampai
Condition	jika, seandainya, andaikata, andaikan, asalkan, kalau, apabila, bilamana, manakala
Purpose	agar, supaya, untuk, biar
Consensual	walau(pun), meski(pun), sekalipun, biar(pun), kendati(pun), sungguh(pun)
Comparison	seperti, bagaikan, laksana, sebagaimana, daripada, alih-alih, ibarat
Cause	sebab, karena, oleh karena
Consequences	sehingga, sampai-sampai, maka
Methods/Tools	dengan, tanpa
Similarity	seolah-olah, akan
Reality	padahal
Explanation	bahwa

3.2.5. Extraction rule formulation

We define a set of rules for extracting FT from Indonesian text as presented in Figure 3. This rule is used to obtain FT from input in the form of functional requirements in the form of Indonesian language text. FT identification using this rule will be explained in section 3.3.

1. Split an input text into the sentences that make it up.
2. Split a compound sentence into single sentences according to the number of clauses in the sentence.
3. For a passive sentence, change it to an active sentence.
4. For a single sentence:
   a. A subject consisting of some nouns or pronouns separated by commas or/and conjunctions (the list of conjunctions is in Table and Table). Break up the sentence into some new sentences, with as many nouns as the subject has.
   b. For predicates with multiple verbs separated by commas or conjunctions like in point a), Divide the sentence into new sentences with as many verbs in the predicate.
5. Recognize FT as a pair of a subject and a predicate.
   a. A single sentence of transitive verbs is recognized as a Binary Fact Type.
   b. In addition to point a) identified as a Unary Fact Type.

Figure 3. Rules for extracting FT from Indonesian input text

3.3. Fact Type identification

This activity is to get Fact Types (FT) where the combination of the concept of noun and the concept of verb is an FT [34]. In SBVR 1.5, noun phrases with 1or more roles of verb concept construct an FT [34]. An FT with a single role is named unary Fact Type (UFT), as an example, ‘Sekolah itu indah’. A Fact Type with two roles is referred to as a BFT, for example, 'Seorang polisi mengatur lalu lintas', in which ‘Seorang polisi’ and ‘lalu lintas’ are two roles, and ‘mengatur’ is a verb connected to the two roles. In the type of single sentences that have been discussed in the previous section, we categorize single sentence types with FT types given in Table 3.

Table 3. Mapping single sentence types to FT types

Types of sentences	Types of FT
Transitive verbs	Binary
Intransitive verbs	Unary
Adjective	Unary
Nominal	Unary
Prepositional	Unary
Numeral	Unary
Adverbial	Unary

3.4. Formulating the rules for mapping and generating BPMN elements

This section contains rules for mapping textual requirements to BPMN elements and generating BPMN diagrams as shown in Figure 4. BPMN elements mapped using this rule include pool, lane, activity,

event, gateway (AND, OR, and XOR), and data store/object. Then, the elements that have been obtained will be made into a diagram which will be discussed in the next section.

1. Map the headings of the input functional requirements as a Pool.
2. Map the noun concept of FT (BFT and UFT) into a Lane.
3. Create a Start Event in the first lane and an end event in the last lane.
4. Map a BFT to an Activity.
5. Map an UFT to an Events except UFT in a condition conjuction (in Indonesian ‘jika, seandainya, andaikata, andaikan, asalkan, kalau, apabila, bilamana, manakala’).
6. Map the following:
   a. Joining conjuction (in Indonesian 'dan', ‘serta’, ‘baik’) becomes a pair of AND branches.
   b. Choosing conjuction (in Indonesian 'atau') becomes a pair of OR branches.
   c. A combination of two conjunctions: condition-consequences, condition conjunction (in Indonesian 'jika', ‘seandainya’, ‘andaikata’ and so on, see Table 2), and consequences conjunction (in Indonesian 'sehingga', ‘sampai-sampai’, or ‘maka’, see Table 2), becomes a pair of XOR branches.
   d. A combination of three conjunctions: condition-consequence-contradicting, contradicting conjunction (in Indonesian 'tetapi', ‘lainya’, ‘kalau tidak’, see Table 1), becomes a pair of XOR branches.
7. Map the repeating mark:
   a. The beginning of a loop (*<number>) becomes the initial XOR branch.
   b. End of a loop (jump <number> in Indonesian ‘lompat <number>’) to final XOR branch.
8. Add a sequence flow line:
   a. From an activity, event, or branch to the next activity, event, or branch.
   b. And the end of a loop to the beginning of a loop.
   c. The two lines from the opening AND, OR, and XOR branches to the activity/event/branch within the branch. Then from the activity/event/branch to the closing branch of the opening branch pair.
9. The noun (or noun phrase) of the verb concept in the Fact Type (FT), containing the keyword verb (listed below), is mapped to the data store/object inflow or outflow.
   a. If the object phrase includes the term 'database', then it is mapped to the datastore. If it doesn't exist, it's mapped to a data object.
   b. Create an association line of the activity to the data store/object, an inflow line that contains keyword verbs for inflow and an outflow that contains keyword verbs for outflow.

   Keyword verbs for inflow: menulis, menambah, menyimpan, mencatat, mendaftarkan, memperbarui, mengedit, dan mengarsipkan.

   Keyword verbs for outflow: membaca, memeriksa, memverifikasi, mendapatkan, memperoleh, memindai, mengambil, memfilter, dan mengambil.

Figure 4. Rules for mapping and generating BPMN diagrams

3.5. Mapping and generating BPMN diagrams

In this activity, we use the rulebase in Figure 4 to map the FT to the BPMN elements and then generate the BPMN diagram using the same rulebase. Examples with case studies are given in subsection 4.1. The case studies are only in the appendix (15 case studies).

3.6. Output validation

To measure the validity of the output, we measure its proximity to the results obtained from the manual creation of BPMN diagrams by experts. We apply the mean magnitude of relative error (MMRE) as in (1) in the estimation of the error rate of the software as the author did in this [38]. MMRE is considered by the entire observation using (2).

$$MRE_i = \frac{Difference_i}{ManualExpert_i} \cdot 100\% \quad (1)$$

$$MMRE = \frac{1}{N}\sum_1^N MRE_i \quad (2)$$

4. RESULTS AND DISCUSSION

4.1. Example of BPMN diagram generation

For example, in this section, we use functional requirements in the development of mini-hospital software. There are 15 case studies that we use as test data as presented in Appendix (A and B). The case study used as an example is case study 14, namely, register referral patients given below.

‘*Bagian IGD ingin membuat rujukan untuk pasien, kemudian sistem menampilkan formulir register rujukan pasien. *1 Bagian IGD memasukkan ID rekam medis, lalu sistem mencari data pasien di database Pendaftaran dan Pasien. Jika pasien ada, maka sistem menampilkan hasil kueri termasuk nama dan alamat, kalau tidak loncat ke 1. Bagian IGD mencakup tujuan rujukan, informasi medis, tindakan atau pengobatan, dan informasi*

lainnya, memilih tombol. Jika tombol adalah simpan, maka sistem menyimpan informasi rujukan ke database Pasien dan Rujukan. Jika tombol adalah batal, maka sistem mengatur semua informasi ke kosong.'
In English
*The Emergency Department wants to make a referral for a patient, then the system displays the patient referral registration form. *1 The Emergency Department enters the medical record ID, then the system searches for patient data in the Registration and Patient database. If the patient exists, the system displays the query results including name and address, otherwise jump to 1. The Emergency Department includes the purpose of the referral, medical information, actions or treatments, and other information, selecting the button. If the button is save, the system saves the referral information to the Patient and Referral database. If the button is cancel, the system sets all information to blank.*

The case study demonstrates single sentences, parallel compounds, and complex compounds. Meanwhile, examples of multilevel compound sentences are given in case study 15 (Appendix A). This case study also demonstrates loops, if-then-else conditions, and verbal and nominal clauses. The textual requirement of the case study has six sentences consisting of parallel compound sentences (sentences 1 and 2) and complex compound sentences (sentences 3, 5 and 6). In sentence 2, there is a database keyword that begins with the verb 'mencari' which indicates a database reading (Registrasi and Pasien).

The process of generating BPMN diagrams consists of several steps. First, the textual requirement is extracted to obtain TF using NLP and sentence analysis. In this process, there is splitting, tokenization (or POS tagging), syntax analysis (or generation of the dependency tree), SRL, and structure and conjunction analysis. The splitting of this example, which is the textual requirements of Case Study 14, is broken down into 6 sentences. Example tokenization for the first sentence: *[Bagian] [IGD] [ingin] [membuat] [rujukan] [untuk] [pasien] [,] [kemudian] [sistem] [menampilkan] [formulir] [register] [rujukan] [pasien] [.]*. POS tagging for the first sentence: *{Bagian/NSD IGD/X-- ingin/VSA membuat/VSA rujukan/NSD untuk/R-- pasien/NSD,/Z-- kemudian/S—sistem/NSD menampilkan/VSA formulir/NSD register/F-- rujukan/NSD pasien/NSD ./Z--}*. Meanwhile, the generation of the dependency tree for the first clause of Sentence 1 in Case Study 14 is given in Figure 2. Next, the SRL identifies the noun concept and the verb concept as the main element pairs to get the Fact Type (the results can be seen in Table 4).

Table 4. The extraction of FT on case study 14

SBVR concept	Qty	Breakdown
Noun/individual concept	15	Bagian (Department), rujukan (referral), pasien (patient), sistem (system), rekam (record), data (data), pendaftaran (registration), hasil (result), nama (name), alamat (address), tujuan (purpose), informasi (information), tindakan (action), pengobatan (treatment), tombol (button).
Verb concept	15	Ingin (want), membuat (make), menampilkan (display), memasukkan (enter), mencari (search), ada (exists), menampilkan (display), termasuk (include), loncat (jump), mencakup (include), memilih (select), menyimpan (save), mengatur (set).
UFT	3	UFT1: Pasien ada (The patient exists). UFT2: Tombol adalah simpan (The button is saved). UFT3: Tombol adalah batal (The button is cancelled).
BFT	9	BFT1 (A)=Bagian IGD ingin membuat rujukan untuk pasien (The Emergency Department wants to make a referral for a patient). BFT2 (B)=Sistem menampilkan formulir register rujukan pasien (The system displays the patient referral registration form). BFT3 (C)=Bagian IGD memasukkan ID rekam medis (The Emergency Department enters the medical record ID). BFT4 (D)=Sistem mencari data pasien di database Pendaftaran dan Pasien (The system searches for patient data in the Registration and Patient database). BFT5 (E)=Sistem menampilkan hasil kueri (The system displays the query results). BFT6 (F)=Bagian IGD mencakup tujuan rujukan, informasi medis, tindakan atau pengobatan, dan informasi lainnya (The Emergency Department includes the purpose of the referral, medical information, actions or treatments, and other information). BFT7 (G)=Bagian IGD memilih tombol (The Emergency Department selects the button). BFT8 (H)=Sistem menyimpan informasi rujukan ke database Pasien dan Rujukan (The system saves the referral information to the Patient and Referral database). BFT9 (I)=Sistem mengatur semua informasi ke kosong (The system sets all information to blank).

The next process is to extract the input text to get FT which is extracted using the rules set in Figure 3. The results of the extraction process for this case study are given in Table 4 which obtained 3 UFT and 9 BFT. A description of the extraction process applying the rule set is provided in Table 5. This case study requires rules #1, #2, #4b, #5a, and #5b.

Table 5. Extraction uses rules set for case study

Rule	Short description	Input/example	Output
#1	Rule #1: the input is divided into six distinct sentences.	The text of case study 14	6 sentences
#2	Examples of parallel compound sentences connected with the conjunction 'kemudian (then)'.	For example, Sentence 1: 'Bagian IGD ingin membuat rujukan untuk pasien, kemudian sistem menampilkan formulir register rujukan pasien (The Emergency Department wants to make a referral for a patient, then the system displays the patient referral registration form).'	Broken into 2 sentences: – Bagian IGD ingin membuat rujukan untuk pasien (The Emergency Department wants to make a referral for a patient). – Sistem menampilkan formulir register rujukan pasien (The system displays the patient referral registration form).
#4b	This example shows a sentence breakdown that contains sequential verbs (rule #4b)	For example, Sentence 4: 'Bagian IGD mencakup tujuan rujukan, informasi medis, tindakan atau pengobatan, dan informasi lainnya, memilih tombol simpan (The Emergency Department includes the purpose of the referral, medical information, actions or treatments, and other information, selecting the button).'	Broken into 2 sentences: – Bagian IGD mencakup tujuan rujukan, informasi medis, tindakan atau pengobatan, dan informasi lainnya (The Emergency Department includes the purpose of the referral, medical information, actions or treatments, and other information). – Bagian IGD memilih tombol simpan (The Emergency Department selects the button).
#5a	This example shows rule #5a identifying the single verbal of a transitive sentence.	Clause 2 in Sentence 3: Sistem menampilkan hasil kueri termasuk nama dan alamat (The system displays the query results including name and address).	Sistem menampilkan hasil kueri termasuk nama dan alamat (The system displays the query results including name and address) as BFT.
#5b	In contrast, this example shows the identification of a nominal sentence as UFT using rule #5b.	Clause 1 in Sentence 3: Jika pasien ada (If the patient exists).	Pasien ada (The patient exists) as UFT.

The second step involves mapping and generating BPMN diagrams, which includes two sub-tasks: i) substitution of FT to input text, and ii) implementation of rules for mapping and diagram generation. Substitution of FT to the following input text: *'BFT1, kemudian BFT2. *1 BFT3, lalu BFT4. Jika UFT1, maka BFT5, kalau tidak loncat ke 1. BFT6, BFT7. Jika UFT2, maka BFT8. Jika UFT3, maka BFT9*'. Then the result of the substitution of FT to the input text is completed with the concept noun as follows:

*'BFT1/Bagian IGD, kemudian BFT2/Sistem. *1 BFT3/ Bagian IGD, lalu BFT4/Sistem. Jika UFT1/Pasien, maka BFT5/Sistem, kalau tidak loncat ke 1. BFT6/Bagian IGD, BFT7Bagian IGD. Jika UFT2/Tombol, maka BFT8/Sistem. Jika UFT3/Tombol, maka BFT9/Sistem*'.

Mapping and generation of BPMN diagrams using the rules set in Figure 4. The title of the input text, namely 'Register referral patients' becomes the title in the Pool (Rule #1), and BFT1 to BFT9 are mapped to activities A to I (Rule #4), so that 'bagian IGD' and 'Sistem' become Lane (Rule #2). UFT1 to UFT3 is not an event because it is in a condition conjunction (Rule #5), while condition-consequence-contradicting conjunctions are XOR branch pairs (Rules #6c and #6d). The beginning of the repetition *1 becomes the XOR of the opening branch and the end of the repetition (jump to 1) becomes the XOR of the closing branch (Rule #7). Then, sequence flow lines are added according to Rule #8, and Datastores 'Pendaftaran' and 'Pasien' are created based on Rule #9 because of the key verbs 'mencari' and 'database'. The diagram of BPMN obtained is given in Figure 5.

4.2. The results of 15 case studies applied

This section covers the findings from 15 case studies included in Appendix A. The FT extraction results, and a brief description of the case studies are also provided in Appendix A. Meanwhile, the BPMN diagram of the generation results is given in Appendix B.

4.3. Finding and discussion

As stated in section 3.4 to test the results of the proposed method, it is measured in proximity to the results of making BPMN diagrams by an expert. Experts, in this case, are those who have well in the field of software requirements engineering, BPMN modelling, and Indonesian. Experts made BPMN diagrams for the same 15 case studies. The results are tabled, and then MRE and MMRE are calculated using (1) and (2). Table shows a comparison of the results of making BPMN diagrams using the proposed methods and those carried out by an expert. Comparisons are made per BPMN element: activities, events, branches, data stores/ objects, and pools/lanes.

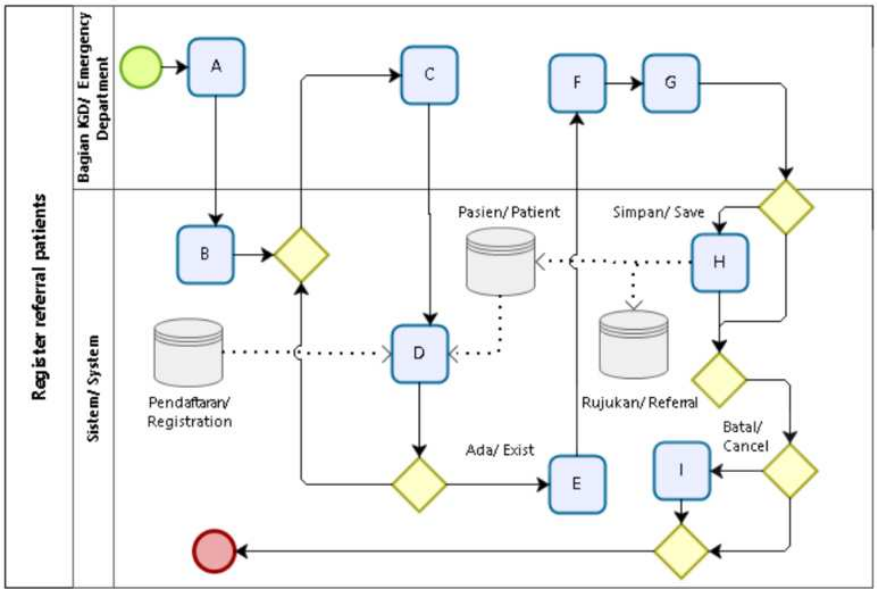

Figure 5. The generated BPMN diagram from case study 14

The tests that have been carried out have obtained an MMRE error rate of 5.6% which means that the accuracy level of the proposed method is 94.4% compared to the BPMN diagram made by experts. If further analysis is carried out, it is found that the deviation is all in the generation of branches or gateways. In Case Studies 6, 7, 10, 11, 12, 14, and 15 that had branching, all of them had deviations of 11,1%, 11,8%, 10.5%, 12.5%, 25%, 9.5%, and 7.4%. This is because the set of rules reads the text sequentially, for example, if two sentences contain if-then or if-then-else, then two pairs of XOR Gateways are created in order, while an expert considers this to be a pair of XOR Gateways that have 3 or more options. For example, for Case Study 11, the last two sentences indicate the existence of two pairs of XOR gateways, while according to an expert, it is enough with a pair of XOR Gateways. For future studies, it is necessary to refine the set of mapping rules and generation of BPMN diagrams to overcome these weaknesses.

When compared to previous studies, this study has advantages, namely: i) This study has a slightly higher accuracy than the previous study of 94.4%, while the previous study was 92.83% [18], 81% [25], and 91.92 [16]. ii) This study does not use the medium of intermediate representation but uses a set of rules and substitutions of text with FT to generate BPMN diagrams so that it is easier to manage them. And iii) this study uses functional requirements input in the form of Indonesian texts that have not been found in previous studies.

This study offers significant benefits in the fields of software engineering, business process management, and information technology. We have identified several key contributions of this research, namely: i) Automation of BPMN Diagram Creation in Indonesian: This study is the first to develop a method for automating the creation of BPMN diagrams directly from functional requirements texts in Indonesian. This contribution not only accelerates manual processes but also expands the scope of natural language processing (NLP) technology to support the Indonesian language, which remains underrepresented in global research; ii) Acceleration of the software development lifecycle: By enabling the rapid transformation of requirement texts into business process diagrams, this study speeds up the software development lifecycle. Automatically generated BPMN diagrams allow development teams to quickly visualize system requirements and begin the implementation phase; iii) Enhancing accuracy and efficiency: The automated approach reduces the risk of misinterpretation commonly encountered in manual diagramming. More accurate results enable more precise identification of functional requirements while improving efficiency in business process design; iv) Supporting multistakeholder communication: BPMN is a visual language comprehensible to various stakeholders, both technical and non-technical. By generating diagrams from functional requirement texts in Indonesian, this research clarifies communication between business analysts, developers, and end-users, minimizing potential misunderstandings; and v) Foundation for further studies in software metrics: BPMN diagrams generated from texts can have their complexity analyzed, as discussed in [39]. Therefore, this study opens a new pathway for quantitative research based on business processes; and vi) application of technology for local contexts: This research not only advances NLP but also contributes to the development of technology that caters to local needs. It facilitates the adoption of automated BPMN technology by Indonesian companies, reducing reliance on international solutions that may not fully align with the language and cultural context. This study not only

provides practical solutions for accelerating and enhancing the efficiency of BPMN diagram creation but also makes a significant scientific contribution to expanding the scope of NLP and business process studies, particularly in the context of the Indonesian language. It establishes a strong foundation for further research in the fields of information technology and business process management.

Table 6. Evaluation of proposed methods and experts for the generation of BPMN diagrams

Req no	Activity		Event		Branch		Pool/ Lane		Data store/ object		∑			MSE (%)
	PM	Exp	PM	Exp	PM	Exp	PM	Exp	PM	Exp	PM	Exp	Dev	
1	3	3	2	2	0	0	3	3	1	1	9	9	0	0
2	6	6	2	2	0	0	3	3	1	1	12	12	0	0
3	6	6	2	2	0	0	3	3	1	1	12	12	0	0
4	8	8	2	2	0	0	3	3	4	4	17	17	0	0
5	9	9	2	2	2	2	3	3	2	2	18	18	0	0
6	8	8	2	2	6	4	3	3	1	1	20	18	2	11.1
7	9	9	2	2	4	2	3	3	1	1	19	17	2	11.8
8	8	8	2	2	4	4	3	3	3	3	20	20	0	0
9	11	11	2	2	6	6	3	3	3	3	24	24	0	0
10	9	9	2	2	6	4	3	3	1	1	21	19	2	10.5
11	7	7	2	2	4	2	3	3	2	2	18	16	2	12.5
12	8	8	6	6	10	4	3	3	3	3	30	24	6	25.0
13	7	7	2	2	4	4	3	3	2	2	18	18	0	0
14	9	9	2	2	6	4	3	3	3	3	23	21	2	9.5
15	15	15	2	2	6	4	3	3	3	3	29	27	2	7.4
							MMRE (%)							5.6

Abbreviations: Dev=Deviation, Exp=Expert, PM=Proposed Method, Req=Requirements

Meanwhile, the limitations of this study and the proposals for the next study include i) The BPMN diagram visualization automation process has not been carried out in this study, thus opening up opportunities to automate the BPMN diagram generation visualization process, so this will make it more practical and user friendly; ii) Mapping and generating rules that cannot handle XOR gateways with multiple choices. For the next study, it is necessary to think about it so that it can be done; and iii) This study needs to be tested with more case studies and involves different types of software in addition to business software applications, such as embedded software, web applications, mobile applications, and others.

5. CONCLUSION

This paper proposes a new method to generate BPMN diagrams from functional requirements in Indonesian text. This study is interesting because, in the early stages of software development, the functional requirements obtained from the client or system owner are usually in the text known as user stories. To get functional requirements in visual form to make them easier to understand, one of them is presented in a BPMN diagram. For this purpose, it is necessary to generate BPMN diagrams from functional requirements in the form of text.

Unlike previous studies, this study does not use intermediate representation media to generate BPMN diagrams. In this study, the substitution of Fact Type to input text is implemented, and then the mapping and diagram generation rule sets are applied. Therefore, the proposed method is simpler and faster to generate diagrams. The test results from examining 15 case studies on the functional requirements of the mini-hospital software application showed an MMRE error rate of 5.6%. This indicates that the proposed method achieves an accuracy rate of 94.4%.

The next study that can be carried out is to improve the set of mapping rules and the generation of BPMN diagrams so that it is possible to get an XOR gateway with multiple choice. The next study also needs to automate the BPMN diagram visualization process with the input of textual functional requirements. Expanded case study testing with embedded software applications, web applications, mobile applications, and others also needs to be carried out in future studies.

FUNDING INFORMATION

This work was supported by the ITS 2024 Scientific Research scheme under Grant No. 1181/PKS/ITS/2024.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Sholiq Sholiq	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
Muhammad Ainul Yaqin	✓	✓			✓			✓	✓	✓				
Apol Pribadi Subriadi		✓	✓			✓						✓		
Bambang Setiawan			✓									✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors (Sholiq Sholiq, Muhammad Ainul Yaqin, Apol Pribadi Subriadi, and Bambang Setiawan) declare no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are openly available in Mendeley Data at https://data.mendeley.com/datasets/9427ww7v22/1, reference number [40].

REFERENCES

[1] A. Ahmed and B. Prasad, *Foundations of software engineering*. Auerbach Publications, 2016. doi: 10.1201/9781315369495.
[2] E. S. Calisaya, "Analysis of natural language scenarios," Universidad Nacional de San Agustin, 2016.
[3] F. Dalpiaz, A. Ferrari, X. Franch, and C. Palomares, "Natural language processing for requirements engineering: the best is yet to come," *IEEE Software*, vol. 35, no. 5, pp. 115–119, 2018, doi: 10.1109/MS.2018.3571242.
[4] M. von Rosing, S. White, and H. de Man, *Business process model and notation-BPMN*, vol. 1. 2017.
[5] D. Lübke, "A BPMN profile for test case execution visualization," in *CEUR Workshop Proceedings*, vol. 3673, pp. 33–40, 2024.
[6] P. Lin, X. Li, Z. Dong, and L. Zhang, "A BPMN-engine based process automation system," in *ICIEA 2022 - Proceedings of the 17th IEEE Conference on Industrial Electronics and Applications*, vol. 301, pp. 765–769, 2022, doi: 10.1109/ICIEA54703.2022.10006152.
[7] R. Waszkowski, T. Nowicki, and K. Worwa, "Corporate efficiency improvement with business process automation," *MATEC Web of Conferences*, vol. 210, Oct. 2018, doi: 10.1051/matecconf/201821002012.
[8] C. Monsalve, A. April, and A. Abran, "On the expressiveness of business process modeling notations for software requirements elicitation," in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, Oct. 2012, pp. 3132–3137. doi: 10.1109/IECON.2012.6389398.
[9] A. Przybyłek, "A business-oriented approach to requirements elicitation," in *ENASE 2014 - Proceedings of the 9th International Conference on Evaluation of Novel Approaches to Software Engineering*, pp. 152–163, 2014, doi: 10.5220/0004887701520163.
[10] O. L. Vega-Márquez, J. Chavarriaga, M. Linares-Vásquez, and M. Sánchez, "Requirements comprehension using BPMN: an empirical study," *Empirical Studies on the Development of Executable Business Processes*, pp. 85–111, 2019, doi: 10.1007/978-3-030-17666-2_5.
[11] H. M. Achraf, E. Redouane, and L. El Mazoui Nadori Yasser, "Transforming the business process diagram into a class diagram by model-driven architecture," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 29, no. 2, pp. 845–851, 2023, doi: 10.11591/ijeecs.v29.i2.pp845-851.
[12] O. Rabhi, S. Filali, and M. Erramdani, "The transformation method from business processes models by BPMN to use cases diagram by UML in agile methods," in *The International Conference on Artificial Intelligence and Smart Environment*, 2023, pp. 384–390.
[13] N. Kharmoum, S. Retal, K. El Bouchti, W. Rhalem, and S. Ziti, "An automatic alignment of the business process and business value models: A novel MDA method," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 30, no. 1, pp. 501–509, 2023, doi: 10.11591/ijeecs.v30.i1.pp501-509.
[14] F. Friedrich, J. Mendling, and F. Puhlmann, "Process model generation from natural language text," in *International Conference on Advanced Information Systems Engineering*, Springer Berlin Heidelberg, 2011, pp. 482–496. doi: 10.1007/978-3-642-21640-4_36.
[15] J. C. de A. R. Gonçalves, F. M. Santoro, and F. A. Baião, "Let me tell you a story - On how to build process models," *Journal of Universal Computer Science*, vol. 17, no. 2, pp. 276–295, 2011.
[16] R. C. B. Ferreira, L. H. Thom, and M. Fantinato, "A semi-automatic approach to identify business process elements in natural language texts," in *Proceedings of the 19th International Conference on Enterprise Information Systems*, 2017, vol. 3, pp. 250–261. doi: 10.5220/0006305902500261.
[17] K. Honkisz, K. Kluza, and P. Wiśniewski, "A concept for generating business process models from natural language description," in *International Conference on Knowledge Science, Engineering and Management*, vol. 11061, Cham: Springer, 2018, pp. 91–103.

doi: 10.1007/978-3-319-99365-2_8.
[18] S. Sholiq, R. Sarno, and E. S. Astuti, "Generating BPMN diagram from textual requirements," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, pp. 10079–10093, 2022, doi: 10.1016/j.jksuci.2022.10.007.
[19] K. M. Habibullah, G. Gay, and J. Horkoff, "Non-functional requirements for machine learning: understanding current use and challenges among practitioners," *Requirements Engineering*, vol. 28, no. 2, pp. 283–316, Jun. 2023, doi: 10.1007/s00766-022-00395-3.
[20] H. Riza, E. Nurfadhilah, M. T. Uliniansyah, A. Santosa, and L. R. Aini, "An overview of BPPT's Indonesian language resources," in *Proceedings of the 12th Workshop on Asian Language Resources*, 2016, pp. 73–77.
[21] M. Mößlang, R. Bernsteiner, C. Ploder, and S. Schlögl, "Automatic generation of a business process model diagram based on natural language processing," in *Knowledge Management in Organisations*, 2024, pp. 237–247.
[22] U. Indahyanti and D. Siahaan, "Auto-generating business process model from heterogeneous documents: A comprehensive literature survey," *2022 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pp. 239–243, 2022, doi: 10.23919/EECSI56542.2022.9946460.
[23] S. Schüler and S. Alpers, "State of the art: Automatic generation of business process models," in *Business Process Management Workshops*, 2024, pp. 161–173.
[24] A. P. Yanuarifiani, F.-F. Chua, and G.-Y. Chan, "Automating business process model generation from ontology-based requirements," in *Proceedings of the 2019 8th International Conference on Software and Computer Applications*, 2019, pp. 205–209. doi: 10.1145/3316615.3316683.
[25] R. Sonbol, G. Rebdawi, and N. Ghneim, "A machine translation like approach to generate business process model from textual description," *SN Computer Science*, vol. 4, no. 3, Mar. 2023, doi: 10.1007/s42979-023-01742-z.
[26] A. C. de A. Bordignon, L. H. Thom, T. S. Silva, V. S. Dani, M. Fantinato, and R. C. B. Ferreira, "Natural language processing in business process identification and modeling: a systematic literature review," in *Simpósio Brasileiro de Sistemas de Informação (SBSI)*, 2018. doi: 10.1145/3229345.3229373.
[27] B. Maqbool *et al.*, "A comprehensive investigation of BPMN models generation from textual requirements—techniques, tools and trends," in *ICISA 2018: Information Science and Applications 2018*, 2019, pp. 543–557. doi: 10.1007/978-981-13-1056-0_54.
[28] A. Ivanchikj, S. Serbout, and C. Pautasso, "From text to visual BPMN process models," in *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems*, Oct. 2020, pp. 229–239. doi: 10.1145/3365438.3410990.
[29] I. N. Fatimah *et al.*, "USESPEC to BPMN: Web generator program for use case specification to BPMN," in *The 8Th International Conference and Workshop on Basic and Applied Science (Icowobas) 2021*, 2023, vol. 2554. doi: 10.1063/5.0103694.
[30] Stanford NLP Group, "Stanza – A Python NLP package for many human languages," GitHub, 2023. https://stanfordnlp.github.io/stanza/ (accessed Jul 23, 2024).
[31] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning, "Stanza: A Python natural language processing toolkit for many human languages," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020, pp. 101–108. doi: 10.18653/v1/2020.acl-demos.14.
[32] I. S. Bajwa, "A natural language processing approach to generate SBVR and OCL," The University of Birmingham, 2012.
[33] I. S. Bajwa, M. G. Lee, and B. Bordbar, "SBVR business rules generation from natural language specification," *AAAI Spring Symposium - Technical Report*, vol. SS-11-03, pp. 2–8, 2011.
[34] OMG, *Semantics of business vocabulary and business rules version 1.5*. SBVR Annex, 2019.
[35] Putri Malomi Marima, "Contrastive analysis between english and Indonesian sentence patterns," IAIN Padangsidimpuan, 2020.
[36] J. N. Sneddon, A. Adelaar, D. N. Djenar, and M. C. Ewing, *Indonesian: A comprehensive grammar*, 2nd editio. Routledge, 2012.
[37] I. D. P. Wijana, "Me (N) - and Ber- In Indonesian," in *Prosiding Seminar Nasional Linguistik dan Sastra (SEMANTIKS) 2021 "Prospek*, 2021, pp. 96–107.
[38] A. B. Nassif, L. F. Capretz, and D. Ho, "Estimating software effort based on use case point model using sugeno fuzzy inference system," in *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, Nov. 2011, pp. 393–398. doi: 10.1109/ICTAI.2011.64.
[39] S. Sholiq, R. Sarno, E. S. Astuti, and M. A. Yaqin, "Implementation of COSMIC function points (CFP) as primary input to COCOMO II: Study of conversion to line of code using regression and support vector regression models," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 5, pp. 92–103, 2023, doi: 10.22266/ijies2023.1031.09.
[40] S. Sholiq and M. A. Yaqin, "Textual Functional Requirements of Software in Indonesian," *Mendeley Data*. Mendeley Data, 2025.

APPENDIX

Appendix A and B in link:https://data.mendeley.com/datasets/9427ww7v22/1.

BIOGRAPHIES OF AUTHORS

Sholiq Sholiq received his bachelor's degree in mechanical engineering from Institut Teknologi Sepuluh Nopember (ITS), Indonesia, in 1995, and his master's degree in informatics in 2004 and Doctorate in Computer Science in 2023 from the same institute. He is currently an associate professor at the Department of Information Systems, ITS. Currently, he is the head of the information systems management laboratory at ITS. His research interests include software costing, software metrics, software requirements engineering, software modelling, and natural language processing. He can be contacted via email: sholiq@is.its.ac.id or sholiq@its.ac.id.

Muhammad Ainul Yaqin is a permanent lecturer at the Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang. The author earned a bachelor's degree in physics from Airlangga University in 2000, followed by a master's degree in informatics engineering and computer science from the Institut Teknologi Sepuluh Nopember (ITS) in 2004. In 2022, he completed his doctoral studies at the same institute with a focus on business process modeling. His research focuses on software metrics, software growth, and software complexity. He is also active in reputable indexed scientific publications and has been a speaker at various international scientific seminars. He can be contacted via email: yaqinov@ti.uin-malang.ac.id.

Apol Pribadi Subriadi is an associate professor at Department of Information Systems, Institut Teknologi Sepuluh Nopember (ITS), Indonesia. Dr Subriadi has many scientific papers and projects either presented or published, indexed by Scopus, Google Scholar or others. Currently, he is Secretary of the Department at the Department of Information Systems at ITS. He has more than 15 years of experience at Hewlett Packard Company as Manager of System Support and Services. Practically, he is an expert in the area of Account Management, Account Support, Solution Architect, and various Hewlett-Packard products or portfolios. In academics, Dr. Subriadi is interested in the area of research on information technology business value, information technology investment management, information technology performance measurement, enterprise architecture, and business continuity management. Dr. Subriadi holds a doctoral degree from Universitas Brawijaya in Management Science. He can be contacted via email: apol@is.its.ac.id.

Bambang Setiawan received the B.Sc. and master's degrees in informatics from Institut Teknologi Sepuluh Nopember (ITS), Indonesia, in 1994 and 1999, and the same institute's Doctorate in Computer Science in 2020. He is currently Head of the Information Technology Infrastructure and Security Laboratory, ITS. He is also a lecturer at the Department of Information Systems, ITS. His research interests include software requirements engineering, IT adoption, and information security governance. He can be contacted at the following email: setiawan@is.its.ac.id or b.setiawan@its.ac.id.