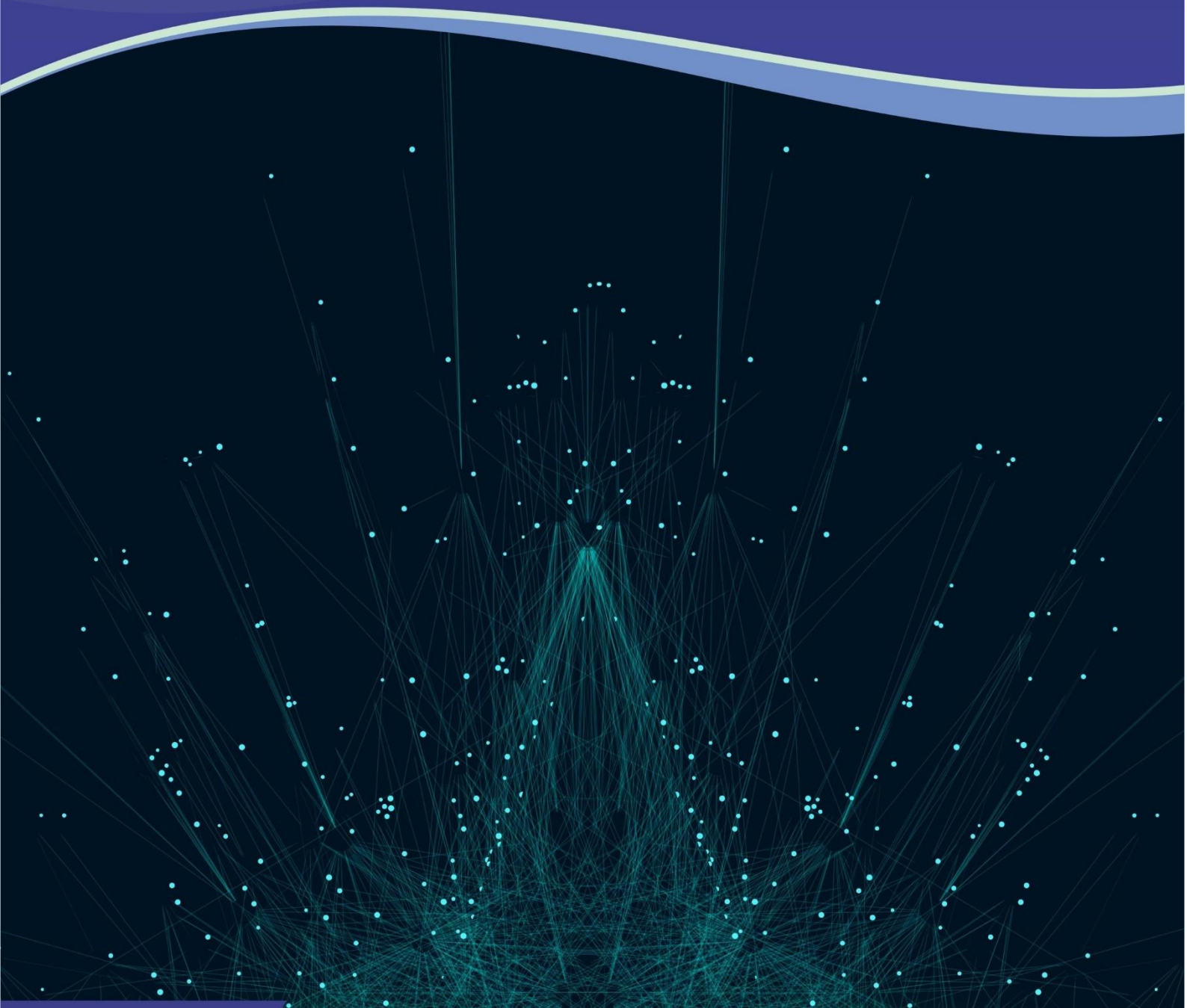
Vol. 13, No. 1 Februari 2020

ISSN *online* : 2442-4528
ISSN *print* : 1979-925X

# Telematika



UNIVERSITAS
AMIKOM PURWOKERTO

# TELEMATIKA

Jurnal Telematika adalah terbitan berkala ilmiah yang fokus pada bidang teknologi informasi, komunikasi dan komputer yang berbentuk kumpulan/akumulasi pengetahuan baru, pengamatan empiric atau hasil penelitian, dan pengembangan gagasan atau usulan baru. Beberapa sub bidang ilmu yang menjadi fokus ilmu Komputer antar lain: 1. Perangkat Lunak, 2. Pemrosesan Sinyal, 3. Sistem Informasi, 4. Interaksi Komputer-Manusia, 5. Perangkat Keras dan Arsitektur, 6. Visi Komputer dan Pengenalan Pola, 7. Data Science, 8. Jaringan Komputer dan Komunikasi, 9. Grafik Komputer dan Desain Berbantu Komputer, 10. Teori Komputasi dan Matematika, 11. Kecerdasan Buatan, 12. Ilmu Komputer (Lain-Lain), 13. Sistem Pendukung Keputusan.

Alamat Redaksi :
Universitas AMIKOM Purwokerto
Jl. Letjend. Pol Soemarto Watumas PurwokertoTelp. (0281) 623321
Fax. (0281) 621662, Website :
www.ejournal.amikompurwokerto.ac.id
Email : telematika@amikompurwokerto.ac.id

# TELEMATIKA

## KATA PENGANTAR

Puji syukur kehadiarat Tuhan Yang maha kuasa atas anugerah dan karunianya sehingga jurnal edisi ini berhasil disusun dan terbit. Beberapa tulisan yang diterbitkan telah melalui koreksi materi dari mitra bestari dan revisi redaksi.

Beberapa pakar di bidang IT juga telah diajak untuk berkolaborasi mengamati penerbitan jurnal ini. Materi tulisan pada jurnal berasal dari dosen, peneliti, praktisi dan ilmuwan. Redaksi mencoba selalu mengadakan pembenahan kualitas dari jurnal dalam beberapa aspek.

Harapan kami semoga jurnal ini dapat terbit secara rutin dan berkelanjutan serta memberi banyak manfaat bagi para pembaca. Untuk itu kritik dan saran sangat kami harapkan dan mohon dialamatkan baik via email, fax maupun disampaikan langsung ke alamat redaksi.

Atas saran dan kritik yang pembaca berikan kami ucapkan terima kasih.

Redaksi

# TELEMATIKA

## DAFTAR ISI

Terbit *online* pada laman web jurnal :
http://ejournal.amikompurwokerto.ac.id/index.php/telematika/

**Telematika**

Terakreditasi Sinta "3" KEMENRISTEKDIKTI, No. 21/E/KPT/2018

# Synonym Measurement Through Semantic Similarity Using the SOC-PMI Method

**Uswatun Hasanah[1], Bambang Pilu Hartato[2], Mitra Yulianti[3], Saeful Haq Faruqi[4]**

[1,2,4]Department of Information Technology
[3]Department of Informatics
Faculty of Computer Science
Universitas Amikom Purwokerto
E-mail: uswatun_hasanah@amikompurwokerto.ac.id[1], bambang.pilu@amikompurwokerto.ac.id[2], mitrayulianti98@gmail.com[3], ruqisaeful@gmail.com[4]

**ARTICLE INFO**

***History of the article:***
Receive December 16, 2019
Revised January 16, 2020
Received February 24, 2020
*Online* February 28, 2020

**Keywords:**
SOC-PMI
Semantic Similarity
Synonym
Corpus-based Method

**Kata Kunci:**
SOC-PMI,
Kemiripan Semantik,
Sinonim,
Metode Berbasis Korpus

**Correspondence:**
Telephone: +62 (857) 12190708
E-mail:
uswatun_hasanah@amikompurwokerto.ac.id

## ABSTRACT

Measurement of synonyms can be an essential task in measuring word similarity. This work cannot be done syntactically but must dig deeper into its semantics. Semantic relations can be anything, such as synonyms, antonyms, hyponymy, homonymy, and polysemy. This research works on finding synonym values using the Second Order Co-occurrence Pointwise Mutual Information (SOC-PMI) method. The data used are 30 questions on the TOEFL exam. Each question consists of one word as a question and four reference answers as alternative answers. The results show very low accuracy (30%) since there are only 9 out of 30 answers that show the synonym. Besides, the LCS method was also tested to get a character-based similarity score. LCS method can achieve a higher similarity score of 43.33%. Finally, the idea of the hybrid method by combining character-based and semantic-based methods can be considered in longer words to produce a fairer similarity score.

## ABSTRAK

Pengukuran sinonim dapat menjadi pekerjaan yang penting dalam mengukur kemiripan kata. Pekerjaan ini tidak dapat dilakukan secara sintaksis, tetapi harus dilakukan dengan menggali lebih dalam tentang semantiknya. Hubungan semantik dapat berupa apa saja, seperti sinonim, antonim, hiponim, homonim, dan polisemi. Penelitian ini berusaha untuk menemukan nilai-nilai sinonim menggunakan metode Second Order Co-occurrence Pointwise Mutual Information (SOC-PMI). Data yang digunakan adalah 30 pertanyaan pada ujian TOEFL. Setiap pertanyaan terdiri dari satu kata sebagai pertanyaan dan empat jawaban referensi sebagai jawaban alternatif. Hasil menunjukkan nilai akurasi yang sangat rendah (30%) karena hanya ada 9 dari 30 jawaban yang benar-benar menunjukkan sinonim. Selain itu, metode LCS juga diuji untuk mendapatkan skor kemiripan berdasarkan karakternya. Metode LCS mampu mencapai skor kemiripan yang lebih tinggi yaitu 43,33%. Akhirnya, gagasan metode hybrid dengan menggabungkan metode berbasis karakter dan metode berbasis semantik semantik dapat dipertimbangkan untuk kata-kata yang lebih panjang agar menghasilkan skor kesamaan yang lebih adil.

## INTRODUCTION

Two concepts or words can be related (or not) by expressing their semantic relatedness (Islam & Inkpen, 2006). The relation of meaning is described by the semantic relationship between one entity and another. In linguistics, entities can be words, phrases, clauses, or sentences. The relationship of meaning to

linguistics includes the similarity of meanings (synonyms), contradictory meanings (antonyms), coverage of meaning (hyponymy), doubling of meaning (homonymy), or excess of meaning (polysemy) (Parera, 2004). Synonyms refer to terms that can be used to describe a particular entity; for example, the entity "Holland" can refer to "Netherlands." In natural language processing applications, entity synonyms play an essential role. Some examples of its application are text summarization (Alguliyev, Aliguliyev, Isazade, Abdi, & Idris, 2017; Barzilay & Elhadad, 1999), query expansion (Aronson & Rindflesch, 1997; Díaz-Galiano, Martín-Valdivia, & Ureña-López, 2009), reformulation (Plovnick & Zeng, 2004), paraphrase detection, and question answering (Ferrucci, 2012).

This study aims to measure the similarity of synonyms by knowing the value of semantic similarity. Since semantic similarity can be of various types, this research limits only the synonym similarity. In research conducted by (Ullmann, 1964) in (Djajasudarma, 1993), synonyms are divided into nine types as follows: 1) Synonyms in which one of its members has a more general meaning, 2) Synonyms in which one of its members has more intensive elements of meaning, 3) Synonyms where one of the members emphasizes emotive meaning, 4) Synonyms where one of the members is reproachful or not justifying, 5) Synonym where one of the members becomes a field term specific, 6) Synonyms where one of its members is more widely used in a variety of written languages, 7) Synonyms where one of its members is more commonly used in conversational languages, 8) Synonyms where one of the members is used in childhood language, 9) Synonyms where one of the members is usually used in certain areas.

The method used in this study considers semantic similarity in measuring synonym similarity. The method, namely, Second-Order Co-occurrence Pointwise Mutual Information (SOC-PMI), was conceived by (Islam & Inkpen, 2006) and has been used in a variety of natural language processing applications. Even though the method focuses on semantic similarity, this research focuses on synonyms, which are one of a series of semantic elements.

## RESEARCH METHODS

### 1. Semantic Similarity

Humans, with their common sense, can recognize the interrelation of a pair of words in various ways. For humans, it is not difficult to judge the relationship between apples and oranges, rather than apples and toothbrushes (Islam & Inkpen, 2006). Semantics can be used in two mechanisms, namely in the detection of similarities and differences (Frawley, 2013). During this time, applications in natural language processing have used semantic similarity measurements, such as in the construction of automated thesaurus (Grefenstette, 1993)(D. Lin, 1998)(Li, Abe, World, & Partnership, 1998), automatic indexing, text annotations and document summarizing (C. Lin, Hovy, & Rey, 2003), text classification, word sense disambiguation (Li et al., 1998)(Lesk, 1986)(Yarowsky, 1992), information extraction and information retrieval (Buckley, Salton, Allan, & Singhal, 1995)(Vechtomova & Robertson, 2014)(Xu & Croft, 2000).

### 2. SOC-PMI

The Second Order Co-occurrence Pointwise Mutual Information method, or from now on referred to as the SOC-PMI method, is a method developed from the predecessor algorithm called PMI-IR. PMI-IR is proposed by (Turney, 2001), and uses the AltaVista Advanced Search query

syntax to calculate probabilities. PMI-IR is a simple method intended to recognize synonyms, using Pointwise Mutual Information as written in Equation (1) below:

$$\text{score}(choice_i) = \text{p}(problem \,\&\, choice_i) \,/\, \text{p}(choice_i) \quad (1)$$

where, $\{choice_1, choice_2, \dots, choice_n\}$ represent the alternatives from problem word $problem$, while probability that problem and $choice_i$ co-occur stated with $\text{p}(problem \,\&\, choice_i)$. Another variation of this equation is based on the closeness of the pair in the document, considering antonyms, and considering the context.

Through the principle of probability PMI-IR, (Islam & Inkpen, 2006) formulate the Pointwise Mutual Information that can be shown by Equation (2) as follows:

$$f^{pmi}(t_i, W) = \log_2 \frac{f^b(t_i,W)\times m}{f^t(t_i) f^t(W)} \quad (2)$$

$W$ is targeted word, while $f^t(t_i)$ is a type of frequency function, and $f^b(t_i, W)$ is a bigram frequency function. Then, the total number of tokens in corpus $C$ represented by $m$. Furthermore, $\beta - PMI\ summation$ functions for $W_1$ and $W_2$ are defined in Equation (3) and (4):

$$f^{\beta}(W_1) = \sum_{i=1}^{\beta_1} (f^{pmi}(X_i, W_2))^{\gamma} \quad (3)$$

$$f^{\beta}(W_2) = \sum_{i=1}^{\beta_2} (f^{pmi}(Y_i, W_1))^{\gamma} \quad (4)$$

Finally, the PMI semantic similarity function between the two words $W_1$ and $W_2$ is shown by the following Equations (5):

$$Sim(W_1, W_2) = \frac{f^{\beta}(W_1)}{\beta_1} + \frac{f^{\beta}(W_2)}{\beta_2} \quad (5)$$

The value $\beta$ is related to the number of times the word W appears in the corpus. The $\beta$ value is defined in the following Equation (6):

$$\beta_1 = (\log\ (f^t(W_i)))^2 \frac{(\log_2(n))}{\delta}, \quad i = 1,2 \quad (6)$$

Where $\delta$ is constant and in research conducted by (Islam & Inkpen, 2006) the value $\delta$ = 6.5 is determined. The value $\delta$ depends on the size of the corpus. The smaller the corpus used, the smaller the value of $\delta$.

## 3. Data

In this research, synonym similarity is obtained from two pairs of words with the semantic meaning approach. The data used is part of the TOEFL test which deals with synonyms. Data are collected from lessons 1-3 in the TOEFL exercise book written by (Matthiesen, 2017). A total of 30 multiple choice questions were used, and each question had four alternative answers. The following is an example of the data used:

Choose the synonym of *appealing*:

(A) refined
(B) encouraging
(C) alluring
(D) popular

The answer key provided by the book will provide information that the synonym of the word "appealing" is "alluring," in which the two words have the same meaning of the word "attractive." Besides, these words also have other synonyms, such as "interesting," "enticing," "catchy," and "catching."

### 4. Research flow

This research goes through the steps shown in Figure 1 below:

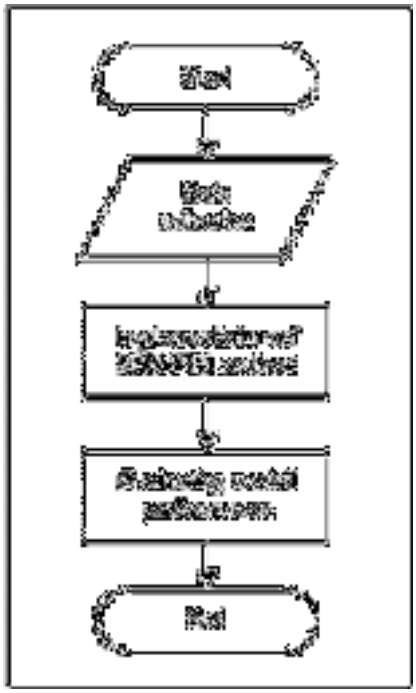


Figure 1. Research steps

The first step is to collect data, as described in the previous section. The data does not experience any pre-processing techniques. In this case, the similarity data will be directly measured using the SOC-PMI method. The library obtained from https://github.com/pritishyuvraj/SOC-PMI-Short-Text-Similarity- is used to measure the semantic similarity. In the library, there are at least three algorithms included, where the three methods are Hybrid methods named Semantic Text Similarity (STS) (Islam & Inkpen, 2008). In the library, there are at least three algorithms included, where the three algorithms are Hybrid methods named Semantic Text Similarity (STS). However, in this study, only the SOC-PMI algorithm was taken and used. This method includes the NLTK library and also uses WordNet as the dictionary. Wordnet is an extensive semantic network in which there are words and groups of words that are connected lexically and conceptually, which are represented by arc labeled (Fellbaum, 2006).

Furthermore, after the SOC-PMI value of each possible answer is obtained, an evaluation of the method's performance is carried out by finding a match between the two answers, both the predicted answer and the actual answer. We also apply another method for comparison. We use a character-based method called Longest Common Subsequence. We use this method because it is not possible to implement string-based methods with questions in the form of word synonyms, since the words used are clearly different.

## RESULTS AND DISCUSSION

The results and discussion of this paper can be seen as follows:

### 1. Results

The synonym question data in lessons 1, 2 and 3 collected and will measure the semantic similarity. Table 1 illustrates an example of the semantic similarity measurement results using the SOC-PMI method in question Number 5, Lesson 2:

Table 1. Word example using SOC-PMI method with synonym value

| *Question* | *Answer* | *SOC-PMI Score* |
|---|---|---|
| appealing | refined | 0.13333 |
| | encouraging | 0.14286 |
| | **alluring** | **0.84211** |
| | popular | 0.15385 |

Based on the results obtained in Table 1, it can be seen that the word "alluring" has the closest semantic relation to the word "appealing" with a similarity value of 0.84211. Furthermore, the word "alluring" in bold indicates that the word is the actual answer. In the end, the whole data is also measured, and the results obtained as shown in Table 2, Table 3, and Table 4.

Table 2. Results in lesson 1

| *No* | *Question* | *Answer* | *SOC-PMI Score* | *LCS Score* |
|---|---|---|---|---|
| 1 | widely | **broadly** | 0 | **0,46154** |
| | | abroad | 0 | 0,16667 |
| | | secretly | 0 | 0,42857 |
| | | truly | 0 | 0,36364 |
| 2 | autonomous | **independent** | 0 | 0,09524 |
| | | sudden | 0 | 0,25000 |
| | | international | 0 | 0,34783 |
| | | abrupt | 0 | **0,37500** |
| 3 | advice | acclaim | **0,66667** | 0,30769 |
| | | attention | 0,30769 | 0,26667 |
| | | **suggestion** | 0,30769 | 0,12500 |
| | | praise | **0,66667** | **0,50000** |
| 4 | attractive | **appealing** | 0 | 0,31579 |
| | | adverse | 0 | 0,35294 |
| | | arbitrary | 0 | **0,42105** |
| | | perfect | 0 | 0,35294 |
| 5 | disapproval | attraction | 0,13333 | **0,28571** |
| | | attention | 0,28571 | 0,20000 |
| | | **objection** | 0,28571 | 0,20000 |
| | | persistence | **0,375** | 0,18182 |
| 6 | haphazardly | suddenly | 0 | 0,31579 |
| | | secretly | 0 | 0,31579 |
| | | **carelessly** | 0 | **0,38095** |
| | | constantly | 0 | 0,28571 |
| 7 | constant | disruption | **0,35294** | 0,33333 |
| | | acceptable | 0 | 0,33333 |
| | | abrupt | 0 | 0,28571 |
| | | **persistent** | 0 | **0,44444** |
| 8 | perfect | attractive | 0 | **0,35294** |
| | | **ideal** | **0,26667** | 0,16667 |
| | | actual | 0 | 0,30769 |
| | | abrupt | 0 | 0,30769 |
| 9 | unfavorably | attractively | 0 | 0,43478 |
| | | haphazardly | 0 | 0,36364 |
| | | acceptably | 0 | 0,47619 |
| | | **adversely** | 0 | **0,50000** |
| 10 | disturbing | perfect | 0,16667 | 0,11765 |
| | | **disruptive** | **0,4** | **0,50000** |
| | | persistent | 0,4 | 0,40000 |
| | | attractive | 0,4 | 0,30000 |

Table 3. Results in lesson 2

| *No* | *Question* | *Answer* | *SOC-PMI Score* | *LCS Score* |
|---|---|---|---|---|
| 1 | inspire | celebrate | **0,28571** | 0,25000 |
| | | attract | 0,22222 | 0,14286 |
| | | **encourage** | 0,25 | **0,37500** |
| | | appeal | 0,14286 | 0,30769 |
| 2 | advantage | **benefit** | 0,26667 | 0,25000 |
| | | persistence | 0,4 | 0,20000 |
| | | nimbleness | 0,30769 | 0,21053 |
| | | allure | **0,66667** | **0,26667** |
| 3 | fragile | modern | 0 | 0,15385 |
| | | famous | 0 | 0,30769 |
| | | allowable | 0 | 0,37500 |
| | | **frail** | 0 | **0,83333** |
| 4 | contemporary | timing | 0,15385 | 0,22222 |
| | | **current** | **0,28571** | **0,31579** |
| | | well-known | 0 | 0,18182 |
| | | perfect | 0,14286 | 0,21053 |
| 5 | appealing | refined | 0,13333 | 0,37500 |
| | | encouraging | 0,14286 | 0,50000 |
| | | **alluring** | **0,84211** | **0,58824** |
| | | popular | 0,15385 | 0,37500 |
| 6 | renown | unknown | **0,14286** | **0,61538** |
| | | **celebrated** | 0 | 0,25000 |
| | | adverse | 0 | 0,30769 |
| | | disapprove | 0 | 0,25000 |
| 7 | worthwhile | **rewarding** | 0 | 0,31579 |
| | | acceptable | 0 | 0,30000 |
| | | agile | 0 | **0,40000** |
| | | permitted | 0 | 0,31579 |
| 8 | vigorous | attractive | 0 | 0,11111 |
| | | beautiful | 0 | 0,23529 |
| | | **energetic** | 0 | 0,11765 |
| | | advantageous | 0 | **0,50000** |
| 9 | refine | persist | **0,25** | 0,30769 |
| | | value | 0,13333 | 0,18182 |
| | | **perfect** | 0,15385 | **0,46154** |
| | | divide | 0,16667 | 0,40000 |
| 10 | distribute | disappoint | 0,25 | 0,50000 |
| | | disrupt | 0,22222 | **0,70588** |
| | | discourage | 0,22222 | 0,50000 |
| | | **dispense** | **1** | 0,44444 |

Table 4. Results in lesson 3

| *No* | *Question* | *Answer* | *SOC-PMI Score* | *LCS Score* |
|---|---|---|---|---|
| 1 | indispensable | abrupt | 0 | 0,21053 |
| | | abroad | 0 | 0,21053 |
| | | **vital** | 0 | **0,33333** |
| | | frail | 0 | 0,22222 |
| 2 | restore | appeal | 0,14286 | 0,15385 |
| | | **revitalize** | **0,22222** | **0,47059** |
| | | attract | **0,22222** | 0,28571 |
| | | disrupt | 0,2 | 0,28571 |
| 3 | conform | annoy | **0,4** | 0,33333 |
| | | divide | 0,2 | 0,00000 |
| | | encourage | **0,4** | **0,37500** |
| | | **adapt** | **0,4** | 0,00000 |
| 4 | notice | **observe** | 0 | 0,30769 |
| | | refine | 0 | 0,33333 |

| | | | | |
|---|---|---|---|---|
| | | distribute | 0 | **0,37500** |
| | | analyze | 0 | 0,30769 |
| 5 | current | energetic | 0 | **0,37500** |
| | | ideal | **0,13333** | 0,16667 |
| | | **ongoing** | 0 | 0,14286 |
| | | intense | 0 | 0,28571 |
| 6 | observe | alter | **0,33333** | **0,33333** |
| | | **notice** | 0,16667 | 0,30769 |
| | | anticipate | 0,25 | 0,11765 |
| | | modify | 0,28571 | 0,15385 |
| 7 | intense | **strong** | 0 | 0,30769 |
| | | intolerant | 0 | **0,58824** |
| | | vitally | 0 | 0,28571 |
| | | allowable | 0 | 0,12500 |
| 8 | enrich | alter | **0,33333** | 0,36364 |
| | | dispense | 0,25 | 0,28571 |
| | | disrupt | 0,22222 | 0,15385 |
| | | **enhance** | 0,2 | **0,46154** |
| 9 | unbearable | inspiring | 0 | 0,21053 |
| | | unfavorable | 0 | **0,76190** |
| | | **intolerable** | **1** | 0,66667 |
| | | ancient | 0 | 0,23529 |
| 10 | proposal | question | 0,28571 | **0,12500** |
| | | attention | **0,33333** | 0,11765 |
| | | benefit | 0,28571 | 0,00000 |
| | | **suggestion** | **0,33333** | 0,11111 |

Based on the results shown in Table 2, Table 3, and Table 4, several things must be considered. First, some vocabularies do not show any semantic relations. Figure 2 illustrates the distribution of words that have semantic relations and those without semantic relations.

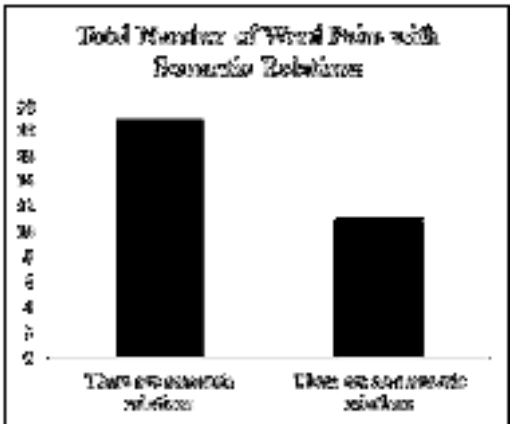


Figure 2. Total number of word pairs with semantic relations

In the end, the accuracy of the values generated by the SOC-PMI method with the actual answers is also measured. By reviewing the SOC-PMI values generated in Table 2, Table 3, and Table 4, it can be seen that there are 9 correct answers and 21 missed answers. Futhermore, the result for accuracy values are:

$$Accuracy = \frac{9}{30} \times 100\% = 30\%$$

Meanwhile, using the LCS method we obtain the following accuracy result:

$$Accuracy = \frac{13}{30} \times 100\% = 43.33\%$$

Unfortunately, it can be seen that the results are not satisfying results. The discussion section will explain the phenomena and analyze what factors influence the results and how this method can be used in the future.

## 2. Discussion

In this section, the results obtained are then analyzed. First, it should be noted that the SOC-PMI method is not evaluating semantic similarities based on synonymous rules. The SOC-PMI method considers the semantic relations between one pair of words, where semantic meaning can be anything. They can be synonymous, antonym, hyponymy, hypernimic, polysemic, or just connected to a certain hierarchy. Furthermore, this method takes into account how a pair of words meet in the same context. At this point, the frequency with which each word appears in the same context window greatly influences the results of semantic similarity. For example, the word pairs "computer" and "machine" will have more similar semantic relations (i.e. 0.94118) than "computer" and “keyboard" (0.82353), "computer" and "portable" (0.76190 ), and "computer" and "RAM" (0.70000).

High or low semantic similarity value is determined by the frequency of occurrence of the two words together in the context window. Even though the completeness of the word dictionary will also affect the results of semantic similarity. In the previous section, there were eleven questions for which the alternative answers did not have any semantic relations. This can happen for two reasons. First, the two words do not appear at all in the corpus, or it can only appear one of them without being followed by the next word. Secondly, the two words do exist in the corpus but do not appear in the same context. Therefore, the SOC-PMI similarity value cannot be obtained. In the case of *Netherlands - Holland*, *computer - keyboard*, *computer - machine*, or *mommy - daddy*, the SOC-PMI method might be able to provide competitive results, depending on the size of the corpus used. However, if the corpus is not able to represent words that are not commonly used, that will be another problem.

Here, the LCS method may have better performance. However, the LCS method ignores the semantic meaning of words because it considers the presence or absence of a character in the two words being compared. Sometimes the LCS method can give a higher score even though the characters are reversed. In this case, the idea of combining character-based and semantic-based methods can be considered in longer words, i.e., phrases or sentences. In the end, the hybrid method can be considered to produce a fairer similarity score. For example, we can give each word score weighting for the SOC-PMI and LCS values. The word "restore" has the synonym word "revitalize" where the SOC-PMI method gives a score of 0.22, and the LCS method gives a score of 0.47. If we give each method a weight of 0.5, we will get a final similarity score of 0.35.

For future NLP works involving word similarity factors, it can be concluded that the SOC-PMI method is not specifically recommended for personal use. As in this case, it is used to determine the synonymity of words. It would be wise to use the SOC-PMI method together with other methods (so that it will become a new hybrid method). This idea starts from the perspective that the relation of semantic meaning can be in any form. Thus, considering the possibility of syntactic similarity will be wiser and more objective on tasks involving more general similarities.

## CONCLUSIONS AND FUTURE WORKS

Finally, a conclusion can be drawn from the results and discussion in the previous section. Firstly, the SOC-PMI method is not suitable for determining specific semantic meanings, such as the case of synonyms between words. Because semantic relations can be of any type, depending on the frequency with which the two words occur together in the context window in the corpus. Secondly, the SOC-PMI method might perform well when measuring the semantic similarity of commonly used words, but this does not necessarily apply to TOEFL synonym questions because they have vocabulary lists that are sometimes "not common." In the end, the SOC-PMI method might work better when combined with other methods. However, the ability of WordNet will be a new challenge for other types of languages.

## ACKNOWLEDGEMENT

The authors would like to thank Universitas Amikom Purwokerto for funding this research.

## REFERENCE

Alguliyev, R. M., Aliguliyev, R. M., Isazade, N. R., Abdi, A., & Idris, N. (2017). A model for text summarization. *International Journal of Intelligent Information Technologies (IJIIT)*, *13*(1), 67–85.

Aronson, A. R., & Rindflesch, T. C. (1997). Query expansion using the UMLS Metathesaurus. In *Proceedings of the AMIA Annual Fall Symposium* (p. 485).

Barzilay, R., & Elhadad, M. (1999). Using lexical chains for text summarization. *Advances in Automatic Text Summarization*, 111–121.

Buckley, C., Salton, G., Allan, J., & Singhal, A. (1995). Automatic query expansion using SMART: TREC 3. *NIST Special Publication Sp*, 69.

Díaz-Galiano, M. C., Martín-Valdivia, M. T., & Ureña-López, L. A. (2009). Query expansion with a medical ontology to improve a multimodal information retrieval system. *Computers in Biology and Medicine*, *39*(4), 396–403.

Djajasudarma, T. F. (1993). *Semantik I: Pengantar ke Arah Ilmu Makna*. *Eresco 145*. Bandung.

Fellbaum, C. (2006). WordNet(s). In *Encyclopedia of Language & Linguistics (Second Edition)* (pp. 665–670).

Ferrucci, D. A. (2012). Introduction to "This is Watson." *IBM Journal of Research and Development*, *56*(3.4), 1.

Frawley, W. (2013). *Linguistic semantics*. Routledge.

Grefenstette, G. (1993). Automatic Thesaurus Generation from Raw Text using Knowledge-Poor Techniques. In *Ninth Annual Conference of the UW Centre for the New OED and Text Research*.

Islam, A., & Inkpen, D. (2006). Second order co-occurrence PMI for determining the semantic similarity of words. In *LREC* (pp. 1033–1038). https://doi.org/10.1145/1376815.1376819

Islam, A., & Inkpen, D. (2008). Semantic Text Similarity Using Corpus-Based Word Similarity and String Similarity. *ACM Transactions on Knowledge Discovery from Data*, *2*(2), 1–25. https://doi.org/10.1145/1376815.1376819

Lesk, M. (1986). Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In *Proceedings of the 5th annual international conference on Systems documentation* (pp. 24–26).

Li, H., Abe, N., World, R., & Partnership, C. (1998). Word clustering and disambiguation based on co-occurrence data. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics* (pp. 749–755).

Lin, C., Hovy, E., & Rey, M. (2003). Automatic Evaluation of Summaries Using N-gram Co-Occurrence Statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 71–78).

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *COLING 1998 Volume 2: The 17th International Conference on Computational Linguistics* (pp. 768–774).

Matthiesen, S. J. (2017). *Essential Words for the TOEFL*. Simon and Schuster.

Parera, J. D. (2004). Teori Semantik [Semantic Theory]. *Jakarta: Erlangga*.

Plovnick, R. M., & Zeng, Q. T. (2004). Reformulation of consumer health queries with professional terminology: a pilot study. *Journal of Medical Internet Research*, *6*(3), e27.

Turney, P. D. (2001). Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001)* (pp. 491–502).

Ullmann, S. (1964). *Language and style: collected papers* (Vol. 1). B. Blackwell.

Vechtomova, O., & Robertson, S. (2014). Integration of Collocation Statistics into the Probabilistic Retrieval Model. In *22nd Annual Colloquium on Information Retrieval Research* (pp. 165–177).

Xu, J., & Croft, W. B. (2000). Improving the Effectiveness of Information Retrieval with Local Context Analysis. *ACM Transactions on Information Systems (TOIS)*, *18*(1), 79–112.

Yarowsky, D. (1992). Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2* (pp. 454–460).

Terbit *online* pada laman web jurnal :
http://ejournal.amikompurwokerto.ac.id/index.php/telematika/

**Telematika**

Terakreditasi Sinta "3" KEMENRISTEKDIKTI, No. 21/E/KPT/2018

UNIVERSITAS AMIKOM PURWOKERTO

# Penerapan *Naïve Bayes* pada Pendeteksian *Malware* dengan Diskritisasi Variabel

**Inda Anggraini[1], Yesi Novaria Kunang[2], dan Firdaus[3]**

[1,2,] Magister Teknik Informatika, Universitas Bina Darma
[3] Magister Teknik Sipil, Universitas Bina Darma
Email : indaanggraini@gmail.com[1], yesinovariakunang@binadarma.ac.id[2], firdaus.dr@binadarma.ac.id[3]

**INFO ARTIKEL**

***Sejarah Artikel:***
Menerima 14 Agustus 2019
Revisi 7 Oktober 2019
Diterima 20 Februari 2020
*Online* 28 Februari 2020

**Keywords:**
*Malware Detection*
*Naïve Bayes*
*Discretization*
*Data Mining*

## *ABSTRACT*

*Malicious software (malware) is rogue software specifically designed to carry out malicious or destructive software activities on computers such as viruses, Trojans, and others that are spread through the internet network. The number of activities that spread malware that occurs through the internet network makes many users uneasy one form of the attack is to insert malicious or malicious files into the computer. For example, such as the web shell scripting script that is inserted into the internet service provider computer. This study aims to analyze malware attacks using the Naïve Bayes Classifier Algorithm with the discretization of 3-interval and 5-interval Min-Max variables for continuous attributes. Discretization (discretion) attribute is a technique for changing a function or continuous value into a discrete form. This technique is done as an adjustment to the possibility of the emergence of continuous values in a very small dataset feature. Discretization of variables is done in a dataset of type continuous so that the probability value indicates the possibility of the same value coming out of a class. Using the Naive Bayes algorithm is expected to help facilitate users in finding the right method for detecting attacks from malware. The experimental results show that the application of Naïve Bayes in the classification of data that has not gone through the discretion stage produces an accuracy of 69.72% with the prediction of malware 63.53 % while the data that has passed the discretization stage can provide accuracy of up to 79.97 % with 81.29 % malware prediction. The use of the Naïve Bayes by the binning method in this study has an increased detection ability compared to the classification process without using the binning process (discretization). The discretion process can make the Naïve Bayes algorithm more accurate in detecting malware.*

**Kata Kunci:**
Pendeteksian *Malware*
*Naïve Bayes*
Diskritisasi
*Data Mining*

**Korespondensi:**
Telepon: +62 81379517789
Email:
indaanggraini@gmail.com

## ABSTRAK

Malicious software (malware) adalah software jahat yang dirancang khusus untuk melakukan aktifitas berbahaya atau perusak perangkat lunak pada komputer seperti virus, Trojan, dan lain-lain yang disebar melalui jaringan internet. Banyaknya aktifitas penyebaran malware yang terjadi melalui jaringan internet membuat banyak pengguna menjadi resah salah satu bentuk dari serangan tersebut yaitu dengan melakukan penyisipan file-file berbahaya atau malicious ke komputer. Contohnya seperti penyisispan skrip web shell yang di sisipkan ke komputer penyedia layanan internet. Penelitian ini bertujuan untuk melakukan analisa terhadap serangan malware dengan menggunakan Algoritme *Naïve Bayes* Clasiffier dengan diskritisasi variabel Min-Max diskritisasi 3-interval dan 5-interval untuk atribut kontinu. Discretization (pendiskritan) atribut merupakan teknik untuk merubah sebuah fungsi atau nilai kontinu kedalam bentuk diskrit. Teknik ini dilakukan sebagai penyesuaian terhadap kemungkinan kemunculan nilai kontinu dalam fitur dataset yang sangat kecil. Pendiskritisasian variabel dilakukan pada dataset yang bertipe kontinu, sehingga nilai probabilitas menunjukan kemungkinan nilai yang sama keluar pada suatu kelas. Dengan menggunakan Algoritme naive bayes ini diharapkan dapat membantu mempermudah pengguna dalam menemukan metode yang tepat untuk

mendeteksi serangan dari malware. Hasil percobaan menunjukan bahwa penerapan *Naïve Bayes* pada klasifikasi data yang belum melalui tahap pendiskritan menghasilkan tingkat akurasi sebesar 69.72 % dengan prediksi malware 63.53 % sedangkan pada data yang telah melewati tahap diskritisasi mampu memberikan akurasi hingga 79.97 % dengan prediksi malware 81.29 %. Penggunaan metode *Naïve Bayes* dalam penelitian ini memiliki kemampuan deteksi yang meningkat dibandingkan dengan proses klasifikasi tanpa menggunakan proses binning (diskritisasi). Proses pendiskritan dapat menjadikan Algoritme *Naïve Bayes* menjadi lebih akurat di dalam mendeteksi malware.

## PENDAHULUAN

Perkembangan teknologi yang semakin pesat terutama teknologi komputer khususnya di bidang jaringan selain memberikan kemudahan, juga memberikan masalah di sisi keamanan dari komputer yang terintegrasi. Permasalahan keamanan komputer yang paling banyak dijumpai adalah penyebaran *malware* (*malicious software*) melalui jaringan *internet* yang menyebabkan berbagai macam kerugian (Setiawan dkk., 2017).

*Malware* sendiri merupakan perangkat lunak yang secara khusus dirancang untuk melakukan aktifitas berbahaya yang bisa merusak perangkat lunak lain. Contoh *malware* seperti Virus, *Trojan, Spyware* dan *Exploit* yang dibuat khusus agar tersembunyi sehingga mereka bisa tetap berada di dalam sistem komputer pada periode waktu tertentu tanpa sepengetahuan pemilik sistem (Cahyanto dkk., 2017). Beberapa *Malware* diciptakan dengan tujuan memata-matai seseorang, melakukan aktifitas merugikan seperti pencurian data dan informasi pribadi, membobol keamanan program dan sistem operasi serta banyak lagi. Untuk membobol suatu perangkat lunak atau sistem operasi dilakukan dengan menggunakan *script* yang diselipkan secara tersembunyi oleh penyerang (Sandag dkk., 2018).

Dengan banyaknya aktifitas penyebaran *malware* yang terjadi melalui jaringan *internet* membuat banyak pengguna menjadi resah. Untuk itu perlu melakukan pendeteksian terhadap serangan *malware* khususnya di jaringan agar pengguna bisa mengetahui apakah data yang berasal dari *internet* aman dari penyisipan malware atau tidak (Akbi & Rosyadi, 2018). Beberapa peneliti menggunakan pendekatan pembelajaran mesin seperti *k-nearest neighbor* (kNN) (Sandag dkk., 2018), Support Vector Machine (SVM) (Herlambang & Basuki, 2019) dan juga metoda klastering (Akbi & Rosyadi, 2018). Penelitian-penelitian tersebut sebagian besar masih belum mencapai nilai akurasi yang maksimal (Herlambang & Basuki, 2019).

Penelitian lain dilakukan oleh (Wirawan & Eksistyanto, 2015) menggunakan pendekatan *Naïve Bayes* pada sistem pendeteksi serangan atau *Intrusion Detection System* dengan menggunakan teknik diskritisasi Variabel. Hasil penelitian ini memperlihatkan penggunaan teknik binning mampu meningkatkan pendeteksian secara signifikan jika dibandingkan dengan proses klasifikasi tanpa menggunakan teknik binning. Dengan Teknik *binning* (pendiskritisasian) menjadikan probabilitas dari Algoritme *Naïve Bayes* bisa lebih dihandalkan dalam penentuan kelas.

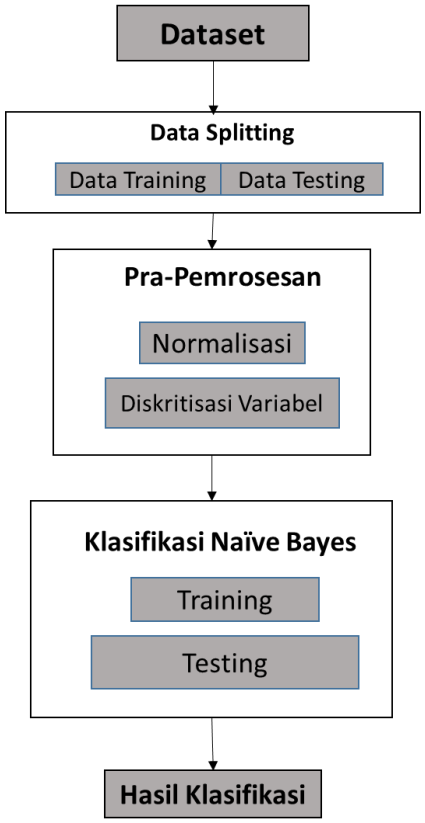
Algoritme *Naïve Bayes* merupakan Algoritme data *mining* yang relatif sederhana yang memiliki kelebihan tingkat akurasi yang tinggi dan dapat menangani data dalam jumlah besar (Huaturuk et al., 2018). Untuk itu dalam penelitian ini akan menerapkan metode *Naïve Bayes Classifier* dalam mendeteksi *malware* dengan teknik diskritisasi variabel. Pendiskritan ini dilakukan sebagai penyesuaian terhadap kemungkinan kemunculan nilai kontinu di dalam fitur dataset yang akan mempengaruhi hasil proses

klasifikasi. Untuk mengatasi hal tersebut dilakukan pendekatan teknik diskritisasi dengan menggunakan mean/standar deviasi.

## METODE PENELITIAN

### 1. Desain Penelitian

Dalam penelitian ini peneliti menggunakan metode Algoritme *Naïve Bayes Clasiffier* dalam mendeteksi *malware.* Alur tahapan penelitian untuk pendeteksian *malware* bisa dilihat pada Gambar 1. yang diproses menggunakan tool *RapidMiner*.




Gambar 1. Desain Penelitian

Adapun prosesnya sebagai berikut: (1) Tahapan dimulai dengan membaca dataset berupa data dengan format data file csv.; (2) Dataset dilakukan proses *splitting* (pembagian) menjadi data Training dan testing.; (3) Sebelum dilakukan proses klasifikasi/ pendeteksian dilakukan pra pemrosesan data. Proses ini sangat penting dan akan sangat berpengaruh pada hasil pendeteksian dan lamanya waktu pemrosesan. Pada tahap pra-pemrosesan, ada dua tahapan yang dilakukan yaitu normalisasi data dan diskritisasi variabel. Normalisasi data dilakukan dengan tujuan mengurangi adanya kesalahan pada proses pembacaan data. Proses pendiskritan dalam *dataset* dilakukan untuk penyesuaian terhadap kemungkinan munculnya nilai kontinu dalam karakteristik *dataset* yang kecil sehingga akan membawa pengaruh dalam proses klasifikasi dengan metode *Naïve Bayes*.; (4) Tahap terakhir dilakukan proses training dengan Algoritme *Naïve Bayes*. Model dilatih untuk mengenali data *malware* dan *benign* menggunakan data Training. Kemudian model pendeteksi *malware* akan dites dengan data testing untuk mengidentifikasi *malware*.

## 2. Dataset

Penelitian ini menggunakan dataset dengan tipe file *CSV (Comma Separated Values)* yang berextensi file excel. *CSV (Comma Separated Values)* merupakan suatu format data dalam basis data dimana setiap record dipisahkan dengan tanda koma (,) atau titik koma (;). Data yang digunakan adalah data sekunder dari dataset *malware* yang diambil dari *website* kaggle milik saravana (Saravana, 2018). Jumlah dataset *malware* yang digunakan peneliti yaitu 100.000 data dengan 34 attribut seperti pada Tabel 1.

Tabel 1. Tipe Data Attribute Dataset *Malware*

| *No* | *Attribut* | *Tipe Data* |
|---|---|---|
| 1 | *Millisecond* | *Numeric* |
| 2 | *Classification* | *String* |
| 3 | *State* | *Numeric* |
| 4 | *Usage_counter* | *Numeric* |
| 5 | *Prio* | *Numeric* |
| 6 | *Static_prio* | *Numeric* |
| 7 | *Normal_Prio* | *Numeric* |
| 8 | *Policy* | *Numeric* |
| 9 | *Vm_pgoff* | *Numeric* |
| 10 | *Vm_truncate_count* | *Numeric* |
| 11 | *Task_size* | *Numeric* |
| 12 | *Cached_hole_size* | *Numeric* |
| 13 | *Free_area_cache* | *Numeric* |
| 14 | *Mm_user* | *Numeric* |
| 15 | *Map_count* | *Numeric* |
| 16 | *Hiwater_rss* | *Numeric* |
| 17 | *Total_vm* | *Numeric* |
| 18 | *Shared_vm* | *Numeric* |
| 19 | *Exec_vm* | *Numeric* |
| 20 | *Reserved_vm* | *Numeric* |
| 21 | *Nr_ptes* | *Numeric* |
| 22 | *End_data* | *Numeric* |
| 23 | *Last_interval* | *Numeric* |
| 24 | *Nvcsw* | *Numeric* |
| 25 | *Nivcsw* | *Numeric* |
| 26 | *Min_flt* | *Numeric* |
| 27 | *Maj_flt* | *Numeric* |
| 28 | *Fs_excl_counter* | *Numeric* |
| 29 | *Lock* | *Numeric* |
| 30 | *Utime* | *Numeric* |
| 31 | *Stime* | *Numeric* |
| 32 | *Gtime* | *Numeric* |
| 33 | *Cgtime* | *Numeric* |
| 34 | *Signal_nvcsw* | *Numeric* |

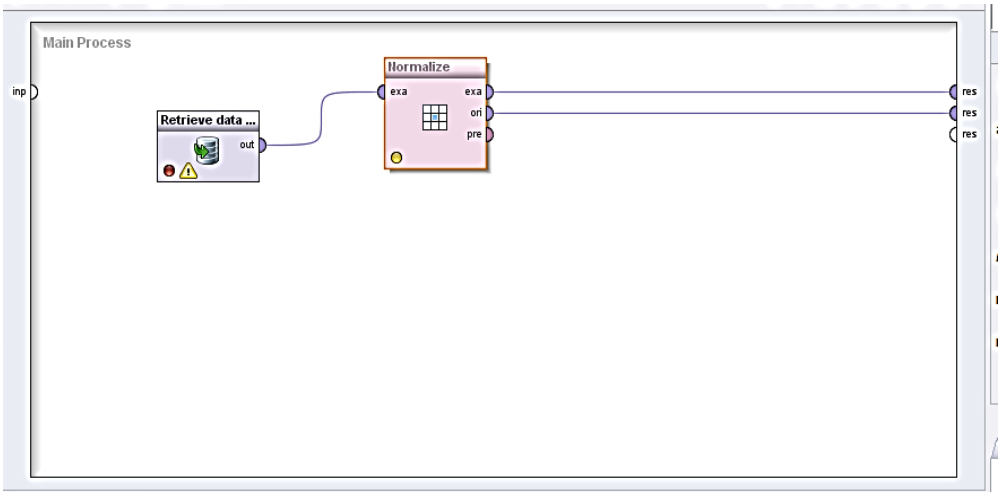
Tabel 2. Contoh isi Dataset *Malware*

| *Milli second* | *Classi Fication* | *State* | *Usage _counter* | *prio* | *static_prio* | *Normal _prio* | *policy* | *vm_pgoff* | *vm_ truncate_count* |
|---|---|---|---|---|---|---|---|---|---|
| 0 | *Malware* | 0 | 0 | 3069378560 | 14274 | 0 | 0 | 0 | 13173 |
| 1 | *Malware* | 0 | 0 | 3069378560 | 14274 | 0 | 0 | 0 | 13173 |
| 2 | *Malware* | 0 | 0 | 3069378560 | 14274 | 0 | 0 | 0 | 13173 |
| 3 | *Malware* | 0 | 0 | 3069378560 | 14274 | 0 | 0 | 0 | 13173 |
| 4 | *Malware* | 0 | 0 | 3069378560 | 14274 | 0 | 0 | 0 | 13173 |
| 5 | *Malware* | 0 | 0 | 3069378560 | 14274 | 0 | 0 | 0 | 13173 |
| 6 | *Malware* | 0 | 0 | 3069378560 | 14274 | 0 | 0 | 0 | 13173 |
| 7 | *Benign* | 319488 | 0 | 3069378560 | 23404 | 0 | 0 | 0 | 14856 |
| 8 | *Benign* | 319488 | 0 | 3069378560 | 23404 | 0 | 0 | 0 | 14856 |
| 9 | *Benign* | 319488 | 0 | 3069378560 | 23404 | 0 | 0 | 0 | 14856 |
| 10 | *Benign* | 319488 | 0 | 3069378560 | 23404 | 0 | 0 | 0 | 14856 |

Contoh isi dari tabel dataset bisa dilihat pada Tabel 2. yang memperlihatkan 10 baris pertama dan 10 kolom pertama dari dataset malware yang digunakan.

### 3. Normalisasi Data

Tahap normalisasi merupakan tahapan pra-pemrosesan yang dilakukan sebagai penyesuaian terhadap kemunculan nilai kontinu dalam fitur dataset yang akan mempengaruhi hasil proses klasifikasi dengan *Naïve Bayes*. Proses dilakukan dengan membaca dataset *malware* yang telah diinputkan sebelumnya ke dalam *tool* RapidMiner, kemudian mulai dilakukan tahap normalisasi pada Gambar 2. Proses normalisasi disini dimaksudkan untuk merubah jenis skala pengukuran yang dari data numerik. Proses normalisasi dilakukan dengan fungsi *Min-Max Normalize* yang akan mentransformasi data dengan rentang nilai (0,0) dan (0,1) seperti pada Tabel 3.



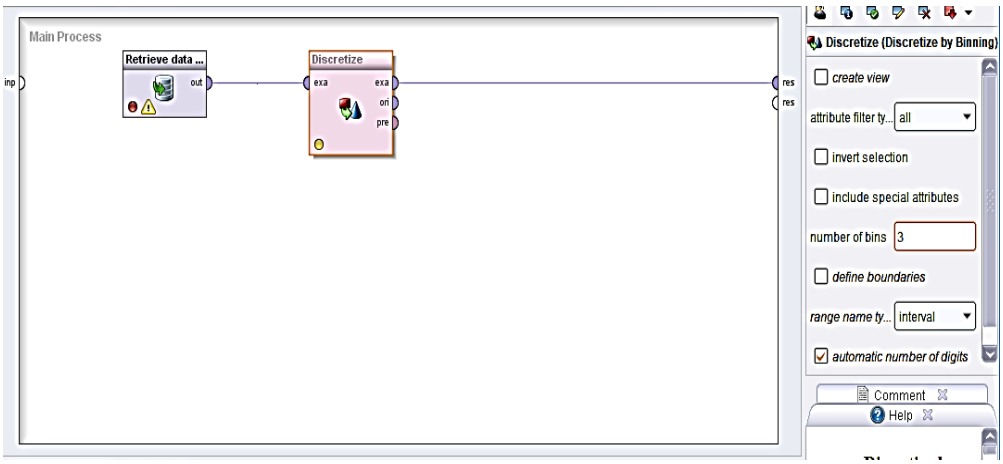

Gambar 2. Proses Normalisasi Data

Tabel 3. Hasil Normalisasi Data

| *Classification* | *Milli Second* | *State* | *Usage_counter* | *prio* | *static_prio* | *Normal_prio* | *policy* | *vm_pgoff* | *vm_truncate_count* |
|---|---|---|---|---|---|---|---|---|---|
| *malware* | 0 | 0 | 0 | 0.183 | 0.016 | 0 | 0 | 0 | 0.199 |
| *malware* | 0.001 | 0 | 0 | 0.183 | 0.016 | 0 | 0 | 0 | 0.199 |
| *malware* | 0.002 | 0 | 0 | 0.183 | 0.016 | 0 | 0 | 0 | 0.199 |
| *malware* | 0.003 | 0 | 0 | 0.183 | 0.016 | 0 | 0 | 0 | 0.199 |
| *malware* | 0.004 | 0 | 0 | 0.183 | 0.016 | 0 | 0 | 0 | 0.199 |
| *benign* | 0 | 0 | 0 | 0.206 | 0.138 | 0 | 0 | 0 | 0.289 |
| *benign* | 0.001 | 0 | 0 | 0.206 | 0.138 | 0 | 0 | 0 | 0.289 |
| *benign* | 0.002 | 0 | 0 | 0.206 | 0.138 | 0 | 0 | 0 | 0.289 |
| *benign* | 0.001 | 0 | 0 | 0.206 | 0.138 | 0 | 0 | 0 | 0.289 |
| *benign* | 0.002 | 0 | 0 | 0.206 | 0.138 | 0 | 0 | 0 | 0.289 |

### 4. Diskritisasi Data

Tahap pra-pemrosesan berikutnya adalah diskritisasi data. Proses diskritisasi ini dilakukan sebagai penyesuaian terhadap kemungkinan munculnya nilai kontinu dalam fitur dataset yang dapat mempengaruhi hasil proses pengklasifikasian dengan menggunakan metode *Naïve Bayes*. Dataset yang telah melalui tahap normalisasi dengan menggunakan metode Min-Max, dilakukan proses diskritisasi dengan teknik *binning*. Dalam penelitian ini dievaluasi proses *binning* kedalam 3 bagian atau interval yaitu (-1,0,1) dan 5 interval (-2, -1, 0, 1, 2). Proses diskritisasi dari data hasil normalisasi pada *tool* RapidMiner bisa dilihat pada Gambar 3.




Gambar 3. *Flow Diagram* Proses Diskritisasi Data

Setelah proses diskritisasi dijalankan maka akan menghasilkan data yang telah diproses ke dalam pendiskritan. Adapun variabel yang didiskritisasi yaitu variabel *millisecond, state, prio, static prio, vm_truncate_count, free_area_cache, mm_user, map_count, total_vm, shared_vm, exec_vm, reserved_vm, end_data, last interval, nvcsw, min_flt, maj_flt, fs_excl_counter, utime, stime, gtime.* Contoh hasil diskritisasi bisa dilihat pada Tabel 4. yang memperlihatkan hasil diskritisasi data 3 interval dan 5 interval. Diskritisasi data 3 interval akan mengelompokan data dalam 3 kelompok, sedangkan diskritisasi data 5 interval akan mengelompokkan data dalam 5 kelompok.

Tabel 4. Hasil Diskritisasi Data

| Label | Diskritisasi 3-interval | Diskritisasi 5-interval |
|---|---|---|
| *Millisecond* | (-∞-0.3), (0.3-0.7), (0.7-∞) | (-∞-0.2), (0.2-0.4), (0.4-0.6), (0.6-0.8), (0.8-∞) |
| *Prio* | (-∞-0.3), (0.3-0.7), (0.7-∞) | (-∞-0.2), (0.2-0.4), (0.4-0.6), (0.6-0.8), (0.8-∞) |
| *Static_Prio* | (-∞-0.3), (0.3-0.7), (0.7-∞) | (-∞-0.2), (0.2-0.4), (0.4-0.6), (0.6-0.8), (0.8-∞) |
| *Last_Interval* | (-∞-0.3), (0.3-0.7), (0.7-∞) | (-∞-0.2), (0.2-0.4), (0.4-0.6), (0.6-0.8), (0.8-∞) |
| *Map_Count* | (-∞-0.3), (0.3-0.7), (0.7-∞) | (-∞-0.2), (0.2-0.4), (0.4-0.6), (0.6-0.8), (0.8-∞) |
| *End_Data* | (-∞-0.3), (0.3-0.7), (0.7-∞) | (-∞-0.2), (0.2-0.4), (0.4-0.6), (0.6-0.8), (0.8-∞) |
| *STime* | (-∞-0.2), (0.2-0.5), (0.5-∞) | (-∞-0.2), (0.2-0.3), (0.3-0.4), (0.4-0.6), (0.6-∞) |

## 5. Metode Yang Digunakan

Metode penelitian yang digunakan dalam penelitian ini adalah Algoritme *Naïve Bayes*. Algoritme *Naïve Bayes* merupakan sebuah metoda klasifikasi menggunakan metode probabilitas dan statistik. *Naïve Bayes* merupakan sebuah pengklasifikasian probabilistik sederhana yang menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari dataset yang diberikan (Saleh, 2015). Algoritme mengunakan teorema Bayes mengasumsikan semua atribut independen tidak saling bergantung yang berdampak pada nilai variabel kelas. Algoritme *Naïve Bayes* memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai *Teorema Bayes* (Mustafa dkk., 2018). Persamaan dari metode *Naïve Bayes Classifier* bisa dilihat pada persamaan (1) dan (2).

$$P(H \mid X) = \frac{P(X \mid H).P(H)}{P(X)} \quad (1)$$

$P(H|X)$ yang dicari merupakan probabilitas hipotesis $H$ berdasar kondisi $X$ (posteriori probabilitas). Dimana $P(X|H)$ merupakan probabilitas $X$ berdasarkan kondisi pada hipotesis $H$; $X$

merupakan data dengan *class* yang belum diketahui; $H$ merupakan data hipotesis suatu *class* tertentu; $P(H)$ merupakan probabilitas hipotesis $H$ (prior probabilitas) dan $P(X)$ merupakan Probabilitas $X$.

Proses klasifikasi memerlukan sejumlah petunjuk untuk menentukan kelas apa yang cocok bagi sampel yang dianalisis tersebut. Karena itu, metode *Naive bayes* persamaan (1) disesuaikan menjadi persamaan (2).

$$P(C \mid F_1 \ldots F_n) = \frac{P(C).P(F_1 \ldots F_n \mid C)}{P(F_1 \ldots F_n)} \quad (2)$$

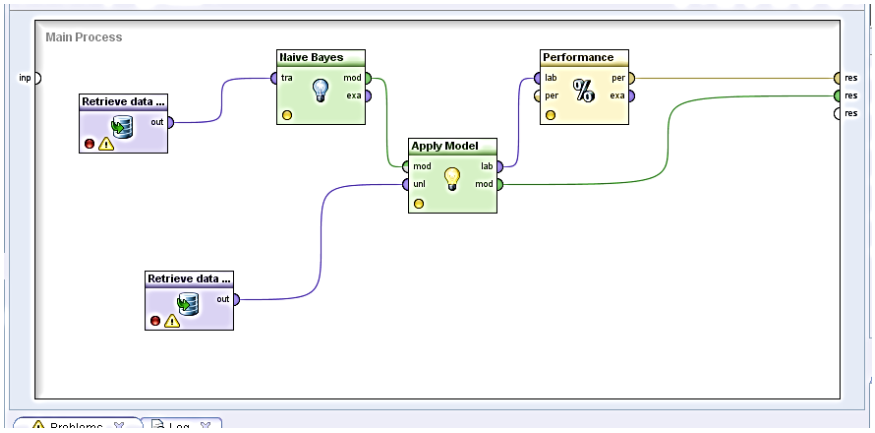
Di mana Variabel $C$ merepresentasikan kelas, sementara variabel $F_1 \ldots F_n$ merepresentasikan karakteristik petunjuk yang dibutuhkan untuk melakukan klasifikasi.

## HASIL DAN PEMBAHASAN

Pada penelitian ini dicoba 3 skenario yaitu proses (1) Pendetksian dengan normalisasi tanpa diskritisasi, (2) Pendeteksian dengan diskritisasi 3 interval, dan yang ke (3) Pendeteksian dengan diskritisasi 5 interval. Dataset dilakukan pembagian dengan 90% data menjadi data training dan 10% data digunakan untuk pengujian (testing).

### 1. Normalisasi Tanpa Diskritisasi

Pada proses normalisasi tanpa diskritisasi ini data yang telah melalui proses normalisasi pada gambar 2 akan menjadi input untuk *Naïve Bayes Classifier*. Pada Gambar 4. terlihat alur proses dari pendeteksian *malware*. Tahap pra-pemrosesan akan menghasilkan *retrieve data* yang sudah melalui proses normalisasi baik untuk data training dan data testing. Dari data training akan dilatih dengan menggunakan Algoritme *Naïve Bayes* untuk menghasilkan model. Selanjutnya model dilatih dengan data testing.




Gambar 4. *Flow Diagram* Klasifikasi Tanpa *Binning*

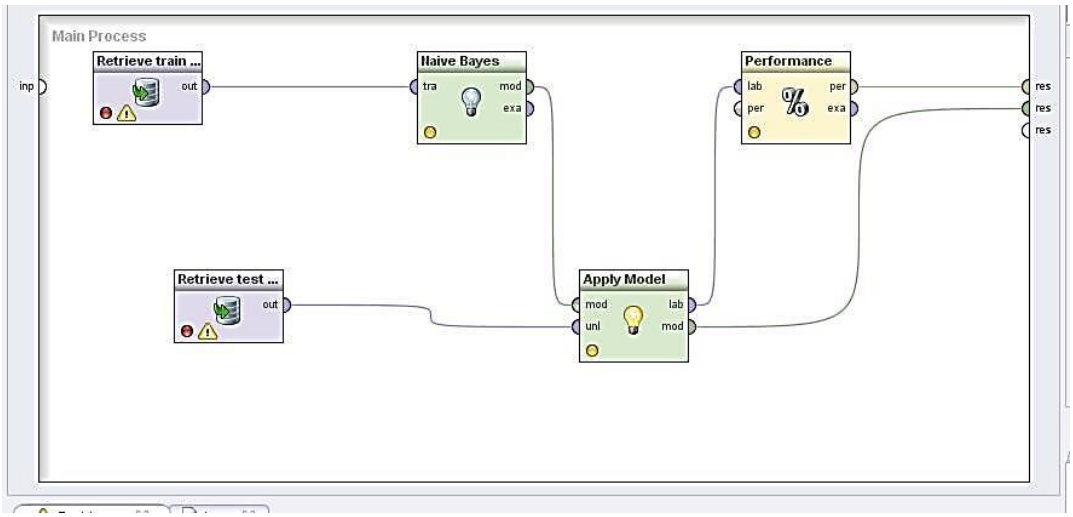
Hasil pengujian model yang pertama ini bisa dilihat pada Confusion Matrix pada Tabel 5. Dengan hanya menggunakan proses normalisasi tanpa diskritisasi pada proses pra-pemrosesan, Algoritme *Naïve Bayes Classifier* menghasilkan tingkat keakurasian sebesar 69,72%. Kemampuan pendeteksian dengan *dataset* ini masih belum optimal karena adanya atribut dengan nilai kontinu yang memiliki probabilitas kemunculan sangat kecil dalam data sehingga data itu tidak dapat diklasifikasikan dengan benar.

Tabel 5. Hasil *Confusion Matrix* Normalisasi tanpa Diskritisasi

| *Accuracy* : 69.72 % | | | |
|---|---|---|---|
| | *True malware* | *True benign* | *Class precision* |
| *Pred.malware* | 4707 | 2702 | 63,53 % |
| *Pred.benign* | 326 | 2265 | 87,42 % |
| *Class recall* | 93,52 % | 45,60 % | |

**2. Diskritisasi dengan 3-interval**

Pada pengujian model ke-2 ini dataset yang telah melalui tahap normalisasi dengan menggunakan metode Min-Max (Gambar 2), kemudian dilakukan proses diskritisasi dengan teknik *binning* kedalam 3 bagian atau interval (-1,0,1) (gambar 3). Proses klasifikasi akan membaca dataset training dan *testing* yang sudah melalui proses diskritisasi. Alur proses untuk klasifikasi dengan menggunakan teknik *binning* bisa dilihat pada Gambar 5. Jika dilihat pada tahapan hampir sama seperti proses pendeteksian tanpa teknik *binning*. Perbedaan proses *binning* ada pada data *retrieve* untuk data training dan testing yang sudah dikonversi menjadi data diskritisasi 3 variabel seperti pada Tabel 4.




Gambar 5. *Flow Diagram* Klasifikasi dengan Teknik Diskritisasi

Hasil *Confusion Matrix* pendeteksian *malware* dengan dataset yang telah dilakukan proses diskritisasi 3 interval bisa dilihat pada Tabel 6. Tingat keakurasian pendeteksian untuk data testing meningkat sebesar 78,16%. Peningkatan keakurasian pendeteksian dikarenakan kemunculan nilai kontinu dari dataset telah dihilangkan dengan teknik *binning*. Secara detail nilai presisi dan sensitivitas (*recall*) dari class *malware* dan *benign* bisa dilihat pada Tabel 6.

Tabel 6. Hasil *Confusion Matrik* untuk Diskritisasi 3-Interval

| *Accuracy* : 78,16 % | | | |
|---|---|---|---|
| | *True malware* | *True benign* | *Class precision* |
| *Pred.malware* | 4565 | 1716 | 72,68 % |
| *Pred.benign* | 468 | 3251 | 87,42 % |
| *Class recall* | 90,70 % | 65,45 % | |

### 3. Diskritisasi dengan 5-interval

Setelah proses normalisasi dan diskritisasi variabel 3 interval, model yang ketiga dataset yang telah dinormalisasi dilakukan teknik diskritisasi dengan *binning* kedalam bentuk 5 interval yaitu (-2,-1,0,1,2) seperti pada Tabel 4. Alur proses model pendeteksian hampir sama seperti pada teknik diskritisasi 3 variabel (Gambar 5). Hasil klasifikasi data *testing* bisa dilihat pada Tabel 7 memperlihatkan *Confusion Matrix* untuk model ke tiga.

Tabel 7. Hasil *Confusion Matrik* untuk Diskritisasi 5-interval

| Accuracy : 79.97 % | | | |
|---|---|---|---|
| | *True malware* | *True benign* | *Class precision* |
| *Pred.malware* | 3936 | 906 | 81,29 % |
| *Pred.benign* | 1097 | 4061 | 78,73 % |
| *Class recall* | 78,20 % | 81,76 % | |

Setelah proses diskritisasi 5 interval dengan teaknik *binning* dilakukan maka didapatkan hasil akurasi pendeteksian meningkat sebesar sebesar 79,97%. Demikian juga untuk nilai *Recall* (sensitivitas) dan presisi terutama untuk *class malware* hasilnya meningkat.

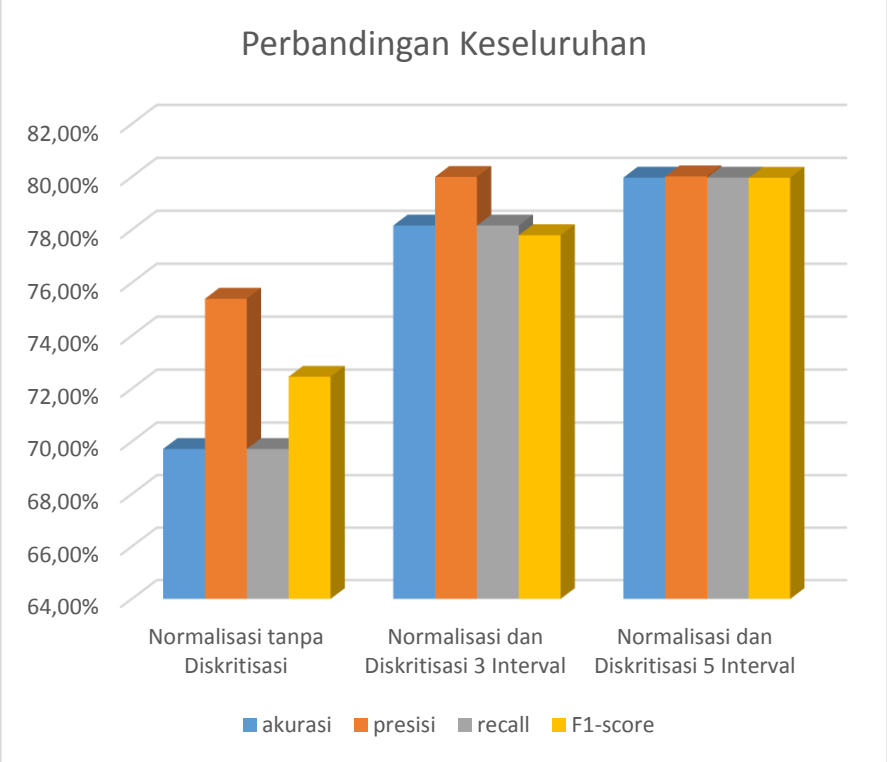
### 4. Perbandingan Hasil

Secara keseluruhan untuk nilai presisi, *recall* dan *F1-Score* dari ketiga model yang diuji bisa dilihat pada Tabel 8 dan Gambar 6. Pada tabel 8 terlihat bahwa tingkat persentase presisi keseluruhan yang dihasilkan dari dataset tanpa proses *binning* (pendiskritan) hanya sebesar 75,40% dan meningkat setelah didiskritisasi 5 interval menjadi sebesar 80.02%. Demikian juga untuk nilai Recall yang tanpa diskritisasi hanya sebesar 69,72% meningkat menjadi 78,16% pada hasil diskritisasi 3 interval dan sebesar 79,97% pada hasil diskritisasi 5 interval. Terutama untuk pengenalan *class malware* nilai presisi meningkat hingga 81,29%.

Tabel 8. Perbandingan hasil persentase dengan *Naïve Bayes Classifier*

| ***Class*** | **Tanpa Diskrit** | | | **Diskrit 3-interval** | | | **Diskrit 5-interval** | | |
|---|---|---|---|---|---|---|---|---|---|
| | ***Pre*** | ***Recall*** | ***F1 Score*** | ***Pre*** | ***Recall*** | ***F1 Score*** | ***Pre*** | ***Recall*** | ***F1 Score*** |
| *Malware* | 63,53% | 93,52% | 75,66% | 72,68% | 90,70% | 80,70% | 81,29% | 78,20% | 79,22% |
| *Benign* | 87,42% | 45,60% | 59,94% | 87,42% | 65,45% | 74,86% | 78,73% | 81,78% | 80,22% |
| *Overall* | 75,40% | 69,72% | 72,45% | 80,00% | 78,16% | 77,80% | 80,02% | 79,97% | 79,97% |

Untuk keseluruhan kinerja dari parameter yang dilihat maka pada Gambar 6 bisa dilihat untuk nilai akurasi, presisi keseluruhan, *recall* keseluruhan dan *F1-score* keseluruhan hasil diskritisasi 5 interval memperlihatkan hasil yang paling baik dibandingkan tanpa proses diskritisasi dan diskritisasi 3 interval.




Gambar 6. Hasil Evaluasi Keseluruhan

Hasil tersebut memperlihatkan dengan proses pra-pemrosesan dengan teknik diskritisasi dapat meningkatkan kinerja dari metode *Naïve Bayes Classifier* dibandingkan dengan penelitian sebelumnya yang menghasilkan persentase yang berbeda (Saravana, 2018). Proses diskritisasi ini sangat sesuai digunakan pada dataset *malware* yang atribut-atribut datanya sebagian besar merupakan data numerik, sehingga berpengaruh pada hasil pendeteksian.

## KESIMPULAN DAN SARAN

Berdasarkan hasil pendeteksian dengan menguji tiga model dapat disimpulkan bahwa penerapan metode *Naïve Bayes* dengan pra-pemrosesan yang menggunakan teknik diskritisasi bisa meningkatkan hasil keakurasian pendeteksian *malware* dibandingkan dengan proses klasifikasi tanpa menggunakan teknik *binning* (diskritisasi). Tahap pendiskritan ini dapat menjadikan Algoritme *Naïve Bayes* menjadi tampak akurat di dalam mendeteksi *malware*.

Untuk penelitian selanjutnya model perlu diuji dengan menggunakan dataset yang besar dan lebih komplek. Selain itu menarik untuk dievaluasi pengaruh penggunaan teknik diskritisasi untuk pra-pemrosesan untuk Algoritme *machine learning* dan *data mining* lainnya.

## DAFTAR PUSTAKA

Akbi, D. R., & Rosyadi, A. R. (2018). Analisis Klasterisasi Malware: Evaluasi Data Training Dalam Proses Klasifikasi Malware. *Jurnal ELTIKOM*, *2*(2), 58–66. https://doi.org/10.31961/eltikom.v2i2.88

Amalia, N., Shaufiah, S., & Sa'adah, S. (2015). Penerapan teknik data mining untuk klasifikasi ketepatan waktu lulus mahasiswa teknik informatika universitas telkom menggunakan Algoritme *naive bayes* classifier. *EProceedings of Engineering*, *2*(3).

Anam, C., & Santoso, H. B. (2018). Perbandingan kinerja Algoritme c4. 5 dan *naive bayes* untuk klasifikasi penerima beasiswa. *ENERGY*, *8*(1), 13–19.

Asih, T. S. N., Waluya, B., & Supriyono, S. (2018). Perbandingan finite difference method dan finite element method dalam mencari solusi persamaan diferensial parsial. *PRISMA, Prosiding Seminar Nasional Matematika*, *1*, 885–888.

Cahyanto, T. A., Wahanggara, V., & Ramadana, D. (2018). Analisis dan Deteksi Malware Menggunakan Metode Malware Analisis Dinamis dan Malware Analisis Statis. *JUSTINDO (Jurnal Sistem dan Teknologi Informasi Indonesia*, *2*(1), 12.

Haryati, S., Sudarsono, A., & Suryana, E. (2015). Implementasi data mining untuk memprediksi masa studi mahasiswa menggunakan Algoritme c4. 5 (studi kasus: Universitas dehasen bengkulu). *Jurnal Media Infotama*, *11*(2).

Herlambang, S., & Basuki, S. (2019). Deteksi Malware Android Berdasarkan System Call Menggunakan Algortima Support Vector Machine. *Prosiding SENTRA (Seminar Teknologi dan Rekayasa*, *4*, 157–165.

Huaturuk, N. R. S., Rahmadani, R. D., & Ak, D. J. (2018). Komparasi Akurasi *Naïve Bayes* dan Support Vector Machine (SVM) untuk Rekomendasi Produk in Fashion Dress. *Conference on Electrical Engineering, Telematics, Industrial technology, and Creative Media (CENTIVE)*, 168–173.

Mustafa, M. S., Ramadhan, M. R., & Thenata, A. P. (2018). Implementasi data mining untuk evaluasi kinerja akademik mahasiswa menggunakan Algoritme *naive bayes* classifier. *Creative Information Technology Journal*, *4*(2), 151–162.

Nasari, F., & Darma, S. (2015). Penerapan K-Means Clustering pada Data Penerimaan Mahasiswa Baru (Studi Kasus: Universitas Potensi Utama). *Seminar Nasional Teknologi Informasi dan Multimedia*, *3*, 73–78.

Novrianda, R., Kunang, Y. N., & Shaksono, P. H. (2014). Analisis Forensik Malware pada Platform Android. *KonferensiNasional Ilmu Komputer (KONIK)*, 377–385.

Nugroho, P. A. (2016). Penanganan dan Pendeteksian Penyebaran Malware Menggunakan X Ray PC Pada Sistem Operasi Windows XP (Studi Kasus Worm "MR. COOLFACE"). *Jurnal Inovasi Informatika*, *1*(2), 32–41.

Saleh, A. (2015). Implementasi metode klasifikasi *naive bayes* dalam memprediksi besarnya penggunaan listrik rumah tangga. *Creative Information Technology Journal*, *2*(3), 207–217.

Sandag, G. A., Leopold, J., & Ong, V. F. (2018). Klasifikasi *Malicious* Websites Menggunakan Algoritme K-NN Berdasarkan Application Layers dan Network Characteristics. *CogITo Smart Journal*, *4*(1), 37. https://doi.org/10.31154/cogito.v4i1.100.37-45

Saravana, N. (2018, April 12). *Malware Detection*. https://www.kaggle.com/nsaravana/malware-detection

Setiawan, F. G. N. D., Ijtihadie, R. M., & Studiawan, H. (2017). Pendeteksian Malware pada Lingkungan Aplikasi Web dengan Kategorisasi Dokumen. *Jurnal Teknik ITS*, *6*(1), 71–74. https://doi.org/10.12962/j23373539.v6i1.22163

Wirawan, I. N. T., & Eksistyanto, I. (2015). Penerapan *Naive bayes* pada Intrusion Detection System dengan Diskritisasi Variabel. *JUTI: Jurnal Ilmiah Teknologi Informasi*, *13*(2), 182. https://doi.org/10.12962/j24068535.v13i2.a487

Terbit *online* pada laman web jurnal :
http://ejournal.amikompurwokerto.ac.id/index.php/telematika/

**Telematika**

Terakreditasi Sinta "3" KEMENRISTEKDIKTI, No. 21/E/KPT/2018

UNIVERSITAS AMIKOM PURWOKERTO

# Desain dan Implementasi Penandatangan Elektronik Sertifikat X509 Menggunakan Platform Bot Telegram

**Herman Kabetta**

Program Studi Rekayasa Kriptografi
Sekolah Tinggi Sandi Negara
Email : herman.kabetta@stsn-nci.ac.id

**INFO ARTIKEL**

***Sejarah artikel:***
Menerima 6 Desember 2019
Revisi 29 Januari 2020
Diterima 24 Januari 2020
*Online* 28 Februari 2020

***Keyword*:**
*Digital Signature,*
*Digital Certificate,*
*X509 Certificate,*
*Chatbot,*
*Rapid Application Development,*
*System Usability Scale*

***ABSTRACT***

*Balai Sertifikasi Elektronik (BSrE) as one of the Certificate Authorities in Indonesia has been released a desktop-based digital signing application, but one of the weaknesses of desktop applications is its low portability. BSrE has also released a digital signing application for mobile operating systems, but it is only intended for users of the Android operating system. The aim of this research is to develop a digital signing application with Telegram Bot platform, that can be used to sign electronic documents using X509 certificates wherever and whenever, and also it can be run on all operating system platforms. The research methodology is using Rapid Application Development (RAD) which consists of four stages, Requirements Planning, User Design, Construction and Cutover. The backend system of the bot is built using Java programming language, and integrated with the MySQL database as conversation sessions storage. There are three main functions of the designed system, sign, verify and setting. Signed documents also have been tested in several pdf reader applications and digital signatures can be recognized and validated. Bot can also verify documents signed by other applications. Testing uses a black-box method, the results of functional testing and non-functional testing show the system can run properly as expected in requirements planning. Evaluation using System Usability Scale (SUS) indicate that the system is suitable for use.*

**Kata Kunci*:***
Tanda Tangan Elektronik,
Sertifikat Elektronik,
Sertifikat X509,
Chatbot,
Rapid Application Development,
System Usability Scale

**ABSTRAK**

Balai Sertifikasi Elektronik (BSrE) sebagai salah satu *Certificate Authority* di Indonesia telah merilis aplikasi penandatangan elektronik berbasis desktop kepada publik, namun salah satu kekurangan aplikasi desktop adalah rendahnya portabilitas dalam penggunaan. BSrE juga telah merilis aplikasi penandatangan elektronik untuk sistem operasi mobile, namun hanya diperuntukkan bagi pengguna sistem operasi Android. Penelitian ini bertujuan mengembangkan sebuah Bot Telegram yang dapat digunakan untuk menandatangani dokumen elektronik menggunakan sertifikat X509 dimanapun dan kapanpun, serta dapat berjalan pada semua platform sistem operasi. Metode yang digunakan dalam penelitian ini adalah Rapid Application Development (RAD) yang terdiri dari empat tahap, Requirements Planning, User Design, Construction dan Cutover. Sistem backend bot dibangun dengan menggunakan bahasa pemrograman Java yang terintegrasi dengan basis data MySQL untuk menyimpan sesi percakapan. Penelitian menghasilkan sebuah sistem Bot Telegram yang memiliki tiga fungsi utama, yakni tanda tangan, verifikasi dan pengaturan. Dokumen yang ditandatangani telah diuji pada beberapa aplikasi pembaca berkas pdf dan tanda tangan elektronik dapat dikenali dan divalidasi. Bot juga dapat memverifikasi dokumen yang ditandatangani oleh aplikasi lain. Hasil pengujian terhadap komponen fungsional dan non-fungsional dengan metode black-box testing menunjukkan sistem dapat berjalan dengan baik sesuai yang diharapkan pada requirements planning. Hasil evaluasi kelayakan menggunakan System Usability Scale (SUS) menunjukkan sistem berada dalam kategori baik dan layak untuk digunakan.

**Korespondensi:**
Telepon: +62 (813) 94616622
E-mail:
hermanka.beta@gmail.com

## PENDAHULUAN

Menurut Undang-Undang Informasi dan Transaksi Elektronik (UU ITE), tanda tangan elektronik adalah tanda tangan yang terdiri atas informasi elektronik yang dilekatkan, terasosiasi atau terkait dengan informasi elektronik lainnya yang digunakan sebagai alat verifikasi dan autentikasi, dengan kata lain, tanda tangan elektronik dapat digunakan sebagai jaminan keabsahan dan keaslian sebuah dokumen elektronik (Dhagat, 2016) (Pereira, 2018). Pada proses penandatangan dokumen elektronik, pengguna membutuhkan sebuah sertifikat elektronik yang diterbitkan oleh *Certificate Authority* (CA). Peran CA tidak hanya menerbitkan, namun juga melakukan verifikasi terhadap sertifikat elektronik. Balai Sertifikasi Elektronik (BSrE) merupakan salah satu CA di Indonesia. BSrE adalah salah satu unit pelaksana teknis BSSN (Badan Siber dan Sandi Negara) yang bertugas dalam pemberian pelayanan penerbitan dan pengelolaan sertifikat elektronik kepada publik (Yusandy, 2019). BSrE telah merilis aplikasi penandatangan elektronik berbasis *desktop* kepada publik, namun salah satu kekurangan dari aplikasi berbasis *desktop* adalah rendahnya portabilitas (Singh, 2017). Pada beberapa wawancara yang telah dilakukan, pengguna sertifikat elektronik membutuhkan sebuah aplikasi penandatangan yang dapat dijalankan dimanapun berada, kapanpun dibutuhkan dan dapat berjalan pada semua platform sistem operasi.

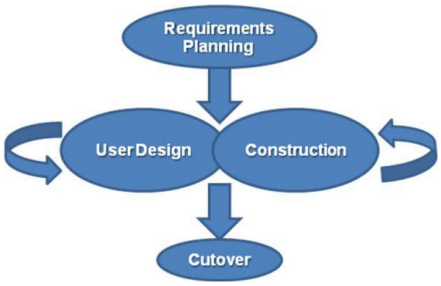
Salah satu solusi tercepat untuk membangun sebuah aplikasi *multiplatform* adalah dengan memanfaatkan fasilitas *chatbot* yang disediakan oleh aplikasi perpesanan (Dubosson, 2017). Penelitian ini akan menggunakan fasilitas *Chatbot* dari aplikasi Telegram. Istilah *Chatbot* sendiri merujuk pada *Bot* yang khusus berjalan pada aplikasi perpesanan yang dapat diintegrasikan dengan sistem-sistem eksternal (Korotaeva, 2018). Penelitian mengenai penggunaan *chatbot* Telegram telah banyak dilakukan, beberapa di antaranya pada bidang pelayanan (Rosid, 2018), pusat informasi (Calvo, 2017), hingga pengawasan (Husni, 2018). Pada penelitian yang dilakukan oleh Rosid (2018), peneliti mencoba meng-integrasikan *chatbot* Telegram dengan aplikasi *e-complaint* berbasis web yang sudah ada. Tujuan dari penelitian ini adalah untuk semakin mempermudah pengguna dalam menyampaikan keluhan. Pengguna tidak perlu membuka aplikasi *e-complaint* melalui *web browser*, namun cukup dengan berkomunikasi menggunakan aplikasi Telegram. Selain menangani keluhan, *chatbot* juga diatur dapat memberikan informasi seputar kampus. Pada penelitian Calvo (2017), dibangun sebuah *chatbot* bimbingan karir. Ide penelitiannya adalah menyediakan antarmuka yang mudah dan ramah bagi pengguna, di samping tujuan utamanya sebagai pengumpul data yang diperlukan untuk memberikan panduan karir. Dari kuesioner yang disebar diperoleh hasil responden sangat setuju dengan sistem *chatbot* tersebut, di samping mudah digunakan, responden juga berkomentar bahwa dengan menggunakan *chatbot*, sistem menjadi lebih mudah untuk dipelajari. Selain beberapa penelitian di atas, *Chatbot* juga dapat digunakan sebagai sistem pengawasan. Husni (2018) membangun sebuah *chatbot* yang mampu mengawasi dan memantau keberadaan pengemudi hingga perilaku pengemudi di jalan. Sistem berfungsi sebagai pengawas yang memberi tahu administrator rental mobil ketika pengemudi melakukan kesalahan berdasarkan aturan yang diinginkan dan memberi tahu pengemudi apa yang harus dilakukan.

Tujuan dari penelitian ini adalah merancang dan membangun sebuah aplikasi penandatangan dokumen elektronik yang bersifat *multiplatform* menggunakan *chatbot*, serta melakukan evaluasi kelayakan terhadap penggunaannya. Sertifikat yang digunakan dalam proses penandatanganan adalah standar sertifikat X509 dalam format PKCS#12. Sertifikat X509 telah banyak digunakan pada beberapa penelitian, diantaranya Forsby (2017) dan Karthikeyan (2019) yang menggunakan sertifikat X509 untuk

mengamankan perangkat-perangkat IoT. Sertifikat X509 sendiri merupakan salah satu standar sertifikat elektronik paling penting yang banyak digunakan pada beberapa mekanisme autentikasi (Vukasović, 2017).

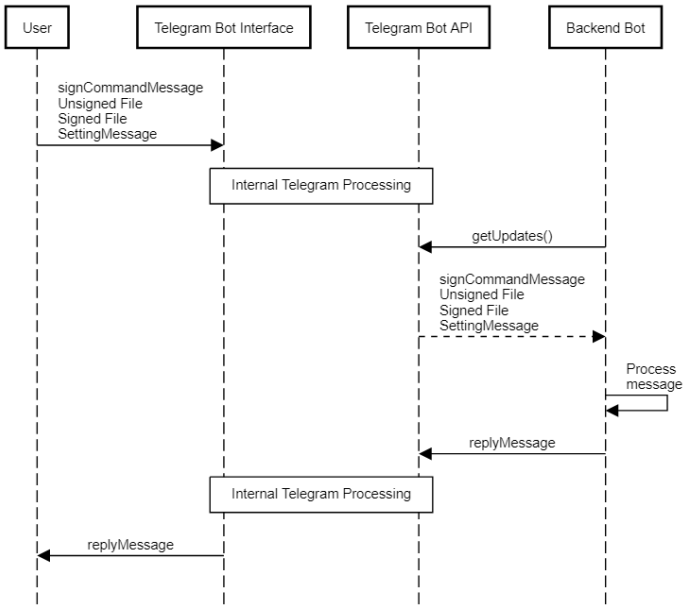
## METODE PENELITIAN

Metode yang digunakan dalam penelitian ini adalah *Rapid Application Development* (RAD). *Rapid Application Development* (RAD) *Life Cycle* pertama kali diperkenalkan oleh James Martin pada awal tahun 1990 (Kneuper, 2018). Menurut James Martin (Hassan, 2015) (Setyatama, 2018), *Rapid Application Development* (RAD) terdiri dari empat tahap yaitu *Requirements Planning, User Design, Construction dan Cutover*.



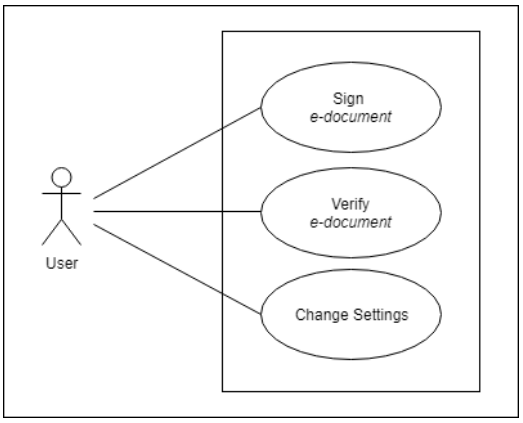

Gambar 1. Model *Rapid Application Development* (RAD) James Martin (Martin, 1991)

Tahapan penelitian pada Gambar 1. dapat dijelaskan sebagai berikut: 1) *Requirements Planning*, merupakan tahap pertama dari model RAD James Martin. Pada tahap ini dilakukan perencanaan berdasarkan kebutuhan yang diinginkan oleh pengguna serta analisis terhadap permasalahan. Data dikumpulkan melalui proses wawancara dan kuesioner yang melibatkan pengguna layanan dari BSrE khususnya para pemegang sertifikat elektronik. Berdasarkan hasil kuesioner dan wawancara, pengguna menginginkan sebuah aplikasi penandatangan dan pemverifikasi dokumen elektronik dengan mobilitas dan portabilitas yang tinggi, *user friendly*, serta dapat berjalan pada semua platform sistem operasi. Sebagai pemecahan masalah, maka akan dikembangkan sebuah *chatbot* Telegram yang dapat menandatangani dan memverifikasi dokumen elektronik.



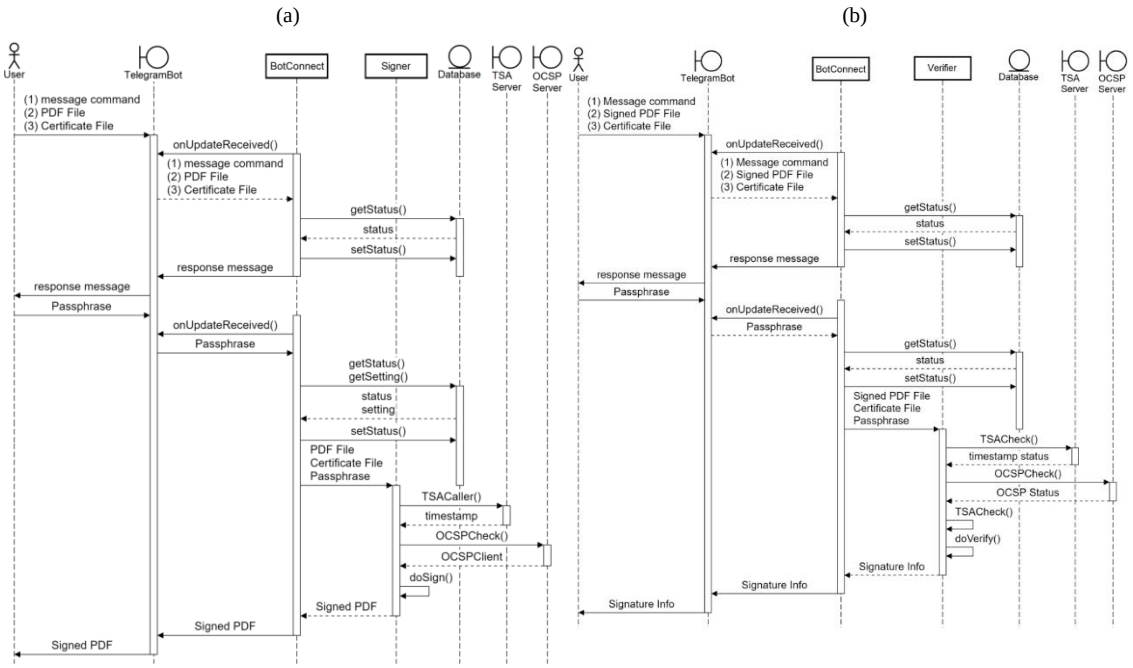

Gambar 2. Gambaran umum sistem

Gambaran umum sistem *chatbot* dapat dilihat pada Gambar 2 yang menunjukkan interaksi antara *user* dengan *chatbot*. *Chatbot* sendiri terdiri dari tiga entitas yang tidak dapat dipisahkan yakni *Telegram Bot Interface*, *Telegram Bot API* dan *Backend Bot*. Telegram Bot *Interface* berperan sebagai antarmuka user yang dalam hal ini adalah aplikasi Telegram itu sendiri, sedangkan Telegram Bot API berperan sebagai penghubung antara program *backend* dengan Telegram Bot *Interface*. Telegram Bot API berada di server Telegram, sedangkan *backend* terpasang di server peneliti. Program *backend* digunakan sebagai pemroses pesan, proses penandatangan dan verifikasi dokumen berjalan pada program *backend*. User mengirimkan pesan melalui Bot Telegram *Interface* yang dapat berupa pesan perintah atau pesan lampiran dokumen elektronik. Pesan-pesan yang diterima oleh antarmuka *chatbot* kemudian akan diproses secara internal untuk selanjutnya diteruskan ke Telegram Bot API. Secara berkala program *backend* akan meminta *update* pesan dari Telegram Bot API untuk kemudian diproses sesuai isi pesan yang masuk didalam program *backend*. Perlu diperhatikan bahwa pengiriman pesan dilakukan satu-persatu, alur pengiriman pesan sama seperti proses *chatting*. Alur dan urutan pengiriman secara detail dijabarkan pada rancangan *sequence diagram* pada tahap RAD selanjutnya.



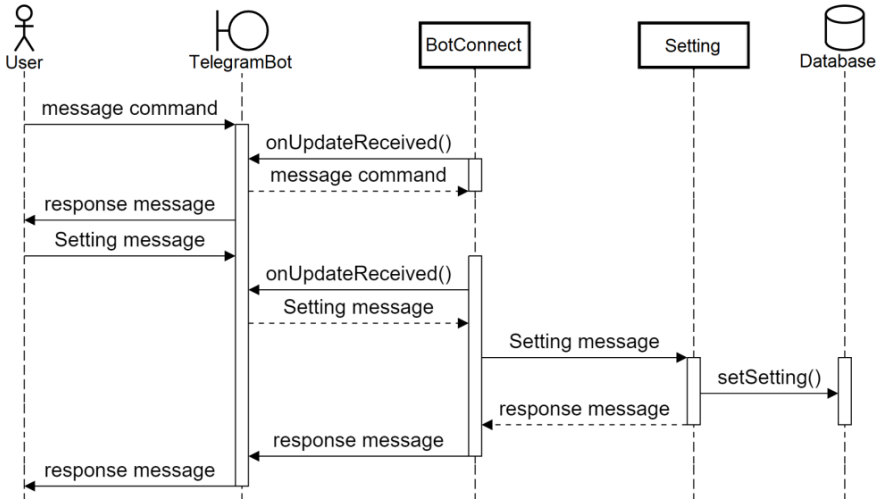

Gambar 3. *Use case diagram*

Tahapan kedua dari metode RAD yakni 2) *User Design*, tahap ini memastikan desain sistem sesuai dengan kebutuhan pengguna berdasarkan data pada tahap sebelumnya. Pada tahap ini ditentukan *input*, proses dan *output* yang diinginkan beserta layanan-layanan yang nantinya akan didukung oleh *chatbot*. Desain sistem ditetapkan menggunakan model UML. Diagram yang digunakan antara lain *Use Case Diagram*, *Sequence Diagram*, *Class Diagram* dan *Deployment Diagram*. Gambar 3 merupakan rancangan *use case* yang telah dibuat. Desain sistem memiliki tiga fungsi yaitu fungsi untuk menandatangani dokumen, memverifikasi dokumen, dan mengubah pengaturan khususnya tata letak tanda tangan elektronik. Rancangan skenario untuk setiap *use case* dituangkan dalam tiga *sequence diagram* di Gambar 4(a), Gambar 4(b) dan Gambar 5.




Gambar 4. *Sequence diagram* (a) skenario penandatanganan; (b) skenario verifikasi

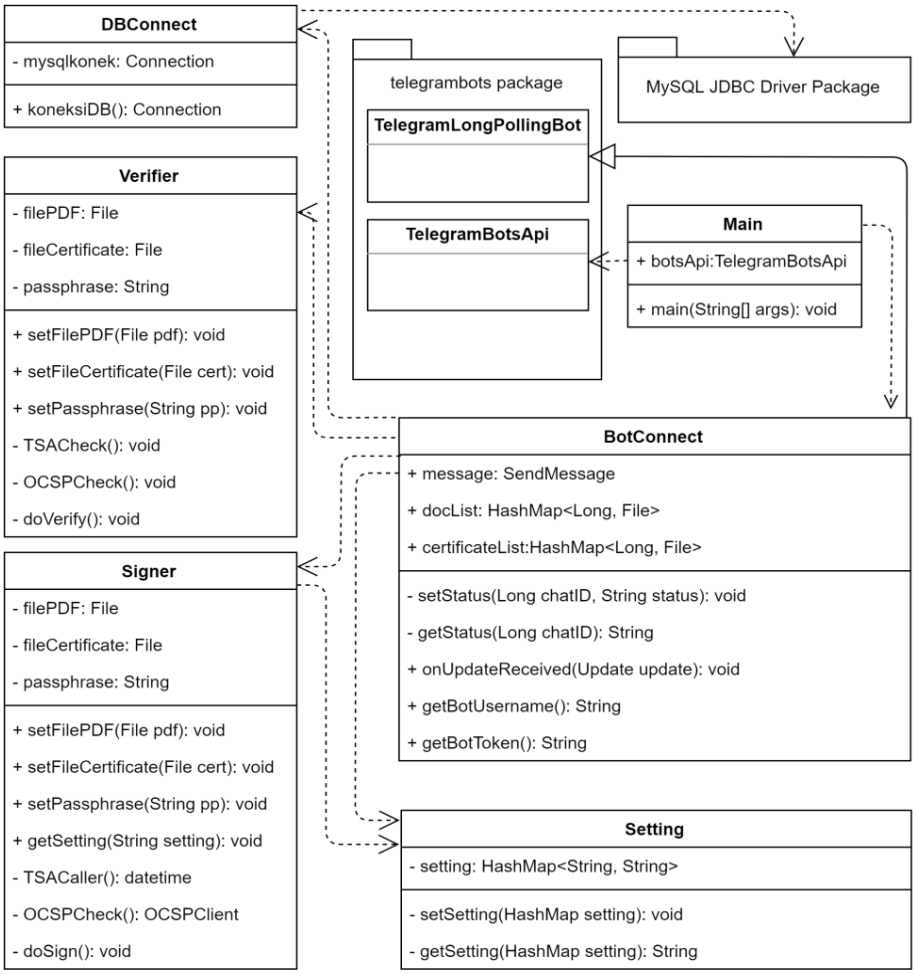
Berdasarkan rancangan *sequence diagram* (Gambar 4(a), 4(b) dan 5), dapat kita amati entitas-entitas yang terlibat antara lain, *User* sebagai aktor pengguna sistem, TelegramBot yang merupakan sistem internal Telegram, yang mencakup aplikasi Telegram, Bot Telegram dan Telegram Bot API. Aplikasi *backend* diwakilkan oleh dua entitas yakni BotConnect dan satu entitas untuk masing-masing skenario yaitu *Signer* untuk skenario penandatanganan, *Verifier* untuk skenario verifikasi dan *Setting* untuk skenario pengaturan letak posisi tanda tangan elektronik. *Database* sebagai entitas penyimpanan data sesi dan status *chat*. Entitas lain pada skenario penandatangan dan verifikasi yaitu TSAServer dan OCSPServer yang merupakan entitas eksternal yang dikelola oleh *Certificate Authority*. TSAServer dan OCSPServer berfungsi sebagai entitas yang memiliki otoritas untuk memeriksa validitas sertifikat elektronik dan *timestamp* pada tanda tangan elektronik.




Gambar 5. *Sequence diagram* skenario pengaturan

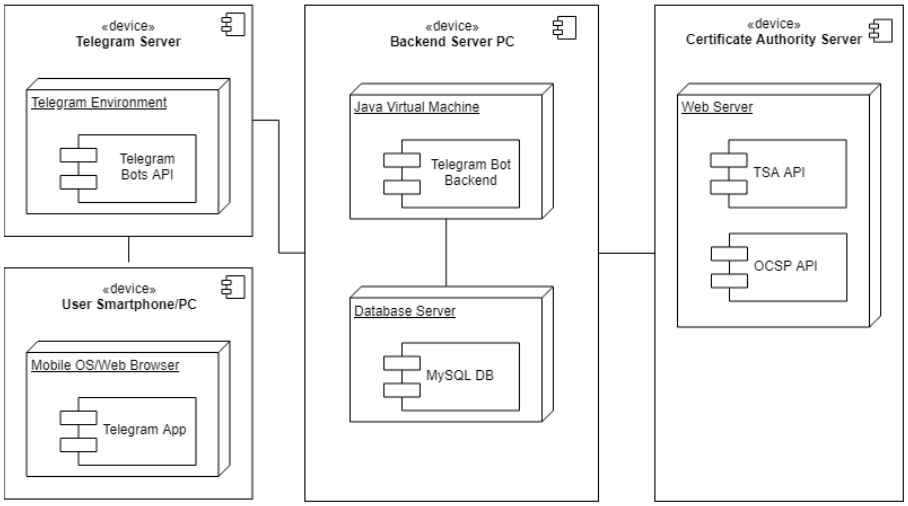
Aplikasi *backend chatbot* akan dikembangkan menggunakan bahasa pemrograman Java. Rancangan *class* untuk aplikasi *backend* dapat dilihat pada Gambar 6. Terdapat enam *class* yang terlibat, tiga class di antaranya yakni *Signer*, *Verifier* dan *Setting* berhubungan dengan skenario pada diagram *use case*. Selain

*class* internal, aplikasi *backend* juga membutuhkan *package* eksternal diantaranya MySQL JDBC *Driver* yang berfungsi sebagai *driver* koneksi dari aplikasi ke basis data, dan telegrambots yang merupakan *library* khusus untuk berkomunikasi dengan Telegram Bot API.




Gambar 6. *Class diagram*

Gambar 7. adalah rancangan *deployment*. *Deployment diagram* digunakan sebagai gambaran arsitektur sistem secara keseluruhan hingga lapisan perangkat keras yang digunakan. Dapat dilihat pada *deployment diagram*, peneliti hanya menyiapkan satu buah komputer *server* yang bertindak sebagai *backend*, dan sebuah *smartphone* untuk pengujian. Sedangkan server lain merupakan *server* eksternal dari pihak Telegram maupun dari pihak *Certificate Authority*.



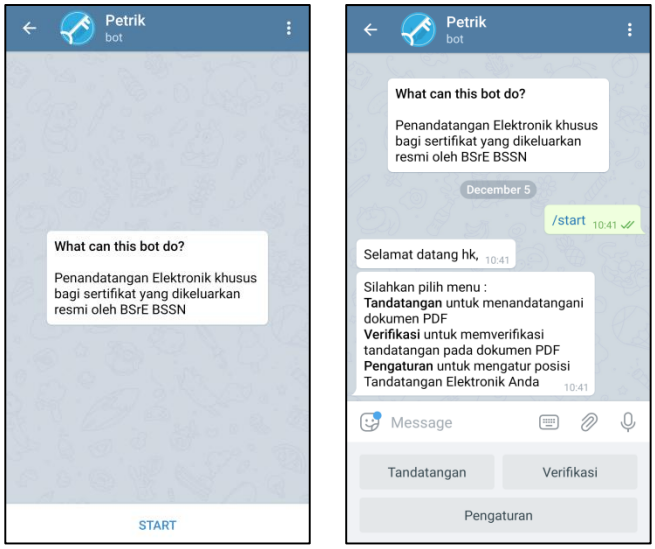

Gambar 7. *Deployment diagram*

Desain sistem yang telah dirancang pada tahap sebelumnya kemudian diimplementasikan menjadi sebuah sistem utuh, tahap ini pada metode RAD disebut dengan 3) *Construction*. Langkah pertama yang dilakukan adalah mendaftarkan *chatbot* melalui akun BotFather di Telegram. Selanjutnya, menuliskan kode untuk sistem *backend* menggunakan bahasa pemrograman Java. Koneksi *backend* dengan Telegram Bot API dilakukan menggunakan metode *long-polling*. Metode *long polling* adalah metode *default* yang digunakan untuk interaksi antara *backend* dengan Telegram API (Sucipto, 2019). Keuntungan penggunaan metode *long-polling* adalah tidak diperlukan sebuah *web server*, sedangkan kelemahannya adalah metode ini tidak berjalan secara *realtime*. Implementasi dan pengujian dilakukan berulang secara interaktif pada tahap *construction*. Tahap terakhir pada metode *Rapid Application Development* yaitu 4) *Cutover*, tahap ini terdiri dari proses *maintenance* dan *deployment*. Pada tahap *cutover* dilakukan pengujian akhir setelah *chatbot* melalui proses *deployment*. Pengujian fungsional dan non-fungsional dilakukan dengan metode *black-box* terhadap sistem *chatbot*. Evaluasi sistem menggunakan *System Usability Scale* (SUS) juga dilakukan untuk mengetahui tingkat kelayakan dari sistem yang akan dibangun.

## HASIL DAN PEMBAHASAN

Berdasarkan rancangan sistem, terdapat tiga proses yang akan diakomodasi oleh sistem yaitu proses tanda tangan, verifikasi dokumen, dan pengaturan letak posisi tanda tangan. Desain antarmuka *chatbot* tidak terlalu rumit karena antarmuka telah diakomodasi secara penuh oleh pihak Telegram. Pengembang hanya perlu fokus pada pesan-pesan balasan untuk setiap tahap proses yang berlangsung. Gambar 8(a) merupakan tampilan awal *chatbot* ketika pertama kali diakses atau ditambahkan pada *chatroom* Telegram. Pada tampilan awal ini, pengguna akan diberikan informasi sekilas mengenai kegunaan *chatbot* dan disediakan tombol "start" untuk memulai obrolan dengan *chatbot*. Gambar 8.(b) adalah tampilan ketika pengguna telah menekan tombol "start", kemudian muncul pesan balasan dari *chatbot* disertai pilihan menu yang disediakan di area *keyboard* yaitu "Tandatangan", "Verifikasi" dan "Pengaturan".

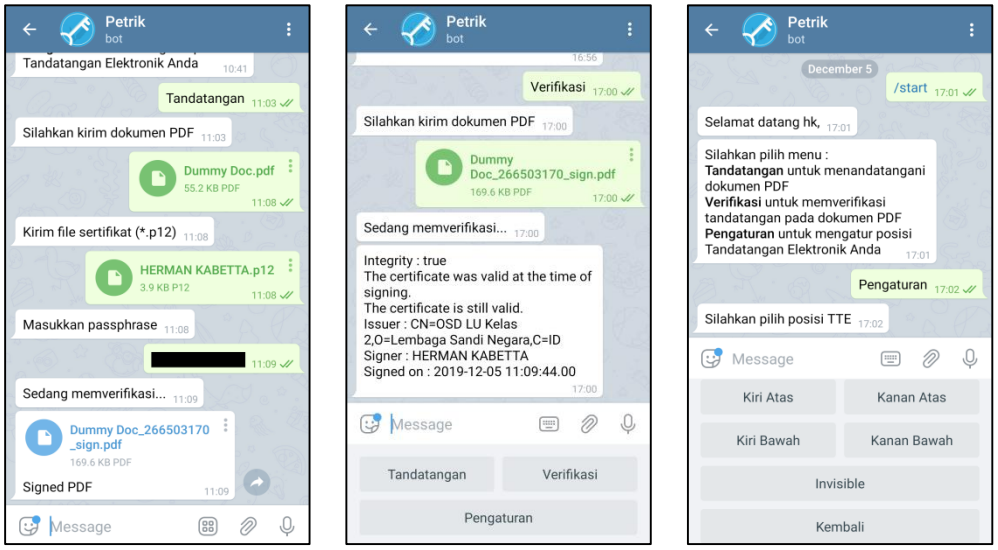
(a) (b)




Gambar 8. Tampilan awal *chatbot*, (a) Tampilan ketika pertama kali diakses; (b) Tampilan ketika tombol "start" ditekan.

Proses penandatanganan dokumen dimulai ketika pengguna menekan tombol "Tandatangan" pada menu awal. Sebelum memulai penandatanganan dokumen, pengguna harus memiliki sertifikat elektronik yang dikeluarkan oleh Balai Sertifikasi Elektronik (BSrE). Sertifikat elektronik biasanya diberikan berpasangan dengan *passphrase*-nya. *Passphrase* adalah kata sandi yang digunakan sebagai autentikasi kepemilikan sertifikat elektronik.

(a) (b) (c)




Gambar 9. Tampilan percakapan (a) Proses tanda tangan dokumen; (b) Proses verifikasi dokumen; (c) Menu pengaturan posisi tanda tangan

Gambar 9(a) merupakan tampilan saat percakapan penandatanganan dokumen. Pada awal percakapan, pengguna diminta untuk mengirimkan dokumen pdf yang akan ditandatangani, dilanjutkan kemudian mengirim berkas sertifikat elektronik dan diteruskan dengan memasukkan *passphrase* dari sertifikat elektronik tersebut. Percakapan berlangsung satu per satu bergantian balas membalas antara *chatbot* dan pengguna. Pada akhir percakapan, *chatbot* mengirimkan dokumen yang telah ditandatangani. Fitur kedua *chatbot* adalah verifikasi tanda tangan elektronik pada dokumen pdf. Gambar 9(b) merupakan

tampilan percakapan saat proses verifikasi dokumen. Pada awal percakapan, pengguna diminta untuk mengirimkan dokumen pdf yang bertandatangan. Selanjutnya *chatbot* akan memproses dan melakukan pengecekan terhadap dokumen, kemudian mengirimkan hasil verifikasi dokumen kepada pengguna. Hasil verifikasi berupa integritas dokumen, validitas sertifikat elektronik, identitas *Certificate Authority*, penandatangan dokumen serta waktu penandatanganan. Gambar 9(c) merupakan tampilan percakapan dan menu untuk mengatur posisi letak tanda tangan elektronik. Terdapat lima menu pilihan posisi, kiri atas, kanan atas, kiri bawah, kanan bawah dan *invisible* (tidak nampak). Pengaturan posisi tanda tangan akan tersimpan pada basis data, sehingga pengguna tidak perlu mengatur berulang kali setiap berkomunikasi dengan *chatbot*.

Pengujian sistem dilakukan untuk memastikan *chatbot* dapat berjalan sesuai dengan spesifikasi dan ekspektasi pengguna. Menurut Ul Haq (2019), terdapat banyak variasi pengujian perangkat lunak, salah satunya dapat dilakukan dengan metode *black-box.* Berikut ini skenario pengujian fungsional dengan metode *black-box* yang dilakukan terhadap sistem *chatbot* dengan daftar disajikan pada Tabel 1.

Tabel 1. Daftar skenario pada pengujian fungsional *chatbot*

| ***No.*** | ***Spesifikasi*** | ***Test Case ID*** | ***Test Case*** |
|---|---|---|---|
| 1 | Tanda tangan dokumen | C01 | Proses tanda tangan dengan format dokumen valid, sertifikat valid dan *passphrase* benar |
| | | C02 | Proses tanda tangan dengan format dokumen valid, sertifikat valid dan *passphrase* salah atau *invalid* |
| | | C03 | Proses tanda tangan dengan format dokumen valid, sertifikat *invalid* atau kadaluwarsa |
| | | C04 | Proses tanda tangan dengan format dokumen *invalid* |
| 2 | Verifikasi dokumen | C05 | Proses verifikasi dengan format dokumen valid |
| | | C06 | Proses verifikasi dengan format dokumen *invalid* |
| | | C07 | Proses verifikasi dengan dokumen yang belum ditandatangani |
| 3 | Pengaturan posisi tanda tangan | C08 | Proses pengaturan dengan perintah valid |
| | | C09 | Proses pengaturan dengan perintah *invalid* |
| | | C10 | Menandatangani dokumen setelah perubahan pengaturan posisi tanda tangan |
| 4 | *Illegal message handling* | C11 | Menuliskan pesan perintah ilegal pada awal percakapan |

Pada spesifikasi pertama yaitu proses tanda tangan dokumen. Masing-masing skenario mewakili tiap tahap percakapan yang berjalan secara berurutan, sehingga percakapan tidak akan berlanjut apabila percakapan sebelumnya tidak valid. Hal ini berlaku pula pada spesifikasi pengujian yang lainnya. Urutan tahap percakapan *user* pada proses tanda tangan adalah, 1) Mengirimkan pesan perintah "Tandatangan", 2) Mengirimkan dokumen PDF, 3) Mengirimkan berkas sertifikat elektronik, 4) Mengirimkan *passphrase*. Masing-masing tahap percakapan kemudian diuji dengan data valid dan tidak valid sehingga terdapat empat skenario yang diuji, daftar skenario secara lengkap disajikan pada Tabel 2.

Tabel 2. Skenario pengujian untuk spesifikasi tanda tangan dokumen

| ***ID*** | ***Test Case*** | ***Step Detail / Input Value*** | ***Expected Result*** | ***Actual Result*** |
|---|---|---|---|---|
| C01 | Proses tanda tangan dengan format | Mengirimkan pesan teks "Tandatangan" | Chatbot memberikan pesan balasan "Silahkan kirim dokumen PDF" | Sesuai |

| | | | | |
|---|---|---|---|---|
| | dokumen valid, sertifikat valid dan *passphrase* benar | Mengirimkan dokumen PDF | Chatbot memberikan pesan balasan "Kirim file sertifikat (*.p12)" | Sesuai |
| | | Mengirimkan berkas sertifikat elektronik (*.p12) yang masih berlaku | Chatbot memberikan pesan balasan "Masukkan passphrase" | Sesuai |
| | | Mengirimkan pesan teks berupa passphrase yang benar dari sertifikat elektronik | *Chatbot* memberikan dokumen yang telah ditandatangani | Sesuai |
| C02 | Proses tanda tangan dengan format dokumen valid, sertifikat valid dan *passphrase* salah atau *invalid* | Mengirimkan pesan teks "Tandatangan" | Chatbot memberikan pesan balasan "Silahkan kirim dokumen PDF" | Sesuai |
| | | Mengirimkan dokumen PDF | Chatbot memberikan pesan balasan "Kirim file sertifikat (*.p12)" | Sesuai |
| | | Mengirimkan berkas sertifikat elektronik (*.p12) yang masih berlaku | Chatbot memberikan pesan balasan "Masukkan passphrase" | Sesuai |
| | | Mengirimkan pesan teks berupa *passphrase* yang salah | Muncul peringatan *passphrase* tidak sesuai | Sesuai |
| C03 | Proses tanda tangan dengan format dokumen valid, sertifikat *invalid* atau kadaluwarsa | Mengirimkan pesan teks "Tandatangan" | Chatbot memberikan pesan balasan "Silahkan kirim dokumen PDF" | Sesuai |
| | | Mengirimkan dokumen PDF | Chatbot memberikan pesan balasan "Kirim file sertifikat (*.p12)" | Sesuai |
| | | Mengirimkan berkas selain sertifikat p12 (pdf, doc, jpg, xls) atau mengirimkan berkas sertifikat elektronik yang kadaluwarsa | Chatbot memberikan pesan peringatan "Sertifikat kadaluwarsa atau tidak valid" | Sesuai |
| C04 | Proses tanda tangan dengan format dokumen *invalid* | Mengirimkan pesan teks "Tandatangan" | Chatbot memberikan pesan balasan "Silahkan kirim dokumen PDF" | Sesuai |
| | | Mengirimkan dokumen selain PDF (doc, jpg, xls) | Chatbot memberikan pesan peringatan "Hanya dapat menandatangani berkas PDF" | Sesuai |

Spesifikasi kedua adalah proses verifikasi dokumen, terdiri dari tiga skenario yang disajikan pada tabel 3. Urutan tahap percakapan *user* pada proses tanda tangan adalah, 1) Mengirimkan pesan perintah "Verifikasi", 2) Mengirimkan dokumen PDF. Pada proses ini, terdapat tiga masukan data berbeda yang diujikan, yaitu berupa dokumen PDF yang telah ditandatangai, dokumen PDF yang belum ditandatangani dan dokumen selain PDF.

Tabel 3. Skenario pengujian untuk spesifikasi verifikasi dokumen

| ***ID*** | ***Test Case*** | ***Step Detail / Input Value*** | ***Expected Result*** | ***Actual Result*** |
|---|---|---|---|---|
| C05 | Proses verifikasi dengan format dokumen valid | Mengirimkan pesan teks "Verifikasi" | Chatbot memberikan pesan balasan "Silahkan kirim dokumen PDF" | Sesuai |
| | | Mengirimkan dokumen PDF yang telah ditandatangani | *Chatbot* memberikan pesan balasan yang berisi informasi sertifikat elektronik | Sesuai |
| C06 | Proses verifikasi dengan format dokumen *invalid* | Mengirimkan pesan teks "Verifikasi" | Chatbot memberikan pesan balasan "Silahkan kirim dokumen PDF" | Sesuai |
| | | Mengirimkan dokumen selain PDF | Chatbot memberikan pesan peringatan "Hanya dapat memverifikasi berkas PDF" | Sesuai |
| C07 | Proses verifikasi dengan dokumen yang belum ditandatangani | Mengirimkan pesan teks "Verifikasi" | Chatbot memberikan pesan balasan "Silahkan kirim dokumen PDF" | Sesuai |
| | | Mengirimkan dokumen PDF yang belum ditandatangani | Chatbot memberikan pesan peringatan "Tidak ditemukan tanda tangan elektronik" | Sesuai |

Spesifikasi ketiga adalah proses pengaturan posisi tanda tangan, terdiri dari tiga skenario yang disajikan pada tabel 4. Urutan tahap percakapan *user* pada proses pengaturan adalah, 1) Mengirimkan pesan perintah "Pengaturan", 2) Mengirimkan pesan pengaturan. Terdapat lima pesan pengaturan yang dapat dipilih, yaitu "Kiri Atas", "Kanan Atas", "Kiri Bawah", "Kanan Bawah" dan "Invisible". Pengujian dilakukan untuk masing-masing pesan pengaturan tersebut beserta pesan pengaturan yang tidak terdapat pada daftar, kemudian dilakukan proses tanda tangan untuk menguji posisi tanda tangan sudah berada pada posisi yang semestinya.

Tabel 4. Skenario pengujian untuk spesifikasi pengaturan posisi tanda tangan

| *ID* | *Test Case* | *Step Detail / Input Value* | *Expected Result* | *Actual Result* |
|---|---|---|---|---|
| C08 | Proses pengaturan dengan perintah valid | Mengirimkan pesan teks “Pengaturan” | Chatbot memberikan pesan balasan “Silahkan pilih posisi TTE” | Sesuai |
| | | Mengirimkan salah satu pesan pengaturan yaitu “Kiri Atas”, “Kanan Atas”, “Kiri Bawah”, Kanan Bawah” atau ”invisible” | *Chatbot* memberikan pesan balasan “Pengaturan telah disimpan” | Sesuai |
| C09 | Proses pengaturan dengan perintah *invalid* | Mengirimkan pesan teks “Pengaturan” | Chatbot memberikan pesan balasan “Silahkan pilih posisi TTE” | Sesuai |
| | | Mengirimkan pesan teks selain “Kiri Atas”, “Kanan Atas”, “Kiri Bawah”, Kanan Bawah” atau ”invisible” | *Chatbot* memberikan pesan balasan “Pengaturan tidak dikenal” | Sesuai |
| C10 | Menandatangani dokumen setelah perubahan pengaturan posisi tanda tangan | Mengirimkan pesan teks “Tandatangan” | Chatbot memberikan pesan balasan “Silahkan kirim dokumen PDF” | Sesuai |
| | | Mengirimkan dokumen PDF | Chatbot memberikan pesan balasan “Kirim file sertifikat (*.p12)” | Sesuai |
| | | Mengirimkan berkas sertifikat elektronik (*.p12) yang masih berlaku | Chatbot memberikan pesan balasan “Masukkan passphrase” | Sesuai |
| | | Mengirimkan pesan teks berupa passphrase yang benar dari sertifikat elektronik | Chatbot memberikan dokumen yang telah ditandatangani dengan posisi sesuai pengaturan yang baru | Sesuai |

*Illegal message handling* merupakan skenario untuk menangani kesalahan penulisan perintah pada awal percakapan dengan *chatbot*. Awal percakapan dengan *chatbot* dimulai setelah *user* menekan tombol ”START”. Pada awal percakapan, *user* diminta untuk memilih satu diantara tiga menu yaitu “Tandatangan”, “Verifikasi” dan “Pengaturan”. Penanganan perlu dilakukan apabila *user* tidak memilih satu diantara tiga daftar menu tersebut, atau salah menuliskan nama menu. Detail skenario disajikan pada tabel 5.

Tabel 5. Skenario pengujian untuk spesifikasi *illegal message handling*

| *ID* | *Test Case* | *Step Detail / Input Value* | *Expected Result* | *Actual Result* |
|---|---|---|---|---|
| C11 | Menuliskan perintah ilegal pada awal percakapan | Mengirimkan pesan berupa teks yang tidak terdaftar, contoh :<br>• “tanda tangan”<br>• “sign” | Chatbot memberikan pesan peringatan “Perintah tidak dikenali” | Sesuai |

Pengujian non-fungsional dilakukan untuk menguji aspek *compatibility chatbot* apakah dapat berjalan dengan baik pada semua platform sistem operasi, dan apakah tanda tangan elektronik yang tersimpan pada dokumen dapat dikenali oleh aplikasi pembaca berkas pdf lainnya. Dua skenario pengujian non-fungsional diujikan pada sistem *chatbot* dengan hasil disajikan pada Tabel 6 dan Tabel 7.

Tabel 6. Daftar skenario pada pengujian non-fungsional *chatbot*

| *No.* | *Domain* | *Test Case ID* | *Test Case* |
|---|---|---|---|
| 1 | *Compatibility* | C12 | *Chatbot* dapat berjalan di sistem operasi *mobile* (Android dan iOS), sistem operasi *desktop* (Windows, MacOS) dan web. |
| | | C13 | Tanda tangan elektronik dapat dikenali oleh aplikasi pembaca PDF |

Tabel 7. Skenario pengujian untuk domain *compatibility*

| *ID* | *Test Case* | *Step Detail / Input Value* | *Expected Result* | *Actual Result* |
|---|---|---|---|---|
| C12 | *Chatbot* dapat berjalan di sistem operasi *mobile* (Android dan iOS), sistem operasi *desktop* (Windows, MacOS) dan web. | Menjalankan proses tanda tangan, verifikasi dan pengaturan pada setiap platform | *Chatbot* dapat berjalan dengan baik pada semua platform sistem operasi dan web | Sesuai |
| C13 | Tanda tangan elektronik dapat dikenali oleh aplikasi pembaca PDF | Memvalidasi tanda tangan elektronik menggunakan aplikasi Adobe Acrobat Reader dan Foxit PDF Reader | Tanda tangan elektronik dapat dikenali pada aplikasi Adobe Acrobat Reader dan Foxit PDF Reader | Sesuai |

Evaluasi sistem dilakukan menggunakan *System Usability Scale* (SUS) untuk mengetahui tingkat kelayakan dan kegunaan dari *chatbot*. *System Usability Scale* (SUS) yang diciptakan oleh John Brooke pada tahun 1986, merupakan salah satu metode yang dapat digunakan untuk mengukur persepsi kegunaan sebuah perangkat keras maupun perangkat lunak (Boyd, 2018). Responden kuesioner SUS berjumlah sepuluh orang yang merupakan pengguna sertifikat elektronik BsrE dan pengumpulan data dilakukan secara daring menggunakan kuesioner dari *Google Forms*. Hasil pengumpulan data disajikan pada tabel 8.

Tabel 8. Hasil kuesioner *System Usability Scale*

| *No.* | *Pertanyaan* | *Rata-rata skor* |
|---|---|---|
| 1 | Saya sepertinya akan sering menggunakan *chatbot* ini | 2,4 |
| 2 | Ada fitur pada *chatbot* yang sebenarnya tidak perlu | 3,9 |
| 3 | Saya merasa mudah menggunakan *chatbot* ini | 1,9 |
| 4 | Saya sepertinya perlu bantuan teknis untuk mengoperasikan *chatbot* ini | 4,1 |
| 5 | Saya menemukan berbagai fungsi dalam *chatbot* ini telah terintegrasi dengan baik | 2,5 |
| 6 | Saya pikir ada terlalu banyak ketidaksesuaian dalam *chatbot* dan sistem pendukungnya | 4,5 |
| 7 | Saya rasa mayoritas pengguna akan dapat mempelajari *chatbot* ini dengan cepat | 2,4 |
| 8 | Saya merasa *chatbot* ini sangat tidak praktis ketika digunakan | 4,3 |
| 9 | Saya sangat yakin dapat menggunakan *chatbot* ini | 2,5 |
| 10 | Sepertinya saya harus belajar banyak hal terlebih dahulu sebelum saya dapat menggunakan *chatbot* ini | 4,3 |
| | **Total** | 32,8 |
| | **Skor SUS** (2,5 * Total) | 82,0 |

SUS terdiri dari 10 pertanyaan, dengan lima pertanyaan positif dan 5 pertanyaan negatif serta setiap pertanyaan memiliki bobot 0 sampai 4. Pertanyaan nomor ganjil merupakan pertanyaan positif, skor setiap pertanyaan dihitung dengan cara bobot tiap pertanyaan dikurangi dengan nilai 1 (bobot - 1). Pertanyaan genap yang merupakan pertanyaan negatif, skor dihitung dengan cara 5 dikurangi bobot setiap pertanyaan (5 - bobot). Total skor diperoleh dari jumlah rata-rata skor setiap pertanyaan, kemudian total skor dikalikan 2,5 untuk mendapatkan skor SUS antara 0-100. Menurut Baumgartner (2019), standar minimum skor SUS adalah 65 untuk memastikan bahwa produk dapat diterima oleh pengguna. Dari hasil perhitungan diperoleh skor SUS sebesar 82,0 yang menunjukkan bahwa sistem yang dibangun berada dalam kategori baik dan layak untuk digunakan.

## KESIMPULAN DAN SARAN

Berdasarkan penelitian dan pengujian yang telah dilakukan, dapat disimpulkan:

1. *Chatbot* penandatangan elektronik dapat dibangun dengan metode *Rapid Application Development* (RAD) dan dapat diakses dengan mudah melalui aplikasi perpesanan Telegram.
2. Dokumen yang telah ditandatangani *chatbot* terbukti dapat dikenali dan divalidasi oleh aplikasi pembaca pdf lainnya.
3. Pengujian dilakukan pada komponen fungsional dan non-fungsional. Hasil pengujian terhadap tiga fungsi utama (fungsional) *chatbot* yaitu tanda tangan, verifikasi dan pengaturan, menunjukkan hasil yang sesuai dengan *requirements planning* dan *chatbot* dapat berjalan dengan baik. Hasil pengujian terhadap komponen non-fungsional menunjukkan *chatbot* dapat berjalan dengan baik pada semua platform sistem operasi maupun pada platform web.
4. Evaluasi sistem menggunakan *System Usability Scale* (SUS) menghasilkan skor sebesar 82,0 yang menunjukkan bahwa *chatbot* berada dalam kategori baik dan layak untuk digunakan.

Adapun saran untuk penelitian lebih lanjut adalah perlunya pengembangan terhadap mekanisme pengamanan data, mengingat *chatbot* menggunakan server dari pihak ketiga yakni Telegram, serta pengembangan lebih lanjut agar *chatbot* dapat menandatangani sertifikat dari *Certificate Authority* selain BSrE.

## DAFTAR PUSTAKA


Baumgartner, J., Frei, N., Kleinke, M., Sauer, J., & Sonderegger, A. (2019). Pictorial System Usability Scale (P-SUS) Developing an Instrument for Measuring Perceived Usability. *In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (pp. 1-11)*.

Boyd, K., Bond, R., Magee, J., & McCormack, P. (2018). Can users recall their user experience with a technology? Temporal bias and the system usability scale. *In Proceedings of the 32nd International BCS Human Computer Interaction Conference 32 (pp. 1-6)*.

Calvo D., Quesada L., López G., Guerrero L.A. (2017) Multiplatform Career Guidance System Using IBM Watson, Google Home and Telegram. In: Ochoa S., Singh P., Bravo J. (eds) Ubiquitous Computing and Ambient Intelligence. UCAmI 2017. *Lecture Notes in Computer Science, vol 10586*. Springer, Cham.

Dhagat, R. and Joshi, P., (2016). New Approach of User Authentication Using Digital Signature. *Symposium on Colossal Data Analysis and Networking (CDAN)* ( pp. 1-3). Indore.

Dubosson, F., Schaer, R., Savioz, R., & Schumacher, M. (2017). Going beyond the relapse peak on social network smoking cessation programmes: ChatBot opportunities. *Swiss medical informatics*, *33*(00).

Forsby, F., Furuhed, M., Papadimitratos, P., & Raza, S. (2017). Lightweight X. 509 digital certificates for the Internet of Things. *In Interoperability, Safety and Security in IoT* (pp. 123-133). Springer, Cham.

Hassan, S., Qamar, U., & Idris, M. A. (2015). Purification of requirement engineering model for rapid application development. In *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)* (pp. 357-362). IEEE.

Husni, E., & Hasibuan, F. (2018). Driver Supervisor System with Telegram Bot Platform. In *International Conference on Computational Collective Intelligence* (pp. 436-444). Springer, Cham.

Karthikeyan, S., Patan, R., & Balamurugan, B. (2019). Enhancement of Security in the Internet of Things (IoT) by Using X. 509 Authentication Mechanism. *In Recent Trends in Communication, Computing, and Electronics* (pp. 217-225). Springer, Singapore.

Kneuper R. (2018). Software Processes in the Software Product Life Cycle. In: Software Processes and Life Cycle Models. Springer, Cham.

Korotaeva, D., Khlopotov, M., Makarenko, A., Chikshova, E., Startseva, N., & Chemysheva, A. (2018). Botanicum: a Telegram Bot for Tree Classification. In *2018 22nd Conference of Open Innovations Association (FRUCT)* (pp. 88-93). IEEE.

Martin, J. (1991). *Rapid application development*. Macmillan Publishing Co., Inc..

Pemerintah Indonesia. (2016). Undang-Undang Nomor 19 Tahun 2016 Tentang Perubahan Atas Undang-Undang Nomor 11 Tahun 2008 Tentang Informasi dan Transaksi Elektronik. Sekretariat negara, Jakarta.

Pereira, C., Barbosa, L., Martins, J., & Borges, J. (2018). Digital Signature Solution for Document Management Systems-The University of Trás-os-Montes and Alto Douro. *In World Conference on Information Systems and Technologies* (pp. 16-25). Springer, Cham.

Rosid, M. A., Rachmadany, A., Multazam, M. T., Nandiyanto, A. B. D., Abdullah, A. G., & Widiaty, I. (2018). Integration Telegram Bot on E-Complaint Applications in College. In *IOP Conference Series: Materials Science and Engineering* (Vol. 288, No. 1, p. 012159). IOP Publishing.

Setyatama, F., & IrwanKurnia, A. (2018). Rapid Application Development (RAD) Method For Developing Clinical Laboratory Information System (Case Study: PT. Populer Sarana Medika). *Journal of Electrical Engineering And Computer Sciences, Vol. 3 Number 2*, 3(2).

Singh, D. A. A. G., Leavline, E. J., & Vijayan, P. M. (2017). Mobile Application for Student Attendance and Mark Management System. *International Journal of Computational Intelligence Research, 13(3)*, 425-432.

Sucipto, S., Resti, N. C., Andriyanto, T., Karaman, J., & Qamaria, R. S. (2019). Transactional Database Design Information System Web-Based Tracer Study Integrated Telegram Bot. *Journal of Physics: Conference Series (Vol. 1381, No. 1, p. 012008)*. IOP Publishing.

Ul Haq, S., & Qamar, U. (2019). Ontology Based Test Case Generation for Black Box Testing. *Proceedings of the 2019 8th International Conference on Educational and Information Technology (pp. 236-241).* Association for Computing Machinery (ACM).

Vukasović, M., Veselinović, B., & Stanisavljević, Ž. (2017). A development of a configurable system for handling X509 certificates. In *2017 25th Telecommunication Forum (TELFOR)* (pp. 1-4). IEEE.

Yusandy, T. (2019). Kedudukan dan Kekuatan Pembuktian Alat Bukti Elektronik dalam Hukum Acara Perdata Indonesia. *Jurnal Serambi Akademica*, 7(5), 645-656.

Terbit *online* pada laman web jurnal :
http://ejournal.amikompurwokerto.ac.id/index.php/telematika/

**Telematika**

Terakreditasi Sinta "3" KEMENRISTEKDIKTI, No. 21/E/KPT/2018

# Optimasi Algoritme *Naïve Bayes* untuk Klasifikasi Data Gempa Bumi di Indonesia Berdasarkan Hiposentrum

**Rastri Prathivi**

Teknik Informatika
Fakultas Teknologi Informasi dan Komunikasi
Universitas Semarang
Email: vivi@usm.ac.id

**INFO ARTIKEL**

***Sejarah Artikel:***
Menerima 21 November 2019
Revisi 13 Januari 2020
Received 27 Februari 2020
*Online* 29 Februari 2020

***Keyword*:**
*Earthquake Classification*
*Hypocenter*
*Adaboost*
*Naïve Bayes*

**Kata Kunci:**
Klasifikasi Gempa
Hiposentrum
*Adaboost*
*Naïve Bayes*

**Korespondensi:**
Telepon: +62 085950251065
E-mail: vivi@usm.ac.id

***ABSTRACT***

*The Hiposentrum or epicentre is the source of an earthquake which is at a certain depth on earth. The classification of earthquake powers based on the depth of Hiposentrum needed to examine the potential earthquake powers spread in Indonesian territory. The results of the classification process often experience problems, namely inaccuracy in classification. To solve that problem, then algorithms optimising classification must be increased. This research uses the Naïve Bayes algorithm, which is optimized using the Adaboost algorithm. Evaluation of the results of the optimized classification algorithm is needed to determine the level of accuracy using prescriptions and recall. In this study, the object of research is earthquake data in Indonesia which will be used as training data and testing data. The average accuracy of the Naïve Bayes algorithm is 72.3%, and the Naïve Bayes and Adaboost algorithm is 85.3%.*

**ABSTRAK**

Hiposentrum atau pusat gempa merupakan sumber gempa yang terdapat pada kedalaman tertentu di bumi. Klasifikasi kekuatan gempa berdasarkan kedalaman hiposentrum diperlukan untuk mengetahui potensi kekuatan gempa yang tersebar di wilayah Indonesia. Hasil dari proses klasifikasi seringkali mengalami masalah yaitu ketidaktepatan dalam klasifikasi. Untuk mengatasi masalah tersebut maka algoritme klasifikasi perlu ditingkatkan optimasinya. Penelitian ini menggunakan algoritme *Naïve Bayes* yang dioptimasi menggunakan algoritme *Adaboost*. Evaluasi terhadap hasil dari algoritme klasifikasi yang telah dioptimasi diperlukan untuk mengetahui tingkat akurasi menggunakan *presicion* dan *recall*. Dalam penelitian ini objek penelitian berupa data gempa bumi di Indonesia yang akan digunakan sebagai data *training* dan data *testing*. Hasil rata - rata akurasi algoritme *Naïve Bayes* sebesar 72,3% dan algoritme *Naïve Bayes* dan *Adaboost* sebesar 85,3%.

## PENDAHULUAN

Hiposentrum atau pusat gempa merupakan sumber gempa yang terdapat pada kedalaman tertentu di bumi. Kekuatan suatu gempa sangat tergantung pada lokasi hiposentrumnya. Menurut Hartuti (2009) klasifikasi gempa berdasarkan hiposentrumnya diklasifikasikan menjadikan tiga bagian yaitu gempa bumi dalam adalah gempa bumi yang terjadi dengan kedalaman hiposentrum >300 km di bawah permukaan bumi. Gempa bumi menengah yaitu gempa bumi yang terjadi dengan kedalaman hiposentrum berkisar antara 60 km sampai 300 km di bawah permukaan bumi. Gempa bumi dangkal yaitu gempa bumi yang terjadi dengan kedalaman hiposentrum <60 km di bawah permukaan bumi. Berdasarkan letak wilayah geografisnya Indonesia adalah negara kepulauan yang memiliki potensi gempa bumi yang besar.

Data pada Badan Meteorologi, Klimatologi dan Geofisika (BMKG) mencatat terjadinya gempa bumi di Indonesia secara berkala dalam selang waktu beberapa bulan saja. Bahkan potensi gempa bumi yang lokasi hiposentrumnya cukup dalam hampir terjadi setiap hari.

Klasifikasi kekuatan gempa berdasarkan kedalaman hiposentrum diperlukan untuk mengetahui potensi kekuatan gempa yang tersebar di wilayah Indonesia. Hasil dari proses klasifikasi seringkali mengalami masalah yaitu ketidaktepatan dalam klasifikasi. Untuk mengatasi masalah tersebut maka algoritme klasifikasi perlu ditingkatkan optimasinya. Menurut Saritas (saritas, 2019) algoritme *Naïve Bayes* merupakan algoritme probabilitas sederhana untuk mengklasifikasikan data berdasarkan probabilitas data dengan menghitung frekuensi dan kombinasi nilai pada dataset yang digunakan. Di dalam algoritme *Naïve Bayes* setiap data diasumsikan sebagai variabel bebas yang dapat mempertimbangkan nilai dari variabel klasifikasi. Pada penelitian yang dilakukan oleh Saritas (saritas 2019) performa akurasi algoritme *Naïve Bayes* sebesar 83.54 terhadap *algoritme* ANN 86.95 yang digunakan untuk mengklasifikasikan data kanker payudara. Dalam penelitian yang ditemukan oleh Nakra (2019) algoritme *Naïve Bayes* memiliki tingkat akurasi yang cukup baik yaitu 79.6935 terhadap algoritme Bayes Net sebesar 78.5441 yang dihitung dengan tool WEKA. Dalam penelitian tersebut algoritme *Naïve Bayes* masih dimungkinkan untuk ditingkatkan akurasinya dengan mengoptimasi algoritme *Naïve Bayes* tersebut. Dalam penelitian ini penulis yang akan meningkatkan optimasi algoritme *Naïve Bayes* yang diterapkan untuk mengklasifikasikan dataset gempa bumi berdasarkan hiposentrumnya. Untuk meningkatkan optimasi algoritme *Naïve Bayes*, penulis menggunakan algoritme *Adaboost*. Evaluasi terhadap hasil dari algoritme klasifikasi yang telah dioptimasi diperlukan untuk mengetahui tingkat akurasi menggunakan *presicion* dan *recall*. Dalam penelitian ini objek penelitian berupa data gempa bumi di Indonesia yang akan digunakan sebagai data *training* dan data *testing*. Berbagai penelitian yang terkait dengan klasifikasi kekuatan gempa dan lokasi gempa telah banyak dilakukan oleh para peneliti sebelumnya. Sari (2018) menggunakan algoritme K-AP Clustering untuk mengklasifikasikan gempa bumi di Indonesia. Fatichah (2017) mengklasifikasi data gempa berdasarkan data dari twitter menggunakan algoritme Decision Tree, Random Forest dan SVM. Halim (2017) mengelompokkan dampak gempa di Indonesia menggunakan algoritme Kohonen Self Organizing Maps (SOM). Saraswathi (2014) membuat komparasi beberapa algoritme*clustering* untuk mendapatkan akurasi dari algoritme K-Means, DB Scan, Hirarki dan Optic.

## METODE PENELITIAN

### 1. Jenis dan Sumber Data

Jenis data primer yang digunakan dalam penelitian ini adalah data gempa bumi tahun 2017 yang terdiri dari data terjadinya gempa. Sumber data dalam penelitian ini berasal dari Badan Meteorologi, Klimatologi dan Geofisik (BMKG) http://repogempa.bmkg.go.id/.

### 2. Variabel Penelitian

Variabel dalam penelitian ini dibagi menjadi tiga bagian atribut yaitu kedalaman hiposentrum, magnitudo dan lokasi gempa. Yang menjadi data primer dalam proses klasifikasi adalah data kedalaman hiposentrum. Data tersebut akan dibagi menjadi tiga klasifikasi yaitu gempa dalam, gempa menengah dan gempa dangkal. Data primer merupakan data utama yang digunakan sebagai acuan dalam suatu penelitian dan merupakan data yang diolah dalam suatu penelitian. Data

### 3. Analisa Data Penelitian

Data primer yang digunakan dalam penelitian masih berupa data mentah yang belum diolah. Ada sekitar 6090 data yang digunakan dalam penelitian ini. Data mentah tersebut memiliki atribut yang datanya hilang (*missing values*) terutama pada atribut magnitudo. Jumlah atribut yang datanya hilang ada sekitar 3678 data. Data yang demikian perlu diolah atau di*preprocessing*. Untuk melakukan *preprocessing* data menggunakan algoritme normalisasi data. Dataset dibagi menjadi dua bagian yaitu dataset sebagai data pelatihan (*training*) sebanyak 80% dari 6090 data yaitu 4872 data. Sedangkan data uji (*testing*) sebanyak 20% yaitu sebanyak 1218 data. Contoh data gempa terlihat pada Tabel 1.

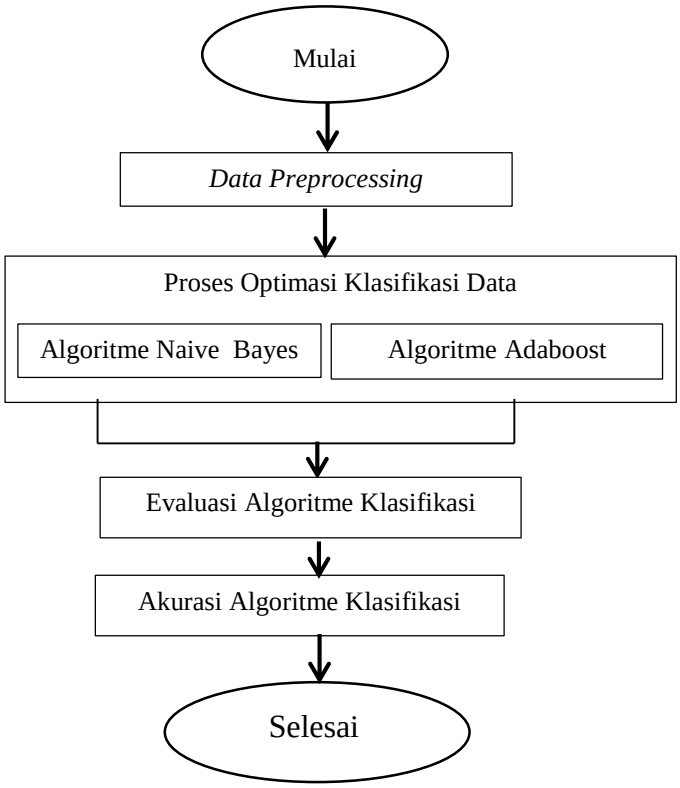
Tabel 1. Contoh data gempa di Indonesia

| *Data ke-* | *Tanggal* | *Waktu (WIB = UTC + 7 Jam)* | *Hipocentrum* | *Mag* | *Type Mag* | *smaj* | *smin* | *az* | *rms* | *Region* |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1/1/2017 | 13:25.9 | 76 | 5 | MLv | 62.78 | 0.96 | 160 | 1.415 | Northern Molucca Sea |
| 2 | 1/1/2017 | 45:54.6 | 74 | 4 | MLv | 1.79 | 0.55 | 158 | 0.601 | Near North Coast of Irian Jaya |
| 3 | 1/1/2017 | 58:13.7 | 10 | 4 | MLv | 7.68 | 1.16 | 210 | 0.894 | Java, Indonesia |
| 4 | 1/1/2017 | 05:01.1 | 533 | 4.2 | mb | 24.47 | 1.13 | 51 | 0.595 | Flores Sea |
| 5 | 1/1/2017 | 39:18.2 | 10 | 4.1 | MLv | 4.95 | 0.87 | 215 | 0.816 | Southwest of Sumatra, Indonesia |
| 6 | 1/1/2017 | 12:09.8 | 66 | 5.1 | mb | 24.41 | 2.07 | 144 | 0.958 | North of Halmahera, Indonesia |
| 7 | 1/1/2017 | 13:39.1 | 32 | 5.5 | Mw(mB) | 84.39 | 2.03 | 126 | 1.342 | North of Halmahera, Indonesia |
| 8 | 1/1/2017 | 17:51.7 | 10 | 4.8 | mb | 17.66 | 2.33 | 262 | 1.238 | North of Halmahera, Indonesia |
| 9 | 1/1/2017 | 13:39.4 | 10 | 4.4 | mb | 12.43 | 2.1 | 242 | 0.545 | North of Halmahera, Indonesia |
| 10 | 1/1/2017 | 27:23.9 | 63 | 1.8 | MLv | 1.71 | 0.44 | 101 | 0.225 | Minahassa Peninsula, Sulawesi |
| 11 | 1/1/2017 | 36:49.8 | 10 | 3.5 | MLv | 4.01 | 1.01 | 213 | 0.416 | South of Bali, Indonesia |
| 12 | 1/1/2017 | 25:16.8 | 15 | 3 | MLv | 2 | 0.95 | 270 | 0.318 | Southwest of Sumatra, Indonesia |
| 13 | 1/1/2017 | 46:40.7 | 21 | 2.9 | MLv | 4.25 | 0.68 | 268 | 0.107 | Sumbawa Region, Indonesia |
| 14 | 1/1/2017 | 01:15.3 | 14 | 2.7 | MLv | 3.27 | 1.43 | 157 | 0.092 | Sulawesi, Indonesia |
| 15 | 1/1/2017 | 18:52.1 | 293 | 3.7 | MLv | 6.76 | 0.47 | 72 | 0.685 | Bali Sea |
| 16 | 1/1/2017 | 36:03.7 | 155 | 4.4 | mb | 9.38 | 1.41 | 210 | 0.851 | North of Halmahera, Indonesia |
| 17 | 1/1/2017 | 45:22.1 | 10 | 3.5 | MLv | 4.09 | 0.1 | 59 | 1.128 | Northern Sumatra, Indonesia |
| 18 | 1/1/2017 | 58:00.8 | 35 | 4.7 | mb | 53.59 | 1.89 | 144 | 0.982 | Off West Coast of |

| *Data ke-* | *Tanggal* | *Waktu (WIB = UTC + 7 Jam)* | *Hipocentrum* | *Mag* | *Type Mag* | *smaj* | *smin* | *az* | *rms* | *Region* |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | Northern Sumatra |
| 19 | 1/2/2017 | 56:50.3 | 10 | 2.9 | MLv | 1.35 | 0.76 | 247 | 0.466 | Southern Sumatra, Indonesia |
| 20 | 1/2/2017 | 52:57.9 | 87 | 4.9 | mb | 16.06 | 1.41 | 79 | 0.819 | Banda Sea |

## 4. Tahapan Penelitian

Berikut tahapan penelitian yang dilakukan oleh penulis, terlihat pada Gambar 1.




Gambar 1. Tahapan Penelitian

a. *Data Preprocessing*

Pada proses ini untuk data yang hilang atau *missing value* perlu dilakukan pembersihan data atau *data cleaning*. Pada *data preprocessing* ini data yang mengalami pengulangan *record* atau redundansi dinormalisasi.

b. Proses Klasifikasi Data

Proses klasifikasi data menggunakan algoritme *Naïve Bayes* dengan rumus

c. Optimasi

Data yang sudah dinormalisasi kemudian diklasifikasi dengan algoritme *Naïve Bayes* dan dioptimasi dengan algoritme *Adaboost*.

Algoritme *Naïve Bayes* terdiri dari beberapa tahap yaitu:

1. Menghitung jumlah kelas yang akan diklasifikasikan
2. Menghitung jumlah kasus dari setiap kelas
3. Kalikan dengan varian variabel yang telah ditentukan

4. Bandingkan hasilnya pada setiap kelas

Konsep dasar dari algoritme *Adaboost* adalah memberikan bobot yang lebih pada klasifikasi yang tidak tepat (*weak classification*). *Boosting* bisa dikombinasi dengan *classifier* algoritme lainnya untuk menambah performa klasifikasi (Listiana, 2017). Tahapan dalam algoritme *Adaboost* (Listiana, 2017) yaitu:

1. Input: Suatu kumpulan *dataset* penelitian dengan label $\{(x_i, y_i)\ (x_N, y_N)\}$, suatu *component learn algoritme*, jumlah perputaran T.
2. Inisialisasi: Bobot $\{w_i\}$ data *training* $w_i^1 = 1/N$ untuk semua i = 1,..., N
3. Iterasi for m = 1,...,M
   a. Gunakan *component learn algoritme* untuk melatih suatu komponen klasifikasi, $y_m$ , pada bobot data pelatihan. Cocokkan *clasifier* $y_m(x_i)$ dengan data pelatihannya untuk meminimalisir bobot fungsi *eror* pada:

$$J_m = \sum_{i=1}^{N} w_i^m \ I, y_m(x_i) \neq t_n \quad (1)$$

   dimana $I(y_m(x_i) \neq t_n)$ sebagai fungsi indikator akan bernilai 1 ketika $y_m(x_i) \neq t_n$ dan 0 untuk $y_m(x_i) = t_n$

   b. Evaluasi kuantitas

$$\varepsilon_m = \frac{\sum_{i=1}^{N} w_i^m \ I, y_m(x_i) \neq t_n}{\sum_{i=1}^{N} w_i^m} \quad (2)$$

   Gunakan persamaan 2. untuk mengevaluasi persamaan 3.

$$\alpha_m = \ln\left\{\frac{1-\varepsilon_m}{\varepsilon_m}\right\} \quad (3)$$

   c. *Update* koefisien bobot

$$w_i^{(m+1)} = \frac{w_i^{(m)} \exp\{\alpha_m \ I(y_m(x_i) \neq t_n)\}}{C_m} \text{ ..... (4)}$$

   dengan $C_m$ adalah konstanta untuk normalisasi

   d. *Output:*

$$Y_M(x) = sign\ (\sum_{m=1}^{M} \alpha_m y_m(x)) \quad (5)$$

d. Evaluasi Algoritme Klasifikasi

Metode klasifikasi *Naïve Bayes* yang sudah dioptimasi pada data hiposentrum gempa dievaluasi menggunakan algoritme *confusion matrix*.

Pada penelitian ini klasifikasi dibagi menjadi tiga kelas maka ada cara menghitung akurasi, presisi dan recall dapat dilakukan dengan menghitung rata-rata dari nilai akurasi, presisi dan recall pada setiap kelas.

$$Akurasi = \frac{TP_i+TN_i}{TP_i+TN_i+FP_i+FN_i} \ x\ 100\% \quad (6)$$

dimana:

- $TP_i$ adalah *True Positive*, yaitu jumlah data positif yang terklasifikasi dengan benar oleh sistem untuk kelas ke-i.
- $TN_i$ adalah *True Negative*, yaitu jumlah data negatif yang terklasifikasi dengan benar oleh sistem untuk kelas ke-i.

- $FN_i$ adalah *False Negative*, yaitu jumlah data negatif namun terklasifikasi salah oleh sistem untuk kelas ke-i.
- $FP_i$ adalah *False Positive*, yaitu jumlah data positif namun terklasifikasi salah oleh sistem untuk kelas ke-i

e. Akurasi AlgoritmeKlasifikasi

Akurasi dari algoritmeoptimasi *Naïve Bayes* menggunakan *precision* dan *recall.*

$$Precision = \frac{\sum_{i=1}^{l} TP_i}{\sum_{i=1}^{l}(FP_i+TP_i)} * 100\% \quad (7)$$

$$Recall = \frac{\sum_{i=1}^{l} TP_i}{\sum_{i=1}^{l}(FP_i+FN_i)} * 100\% \ (8)$$

## HASIL DAN PEMBAHASAN

### 1. Hasil dari *Preprocessing Data* menggunakan proses normalisasi

(*Z-Transformation)* dan *replace missing values*. Normalisasi yang dilakukan adalah normalisasi pada atribut magnitudo dengan hasil normalisasi mean: 4.047947569271616, variance: 113.84787442937836. Untuk hasil *replace missing values* dengan menambahkan data dengan perhitungan rata-rata (*average*) pada atribut magnitudo.

### 2. Hasil klasifikasi gempa bumi dan evaluasi algoritme klasifikasi

Hasil klasifikasi gempa bumi terlihat pada tabel 1 berdasarkan data testing yang diujikan yaitu sebanyak total 1218 data. Gempa dangkal hiposentrumnya kurang dari 60 km, gempa sedang hiposentrumnya antara 60 km sampai 300 km dan gempa dalam memiliki hiposentrum lebih dari 300 km. Akurasi yang dihitung adalah akurasi dari hasil klasifikasi yaitu akurasi gempa dangkal, sedang dan dalam. Algoritme yang digunakan untuk menghitung akurasi menggunakan *algoritme confusion matrix.*
Perhitungan akurasi dari setiap metode *Naïve Bayes* dan *Naïve Bayes + Adaboost* dapat dilihat pada Tabel 2 sampai dengan Tabel 7.

Tabel 2. Akurasi Gempa Dangkal dengan *Naïve Bayes*
Banyak Gempa =214

| | *Predicted False* | *Predicted True* |
|---|---|---|
| Actual False | TN=38 | FP=37 |
| Actual True | FN=29 | TP = 110 |
| Akurasi = | 69% | |

Tabel 3. Akurasi Gempa Sedang dengan *Naïve Bayes*
Banyak Gempa =998

| | *Predicted False* | *Predicted True* |
|---|---|---|
| Actual False | TN=96 | FP=99 |
| Actual True | FN=89 | TP = 714 |
| Akurasi = | 81% | |

Tabel 4. Akurasi Gempa Dalam dengan *Naïve Bayes*
Banyak Gempa 6

| | *Predicted False* | *Predicted True* |
|---|---|---|
| Actual False | TN=1 | FP=2 |
| Actual True | FN=0 | TP = 3 |
| Akurasi = | 67% | |

Tabel 5. Akurasi Gempa Dangkal dengan Optimasi *Naïve Bayes* dan *Adaboost*
Banyak Gempa = 119

| | *Predicted False* | *Predicted True* |
|---|---|---|
| Actual False | TN=23 | FP=13 |
| Actual True | FN=7 | TP = 76 |
| Akurasi = | 83% | |

Tabel 6. Akurasi Gempa Sedang dengan Optimasi *Naïve Bayes* dan *Adaboost*
Banyak Gempa = 1093

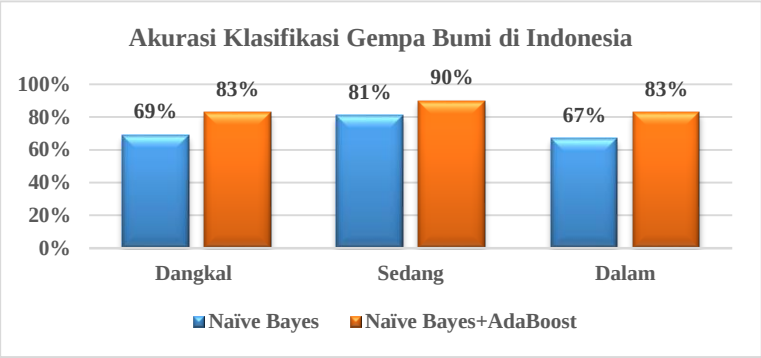
| | *Predicted False* | *Predicted True* |
|---|---|---|
| Actual False | TN =110 | FP = 11 |
| Actual True | FN = 96 | TP = 876 |
| Akurasi = | 90% | |

Tabel 7. Akurasi Gempa Dalam dengan Optimasi *Naïve Bayes* dan *Adaboost*
Banyak Gempa = 6

| | *Predicted False* | *Predicted True* |
|---|---|---|
| Actual False | TN=2 | FP=1 |
| Actual True | FN=0 | TP = 3 |
| Akurasi = | 83% | |

Tabel 8. Jumlah Lokasi Berdasarkan Klasifikasi Hiposentrum Gempa Bumi

| *Algoritme Klasifikasi* | *Gempa Dangkal* | *Akurasi Gempa Dangkal* | *Gempa Sedang* | *Akurasi Gempa Sedang* | *Gempa Dalam* | *Akurasi Gempa Dalam* | *Rata-Rata* |
|---|---|---|---|---|---|---|---|
| *Naïve Bayes* | 214 | 69% | 998 | 81% | 6 | 67% | 72,3% |
| *Naïve Bayes + Adaboost* | 119 | 83% | 1093 | 90% | 6 | 83% | 85,3% |




Gambar 2. Hasil Akurasi Setiap Klasifikasi Gempa

Pada Gambar 2. Disimpulkan bahwa hasil rata - rata akurasi algoritme *Naïve Bayes* sebesar 72,3% dan optimasi algoritme *Naïve Bayes* dan *Adaboost* sebesar 85,3%

## KESIMPULAN DAN SARAN

Kesimpulan dan saran dari penelitian ini adalah Hasil rata - rata akurasi algoritme *Naïve Bayes* sebesar 72,3% dan algoritme *Naïve Bayes* dan *Adaboost* sebesar 85,3%. Dari hasil klasifikasi terlihat bahwa jumlah lokasi yang paling banyak terjadi gempa bumi adalah lokasi gempa bumi sedang sebanyak 1093 lokasi. Penelitian ini juga bisa dikembangkan untuk klasfikasi data gempa yang berpotensi tsunami dan dengan menggunakan algoritme *clustering* untuk mengelompokkan gempa bumi berdasarkan bagian wilayah Indonesia.

## DAFTAR PUSTAKA

Chengsheng, T., Huacheng, L., & Bing, X. (2017). *Adaboost* typical Algorithm and its application research. In *MATEC Web of Conferences* (Vol. 139, p. 00222). EDP Sciences.

Dey, L., Chakraborty, S., Biswas, A., Bose, B., & Tiwari, S. (2016). Sentiment analysis of review datasets using *Naïve Bayes* and k-nn classifier. *arXiv preprint arXiv:1610.09982*.

E. Prasetyo, (2012) *Data Mining: Konsep dan Aplikasi menggunakan Matlab*, 1 ed. Yogyakarta: Andi Offset.

Fatichah, C., & Purwitasari, D. (2017). Deteksi Gempa Berdasarkan Data Twitter Menggunakan Decision Tree, Random Forest, dan SVM. *Jurnal Teknik ITS*, *6*(1), 153-158.

Haixiang, G., Yijing, L., Yanan, L., Xiao, L., & Jinling, L. (2016). BPSO-*Adaboost*-KNN ensemble learning algorithm for multi-class imbalanced data classification. *Engineering Applications of Artificial Intelligence*, *49*, 176-193.

Halim, N. N., & Widodo, E. (2017). Clustering Dampak Gempa Bumi di Indonesia Menggunakan Kohonen Self Organizing Maps (SOM). In *Prosiding SI MaNIs (Seminar Nasional Integrasi Matematika dan Nilai-Nilai Islami)* (Vol. 1, No. 1, pp. 188-194).

Hartuti, E. R. 2009. Buku Pintar Gempa. Yogyakarta: Diva Perss

Listiana, E., & Muslim, M. A. (2017). Penerapan *Adaboost* untuk Klasifikasi Support Vector Machine Guna Meningkatkan Akurasi pada Diagnosa Chronic Kidney Disease. *Prosiding SNATIF*, 875-881.

Nakra, A., & Duhan, M. (2019). Comparative Analysis of Bayes Net Classifier, *Naïve Bayes* Classifier and Combination of both Classifiers using WEKA. *IJ Inf. Technol. Comput. Sci*, *11*, 38-45.

Saraswathi, S., & Sheela, M. I. (2014). A comparative study of various clustering algorithms in data mining. *International Journal of Computer Science and Mobile Computing*, *11*(11), 422-428.

Sari, N. N. (2018). K-Affinity Propagation (K-AP) Clustering Untuk Klasifikasi Gempa Bumi (Studi Kasus: Gempa Bumi di Indonesia Tahun 2017).

Saritas, M. M., & Yasar, A. (2019). Performance Analysis of ANN and *Naïve Bayes* Classification Algorithm for Data Classification. *International Journal of Intelligent Systems and Applications in Engineering*, *7*(2), 88-91.

Terbit *online* pada laman web jurnal :
http://ejournal.amikompurwokerto.ac.id/index.php/telematika/

**Telematika**

Terakreditasi Sinta "3" KEMENRISTEKDIKTI, No. 21/E/KPT/2018

# Implementasi Keamanan Pesan pada Citra Steganografi Menggunakan Modifikasi Cipher Block Chaining (CBC) Vigenere

**Hanifatus Sa'diyah[1], Vera Wati[2], Dony Ariyus[3]**

[1,2,3] Magister Teknik Informatika
Universitas AMIKOM Yogyakarta
Email : hanifputri2013@gmail.com[1], verave.wati@gmail.com[2], dony.a@amikom.ac.id[3]

**I N F O A R T I K E L**

***Sejarah Artikel:***
Menerima 17 Desember 2019
Revisi 1 Februari 2020
Diterima 27 Februari 2020
*Online* 29 Februari 2020

***Keywords:***
*Cryptography,*
*Cipher Block Chaining,*
*Vigenere,*
*LSB,*
*Data Security*

**Kata Kunci:**
*Cryptography,*
*Cipher Block Chaining,*
*Vigenere,*
*LSB,*
Keamanan Data

**Korespondensi:**
Telepon: +62 81229883223
E-mail:
hanifputri2013@gmail.com

***ABSTRACT***

*Internet of Things (IoT) provides easy transportation of data and information, but on the other hand, provides opportunities for cyber-terrorists and attackers to carry out attacks on data and information so that security of data and information is needed. This study aims to combine cryptographic techniques with the classical algorithm that is Vigenere Cipher and modern algorithms that Cipher Block Chaining (CBC), which will be integrated with steganographic techniques Least Significant Bit (LSB) to insert the message information on an object image to provide data security and information more. Is expected to support the various fields of digital watermark and capable of being used in the picture. Testing with 25 times the encryption and decryption process was successfully carried out 18 times and failed seven times, influenced by the size and dimensions of the image. Performance on this algorithm can accommodate both symbols, characters and numbers. However, changes in image size affect the process of decryption and encryption.*

**ABSTRAK**

*Internet of Things* (IoT) menghadirkan kemudahan pertukaran data dan informasi, namun demikian di sisi lain memberikan peluang kepada *cyber-terrorist* dan penyerang untuk melakukan serangan terhadap data dan informasi sehingga pengamanan data dan informasi diperlukan. Penelitian ini bertujuan mengkombinasikan teknik kriptografi dengan algoritme klasik yaitu *Vigenere Cipher* dan algoritme modern yaitu *Cipher Block Chaining* (CBC) yang akan diintegrasikan dengan teknik steganografi *Least Significant Bit* (LSB) untuk menyisipkan pesan informasi di sebuah objek gambar sehingga memberikan keamanan data dan informasi yang lebih tinggi. Diharapkan mampu mendukung layanan berbagai bidang sehingga mampu digunakan digital *watermark* pada gambar. Hasil penelitian yang telah dilakukan menghasilkan visualisasi tidak adanya perbedaan pesan yang belum dan sudah terenkripsi. Pengujian dengan 25 kali proses enkripsi dan dekripsi berhasil dilakukan sebanyak 18 kali dan gagal sebanyak 7 kali, dipengaruhi oleh ukuran dan dimensi citra. Kinerja pada algoritme ini mampu menampung dengan baik simbol, karakter dan angka. Namun perubahan pada *size* gambar berpengaruh ketika proses dekripsi dan enkripsi.

## PENDAHULUAN

*Internet of Things* memiliki potensi besar menawarkan berbagai jenis layanan yang mampu menyelesaikan permasalahan di kehidupan sosial maupun lingkungan bisnis, salah satu jenis layanan yang ditawarkan adalah layanan komunikasi (Atzori, dkk., 2010)(Abomhara dan Koien, 2015). Layanan komunikasi dengan IoT menghadirkan kemudahan pertukaran data dan informasi. Statistik terbaru berdasarkan *International Telecommunication Union* (ITU) mengungkapkan bahwa (ITU, 2017) lebih dari

830 juta pelanggan diseluruh dunia dan 80% populasi dunia memiliki akses ke *internet*. Hal ini membuktikan bahwa IoT memberikan kemudahan berkomunikasi.

Hadirnya IoT memberikan kemudahan berkomunikasi, namun demikian di sisi lain juga menghadirkan ancaman dan memberikan peluang kepada *cyber-terrorist* dan penyerang untuk melakukan serangan (Goutam, 2015). Serangan yang dapat mengancam data dan informasi berupa interupsi, penyadapan informasi, pencurian identitas, pelanggaran hak privasi, penyisipan virus, dan penyisipan data maupun informasi (Ijemaru, dkk., 2018)(Zou, dkk., 2016). Jumlah ancaman semakin meningkat setiap hari dengan jumlah dan kompleksitas yang tinggi (Abomhara dan Koien, 2015). Tidak hanya jumlah penyerang berpotensial yang semakin meningkat, namun jaringan yang semakin meluas dan alat yang tersedia lebih canggih, efektif dan efisien (Kizza, 2013)(Taneja, 2013). Oleh karena itu, keamanan data dan informasi diperlukan dan hal ini menjadi perhatian utama untuk memerangi adanya *cyber crime* karena meluasnya penggunaan *internet* (Zou, dkk., 2016)(Laskar dan Hemachandran, 2012). Keamanan media data dan informasi dapat disisipkan pada sebuah media, yakni dalam format gambar, audio, dan format teks file. Teknik pendekatan ini bisa menggunakan steganografi. Tujuan dari steganografi menjadikan penyembunyian informasi tanpa dicurigai perubahannya. Seperti kasus untuk legitimasi pada gambar dilakukan upaya *watermarking* untuk melindungi keaslian isi informasi dan menghindarkan dari *copyright*.

Kini teknik steganografi bisa lebih ditingkatkan dengan penggunaan kriptografi. Kedua teknik tersebut memiliki tujuan menyembunyikan informasi yang memberikan jaminan akan kerahasiaan dan integritas data (Laskar dan Hemachandran, 2012)(Raphael dan Sundaram, 2010). Steganografi merupakan jenis komunikasi tersembunyi (Laskar dan Hemachandran, 2012)(Li, dkk., 2011) yang bertujuan menyembunyikan informasi di media digital sehingga keberadaan pesan rahasia tidak terdeteksi (Laskar dan Hemachandran, 2012)(Johnson dan Mason, 1998). Teknik kriptografi merupakan teknik mengacak data (Maruf, Riadi dan Prayudi, 2015). Pada teknik kriptografi, struktur pesan akan diacak sedemikian rupa agar pesan tidak memiliki makna dan tidak dapat dipahami(Laskar dan Hemachandran, 2012)(Anderson, 1989). Proses enkripsi akan mengubah informasi dan menjadikan informasi tersebut tidak dapat dibaca (Kester, 2012)(Sinkov, 2009).

Pada dasarnya steganografi dan kriptografi memiliki perbedaan hasil dalam hal teknik penyembunyian data dan informasi. Namun demikian, kedua teknik penyembunyian informasi ini akan saling melengkapi satu sama lain (Laskar dan Hemachandran, 2012). Seberapa baik suatu pesan disembunyikan di dalam media digital, ada kemungkinan pesan tersembunyi itu ditemukan oleh pihak ketiga. Penggabungan steganografi dan kriptografi maka pengamanan pesan yang lebih baik akan tercapai dengan cara menyembunyikan keberadaan pesan yang telah terenkripsi (Laskar dan Hemachandran, 2012)(Raphael dan Sundaram, 2010)(Song dkk., 2011).

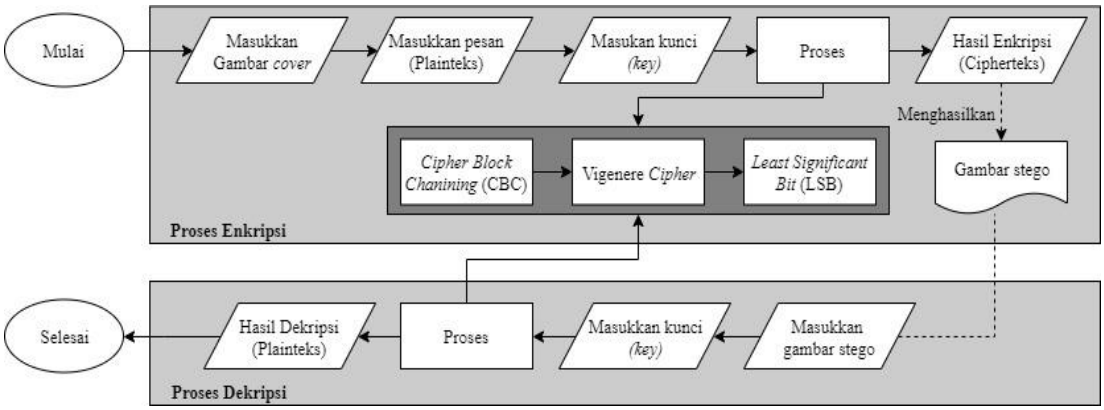
Teknik penerapan penggabungan steganografi dan kriptografi telah dikembangkan pada beberapa penelitian. Penelitian (Permana, 2018), mengamankan pesan teks menggunakan algoritme *Vigenere Cipher*. Proses mengamankan pesan dilakukan dengan cara substitusi, yaitu mengubah setiap huruf menjadi huruf lain berdasarkan kunci yang digunakan. Namun demikian, metode kasiski telah mampu memecahkan enkripsi pesan rahasia algoritme *Vigenere Cipher*. Pemecahan enkripsi ini didasari karena penggunaan kunci yang hanya terdiri dari 26 karakter kunci pada pesan sehingga dapat dengan mudah dipecahkan (Hidayat, Gerhana dan Syaripudin, 2018).

Penelitian melakukan peningkatan keamanan pesan dengan mengkombinasikan algoritme kriptografi klasik dan modern yaitu algoritme *Vigenere Cipher* dan *Cipher Block Chaining* (CBC).

Keunggulan mode operasi CBC adalah pengacakan data biner di dalam blok. Hasil enkripsi blok sebelumnya diumpan-balikkan *(feedback)* ke dalam enkripsi blok *current*, sehingga *cipher* blok yang dihasilkan sepenuhnya tergantung pada semua blok biner dari plainteks (Humendru dan Zebua, 2018). Penelitian ini mengkombinasikan hasil enkripsi *Vigenere Cipher* dan *Cipher Block Chaining* (CBC) dengan algoritme *Least Signicant Bit* (LSB) sehingga menghasilkan proteksi ganda pada keamanan pesan. Kelebihan metode ini pada pengamanan data yang tinggi, sehingga mencegah adanya serangan *stego-attack*, mempertahankan resolusi gambar agar tidak banyak berubah, dan gambar tidak mencurigakan di mata manusia yang akan diabaikan ketika ada pesan rahasia, serta mudah diimplementasikan (Kavitha, dkk., 2012). Maka dalam mendukung layanan kehidupan sosial dalam berbagai bidang, penelitian ini diharapkan mampu digunakan untuk digital *watermark.* Dimana informasi disembunyikan pada format gambar sehingga dilakukan perlindungan *copyright* meskipun dapat diakses dengan *internet* dimanapun dan kapanpun. Namun metode steganografi LSB dalam menyisipkan informasi pada gambar masih bergantung pada resolusi gambar, maka perlu peningkatan metode steganografi untuk menampung lebih banyak karakter pada informasi yang akan disisipkan.

## METODE PENELITIAN

Penelitian ini akan mengkombinasikan algoritme klasik yaitu *Vigenere Cipher* dan algoritme modern *Cipher Block Chainning* (CBC). Alur kinerja dari kombinasi terdapat pada Gambar 1.




Gambar 1 Skema Proses Penyandian dan Penyisipan Pesan

Dijelaskan pada Gambar 1 skema proses penyandian dan penyisipan pesan menggunakan *Vigenere Cipher* dan *Cipher Block Chainning* (CBC) melalui 2 tahapan, yakni :

1. Proses Enkripsi

Pada proses enkripsi dimulai dari penyisipan gambar, kemudian masukan pesan asli (plainteks) dengan beberapa karakter. Pesan asli merupakan informasi yang ingin dilakukan proses enkripsi, masukkan kunci sebagai inisiasi proses. Proses enkripsi melibatkan 3 (tiga) metode, yaitu modifikasi CBC dengan *Vigenere Chiper* untuk proses kriptografi pada pesan dan pendekatan LSB sebagai proses steganografi. Jika berhasil diproses, maka akan menghasilkan informasi acak (cipherteks) dan gambar stego (gambar yang tersisipi pesan).

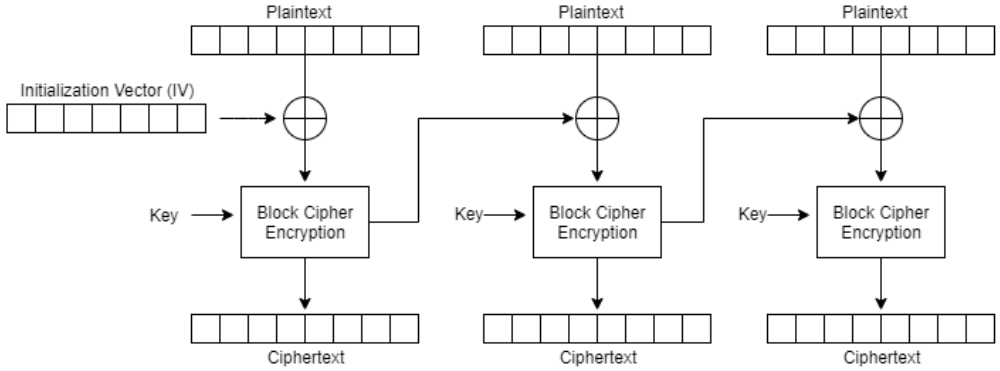
2. Proses Dekripsi

Proses dekripsi menjadi langkah dalam pengembalian pesan yang disisipi pada gambar (gambar stego) untuk kembali ke semula. Proses awal dimulai dari memasukkan gambar stego, kemudian memasukkan kunci penyandian. Dilakukan proses dengan metode yang digunakan sehingga isi pesan asli (plainteks) dapat diketahui.

Ada beberapa metode yang digunakan pada penelitian, yaitu *Cipher Block Chaining* (CBC) digunakan untuk proses kriptografi yang dimodifikasi menggunakan *Vigenere Cipher*. Proses steganografi digunakan pendekatan *Least Significant Bit* (LSB) yakni perubahan dilakukan penyisipan pada bit terakhir pada suatu gambar stego. Metode yang digunakan antara lain :

1. *Cipher Block Chaining* (CBC)

Tahapan proses enkripsi algoritme *Cipher Block Chaining* (CBC) adalah dengan cara meng-XOR-kan blok plainteks dengan *Initialization Vector* (IV). Hasil XOR yang didapat diawal kemudian akan dilakukan XOR kembali dengan menggunakan kunci sehingga menghasilkan cipherteks untuk blok pertama. Cipherteks di blok pertama selanjutkan digunakan sebagai *Initialization Vector* (IV) untuk enkripsi pada blok yang selanjutmya (Dashti, Kheradmand dan Jazi, 2016)(Lestiawan dan Purnama, 2016). Tahapan proses enkripsi dapat dilihat pada Gambar 2.

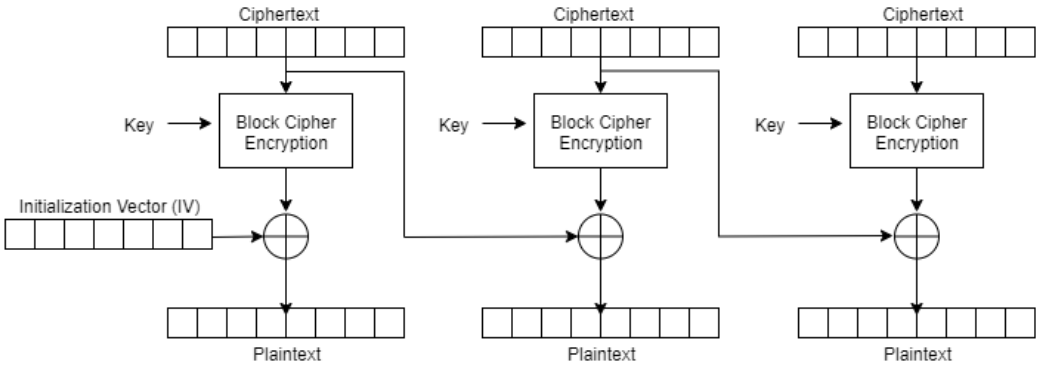



Gambar 2 Proses Enkripsi Algoritme *Cipher Block Chaining* (CBC)

Secara sistematis proses enkripsi CBC dituliskan pada persamaan 1 berikut ini.

$$C_i = E_k\ (P_i \oplus C_{i-1}), C_0 = \text{IV} \qquad (1)$$

Pada proses dekripsi algoritme *Cipher Block Chaining* (CBC) akan dilakukan hal yang sebaliknya dari proses enkripsi. Blok plainteks yang pertama akan didapatkan dengan cara meng-XOR-kan *Initialization Vector* (IV) dengan hasil dekripsi dari blok cipherteks yang pertama (Dashti, Kheradmand dan Jazi, 2016)(Lestiawan dan Purnama, 2016). Ilustrasi proses dekripsi disajikan pada Gambar 3.




Gambar 3 Proses Dekripsi Algoritme *Cipher Block Chaining* (CBC)

Secara sistematis proses dekripsi CBC dituliskan pada persamaan 2 berikut ini.

$$C_i = E_k\ (P_i \oplus C_{i-1}), C_0 = \text{IV} \qquad (2)$$

2. *Vigenere Cipher*

*Vigenere Cipher* dipopulerkan oleh Giovan Bellaso tahun 1553 yang kemudian telah dikembangkan oleh Blaisede Vigenere dengan menggunakan *autokey cipher*. *Vigenere Cipher* untuk proses enkripsi dan

dekripsi akan menggunakan bujur sangkar *Vigenere* (Handoko, dkk., 2019)(Senthil, Prasanthi dan Rajaram, 2013). Enkripsi *Vigenere Cipher* secara matematis dituliskan pada persamaan (3) berikut ini:

$$C_i = (P_i + K_i)\ mod\ 26 \qquad (3)$$

$C_i$ = nilai ascii dari karakter ciphertext ke-$i$

$P_i$ = nilai ascii dari karakter plaintext ke-$i$

$K_i$ = nilai ascii dari karakter kunci ke-$i$

Sedangkan dekripsi *Vigenere Cipher* secara matematis dituliskan pada persamaan (4) berikut ini:

$$P_i = (C_i - K_i)\ mod\ 26 \qquad (4)$$

$C_i$ = nilai ascii dari karakter ciphertext ke-$i$

$P_i$ = nilai ascii dari karakter plaintext ke-$i$

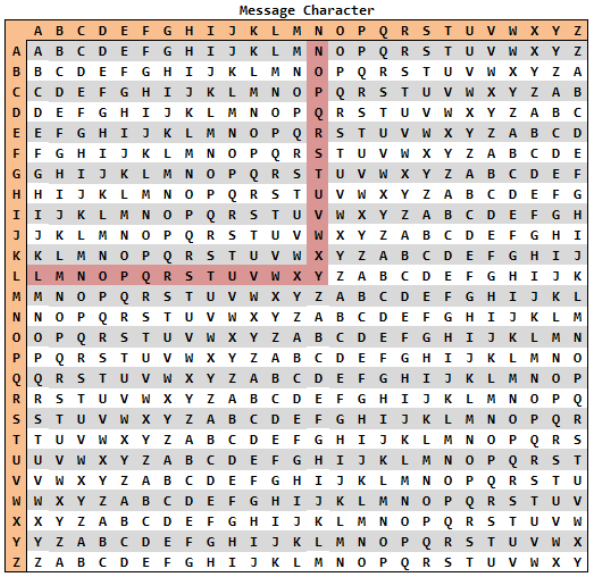
$K_i$ = nilai ascii dari karakter kunci ke-$i$

Dimana nilai desimal karakter : A=0, B=1, C=2, D=3 ... Z=25

Angka module yang digunakan pada persamaan (3)(4) hanya digunakan untuk proses enkripsi dan dekripsi dengan jumlah karakter 26. Jika semua karakter ASCII digunakan untuk proses enkripsi, maka persamaan yang digunakan menggunakan modulo 256.

3. Modifikasi *Cipher Block Chaining* (CBC) dan *Vigenere Cipher*

CBC menjadi algoritme yang melibatkan nilai Inisialisasi Vektor (IV) pada blok *cipher*. Hasil enkripsi sesuai kinerja CBC pada Gambar 2 selanjutnya akan dilakukan proses enkripsi kembali dengan mengadopsi kinerja *Vigenere* dengan menggunakan persamaan (3). Sehingga proses dekripsi pun dilakukan dengan metode yang sama. Penulis memodifikasi mode *Vigenere Cipher* dengan menggunakan tabel yang mengadopsi tabel *Vigenere Cipher* seperti pada Gambar 4.




Gambar 4. Tabel Vigenere Cipher (Wikibooks, 2020)

Pada *"message character"* di isi untuk informasi (plainteks) yang akan di lakukan proses enkripsi, kemudian dicocokkan dengan kunci yaitu "key character" kemudian dilakukan titik temu hingga menghasilkan isi pesan baru dalam bentuk cipherteks. Proses bisa dipercepat dengan persamaan (3)(4) untuk proses enkripsi dan dekripsinya yang dibantu dengan module. Modifikasi penelitian yakni hasil dari CBC kemudian diproses dengan *Vigenere Cipher*. Proses ini sebagai proses kriptografi.

4. *Least Signicant Bit* (LSB)

Teknik steganografi yang umum digunakan adalah metode *Least Significant Bit* (LSB) walaupun umum dan sering digunakan karena kemudahan dalam penerapannya, namun akan sulit jika dikombinasikan dengan teknik kriptografi menggunakan populasi kunci tertentu (Syawal, Fikriansyah dan Agani, 2016). LSB merupakan salah satu algoritme yang digunakan untuk menyembunyikan pesan dalam media digital sehingga pihak lain tidak menyadari bahwa terdapat informasi rahasia dalam gambar tersebut (Song, dkk., 2011). Proses LSB bit dari gambar *cover* di ilustrasikan pada Gambar 5. Misalkan; terdapat karakter JOG dengan nilai biner 01001010 01001111 01000111.

| Biner Gambar Cover | | | | | | | |
|---|---|---|---|---|---|---|---|
| 11100100 | 10000001 | 10110100 | 10001111 | 10111100 | 10110111 | 11100000 | 11001100 |
| 10101010 | 10000110 | 11110111 | 11010100 | 10000000 | 11100101 | 11111111 | 10000011 |
| 11000011 | 11000000 | 11000001 | 11010001 | 10000111 | 11100000 | 11111110 | 11000000 |

↓

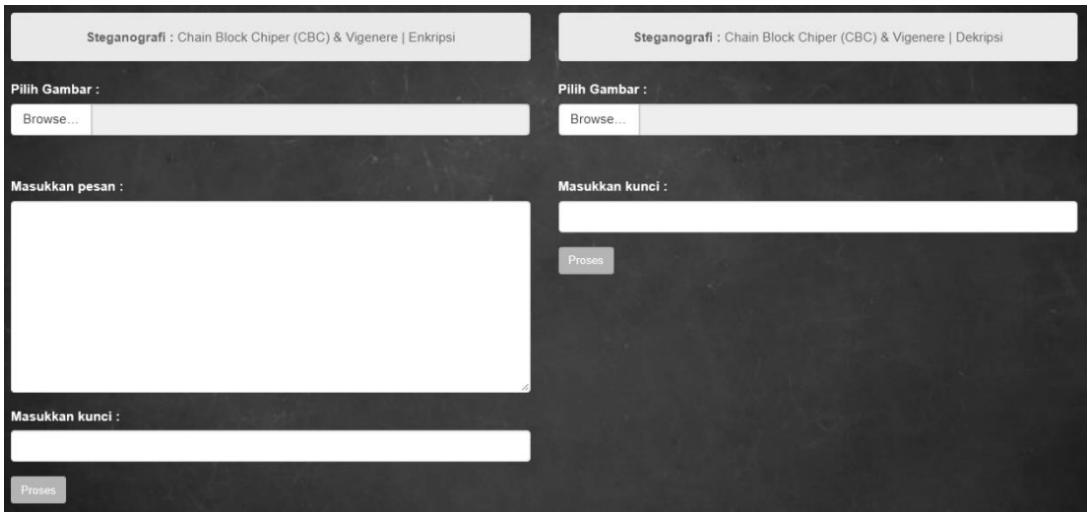
| Biner Gambar Stego | | | | | | | |
|---|---|---|---|---|---|---|---|
| 11100100 | 10000001 | 10110100 | 10001110 | 10111101 | 10110110 | 11100001 | 11001100 |
| 10101010 | 10000111 | 11110110 | 11010100 | 10000001 | 11100101 | 11111111 | 10000011 |
| 11000010 | 11000001 | 11000000 | 11010000 | 10000110 | 11100001 | 11111111 | 11000001 |

Gambar 5 Pengubahan Biner pada Gambar *Cover* ke Gambar Stego

Pada Gambar 5 nilai dari LSB dalam suatu bit terletak pada angka bit paling terakhir, dan merupakan angka yang cocok untuk diganti dengan mengubah nilai bit satu lebih tinggi atau lebih rendah dari nilai sebelumnya (Raphael dan Sundaram, 2010). Dalam penerapan algoritme CBC dan *Vigenere* yang akan diintegrasikan dengan LSB, hasil cipherteks dari CBC dan *Vigenere* diubah ke kode biner berdasarkan tabel ASCII untuk disisipkan bit terakhir pada gambar.

## HASIL DAN PEMBAHASAN

Implementasi pengamanan informasi dengan teknik kriptografi modifikasi CBC dan vigenere yang kemudian dilanjutkan dengan teknik steganografi menggunakan mode Least Signicant Bit (LSB) disajikan pada Gambar 6.




Gambar 6 Desain Antar Muka Kriptografi Modifikasi CBC dan Vigenere dengan Teknik Steganografi

1. Skenario Pengujian

a. Pengujian Visual

Pengujian visual digunakan untuk memberikan perbandingan pada cover image dan stego image setelah dilakukan enkripsi maupun sebelum dilakukan enkripsi secara kasat mata. Implementasi teknik steganografi memberikan pengamanan pesan agar tidak terdeteksi secara kasat mata(Tiwari, Yadav and Mittal, 2014), sehingga diperlukan skenario pengujian visual.

Tabel 1 Skenario Pengujian Visual 1

| | | |
|---|---|---|
| Gambar Stego | : |  |
| Pesan Rahasia | : | Belajar Kriptografi CBC Vigenere |
| Gambar *Cover* | : |  |

Tabel 2 Skenario Pengujian Visual 2

| | | |
|---|---|---|
| Gambar *Cover* | : |  |
| Pesan Rahasia | : | Local Wisdom Indonesia is gudeg |
| Gambar Stego | : |  |

Tabel 3 Skenario Pengujian Visual 3

| | | |
|---|---|---|
| Gambar *Cover* | : |  |
| Pesan Rahasia | : | Kesultanan Ngayogyakarta Hadiningrat berada di Jogja |
| Gambar Stego | : |  |

Tabel 1,2,3 menunjukkan hasil pengujian visual secara kasar mata pada gambar *cover* dan gambar stego yang menunjukkan tidak ada perbedaan pada hasil enksipsi gambar. Hasil gambar stego yang menunjukkan tidak adanya perbedaan dengan gambar *cover* mengacu pada penelitian yang memberikan perubahan keduanya tidak terlalu mencolok.

b. Pengujian Kapasitas Penyisipan Gambar

Kelebihan teknik steganografi adalah pada kapasitas data yang berhasil disisipkan atau disimpan pada gambar *cover*. Pengujian kapasitas penyisipan dilakukan untuk mengetahui seberapa besar kapasitas pesan yang dapat disisipkan pada gambar *cover*. Hal inilah yang mendasari perlunya dilakukan pengujian pada kapasitas pennyisipan pesan. Hasil pengujian kapasitasn penyisipan disajikan pada Tabel 4. berikut ini.

Tabel 4 Hasil Pengujian Kapasitas Penyisipan Gambar

| No | Info Enkripsi | | | Info Embedd | | | Proses | |
|---|---|---|---|---|---|---|---|---|
| | Nama Gambar | Kunci | Karakter | Ukuran Awal | Ukuran Akhir | Tipe File | Enkripsi | Dekripsi |
| 1 | Doraemon.jpg | Amikom | The Avengers is a 2012 American superhero film | 7,391 bytes | - | .png | Gagal | Gagal |
| | Doraemon.jpg | Amikom | 2012567653 | 7,391 bytes | 53,737 bytes | .png | Berhasil | Berhasil |
| | Doraemon.jpg | Amikom | !@#$%%$#^ | 7,391 bytes | 53,739 bytes | .png | Berhasil | Berhasil |
| | Doraemon.jpg | Amikom | The Avengers is !@# American superhero film | 7,391 bytes | - | .png | Gagal | Gagal |
| | Doraemon.jpg | Amikom | The Avengers !@# American | 7,391 bytes | 53,818 bytes | .png | Berhasil | Berhasil |
| | Doraemon.jpg | Amikom | The Ave ngers !@# 12 American | 7,391 bytes | 53,836 bytes | .png | Berhasil | Berhasil |
| | Doraemon.jpg | Indonesia jaya | The Avengers !@# 12 American | 7,391 bytes | 53,839 bytes | .png | Berhasil | Berhasil |
| | Doraemon.jpg | Indonesia jaya | The Avengers !@# 12 American film | 7,391 bytes | - | .png | Gagal | Gagal |
| | Doraemon.jpg | !@#12 | The Avengers !@# 12 Americanfilm | 7,391 bytes | 53,844 bytes | .png | Berhasil | Berhasil |
| | Doraemon.jpg | !@#12 | The Avengers !@# 12 Americansuperherofilm | 7,391 bytes | - | .png | Gagal | Gagal |
| | | | | | | | | |
| 2 | Lena.jpg | Amikom | The Avengers is a 2012 American superhero film | 11,782 bytes | - | .png | Gagal | Gagal |
| | Lena.jpg | Amikom | The Avengers is a 2012 American superhero | 11,782 bytes | 103,511 bytes | .png | Berhasil | Berhasil |
| | Lena.jpg | Amikom | The Avengers is a 2012 American superherofilm | 11,782 bytes | 103,529 bytes | .png | Berhasil | Berhasil |
| | Lena.jpg | Amikom | 201256235712391723 | 11,782 bytes | 103,455 bytes | .png | Berhasil | Berhasil |
| | Lena.jpg | Amikom | !@#$%^&*(__+){\|":?>< | 11,782 bytes | 103,448 bytes | .png | Berhasil | Berhasil |
| | Lena.jpg | amikom unggul | 201256235712391723 | 11,782 bytes | 103,455 bytes | .png | Berhasil | Berhasil |
| | Lena.jpg | 12345 | !@#$%^&*(__+){\|":?><?><:"><}(*&^%^&*^^*&^^^^%$<>:?":{}+ | 11,782 bytes | - | .png | Gagal | Gagal |
| | Lena.jpg | 12345 | !@#$%^&*(__+){\|":?><?><:"><}(*&^%^&*^^*&^^^^%$<>:?":{}+ | 11,782 bytes | - | .png | Gagal | Gagal |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | !@###@@$$%^>?:"\|<br>){+_}(*&&*_ | | | | | |
| | Lena.jpg | @#$%^<br>& | !@#$%^&*(__+){\|":<br>?><&**^$ | 11,782<br>bytes | 103,479<br>bytes | .png | Berhasil | Berhasil |
| | | | | | | | | |
| 3 | Monas.jpg | Amikom | The Avengers is a<br>2012<br>American superhero<br>film | 20,246<br>bytes | 142,034<br>bytes | .png | Berhasil | Berhasil |
| | Monas.jpg | Amikom | 20125623571239172<br>3 | 20,246<br>bytes | 141,894<br>bytes | .png | Berhasil | Berhasil |
| | Monas.jpg | Amikom | !@#$%^&*(__+){\|":<br>?>< | 20,246<br>bytes | 141,920<br>bytes | .png | Berhasil | Berhasil |
| | Monas.jpg | Amikom<br>unggul | 20125623571239172<br>3 | 20,246<br>bytes | 141,889<br>bytes | .png | Berhasil | Berhasil |
| | Monas.jpg | 12345 | !@#$%^&*(__+){\|":<br>?><?><:"><}<br>(*&^%^&*^^*&^^^^<br>%$<>:?":{}+ | 20,246<br>bytes | 142,088<br>bytes | .png | Berhasil | Berhasil |
| | Monas.jpg | @#$%^<br>& | !@#$%^&*(__+){\|":<br>?><?><:"><}<br>(*&^%^&*^^*&^^^^<br>%$<>:?":{}+<br>!@###@@$$%^>?:"\|<br>){+_}(*&&*_<br>!@###@@$$%^>?:"\|<br>){+_}(*&&*_ | 20,246<br>bytes | 142,314<br>bytes | .png | Berhasil | Berhasil |

Pada pengujian penyisipan pesan pada 25 kali masing-masing pada proses enkripsi dan dekripsi, telah ditemukan keberhasilan sebanyak 18 kali pada proses enkripsi dan dekripsi dan kegagalan proses sebanyak 7 kali. Pada pengujian ini modifikasi CBC *Vigenere* yang di sisipkan pada citra steganografi menemukan beberapa temuan yaitu sistem ini mampu membaca pesan karakter, angka dan simbol. Hasil inputan ketika proses enkripsi mampu memanggil citra yang berwarna dengan format .jpg namun hasil enkripsi dijadikan format .png dan dan citra tetap terjaga warnanya. Jumlah pesan, kunci dan ukuran citra berpengaruh pada proses enkripsi dan dekripsi karena dipengaruhi oleh file gambar yang terlalu kecil *size* sekitar sampai 10 KB dengan nilai lebar dan tinggi (dimensi) gambar yang terlalu kecil. Namun sejauh ini kinerja dari metode ini berkerja dengan baik dalam proses enkripsi dan dekripsi.

c. Pengujian Enkripsi dan *Embedding*

Pengujian enkripsi dan embedding dilakukan untuk menguji keberhasilan sistem dalam melakukan penyembunyian informasi dengan menggunakan modifikasi teknik kriptografi CBC dan *Vigenere Cipher* yang diintegrasikan dengan teknik steganografi *Least Signicant Bit* (LSB). Tabel 5 menunjukkan hasil pengujian enkripsi dan embedding pada sistem.

Tabel 5 Hasil Pengujian Enkripsi dan Embedding

| No | Info Enkripsi | | | Info Embedd | | | Proses | |
|---|---|---|---|---|---|---|---|---|
| | Nama Gambar | Kunci | Jumlah Pesan | Ukuran Awal | Ukuran Akhir | Tipe File | Enkripsi | *Embedd* |
| 1 | Doraemon.jpg | Amikom | 4 kata | 7,391 bytes | 53,818 bytes | .png | Berhasil | Berhasil |
| 2 | Lena.jpg | Amikom | 4 word | 13,459 bytes | 103,458 bytes | .png | Berhasil | Berhasil |
| 3 | Monas.jpg | Jawa | 10 word | 20,246 bytes | 142,057 bytes | .png | Berhasil | Berhasil |

Tabel 5. menunjukkan bahwa setiap kunci yang digunakan untuk skenario uji enkripsi dan *embed* berhasil. Jumlah pesan yang disisipkan memiliki perbedaan satu dengan yang lainnya, hal ini bergantung pada ukuran bit citra yang digunakan. File gambar sebelum dilakukan enkripsi memiliki tipe file .jpg dan

setelah dilakukan enkripsi tipe file akan berubah menjadi .png dengan ukuran gambar yang berbeda sesuai dengan jumlah pesan dan kunci yang digunakan.

d. Pengujian Dekripsi dan *Extracting*

Pengujian dekripsi dan *extracting* dilakukan untuk menguji keberhasilan sistem dalam melakukan pengembalian pesan informasi rahasia menggunakan modifikasi teknik kriptografi CBC dan *Vigenere Cipher* yang diintegrasikan dengan teknik steganografi *Least Signicant Bit* (LSB). Tabel 6 menunjukkan hasil pengujian dekripsi dan *extracting* pada sistem.

Tabel 6 Hasil Pengujian Dekripsi dan Extracting

| No | Info Extracting | | | Info Dekripsi | | | | Proses | |
|---|---|---|---|---|---|---|---|---|---|
| | Nama Gambar | Kunci | Jumlah Pesan Awal | Ukuran Awal (bytes) | Ukuran Akhir (bytes) | Tipe File | Jumlah Pesan Akhir | Dekripsi | *Extract* |
| 1 | Doraemon.jpg | Amikom | 7 kata | 7,391 | 53,922 | .png | 7 kata | Berhasil | Berhasil |
| 2 | Lena.jpg | jogja | 4 word | 13,459 | 103,458 | .png | 4 word | Berhasil | Berhasil |
| 3 | Monas.jpg | jawa | 10 word | 20,246 | 142,057 | .png | 10 word | Berhasil | Berhasil |

Berdasarkan pengujian yang dilakukan pada Tabel 6. menunjukkan bahwa dekripsi dan extracting yang dilakukan oleh sistem berhasil dalam melakukan pengembalian pesan informasi rahasia menggunakan modifikasi teknik kriptografi cipher block chaining dan vigenere cipher yang diintegrasikan dengan teknik steganografi Least Signicant Bit (LSB). Jumlah pesan tidak ada yang berkurang maupun terpotong, pesan dapat dikembalikan tanpa ada perubahan apapun.

## KESIMPULAN DAN SARAN

Berdasarkan pada pembahasan hasil dan pengujian yang telah dilakukan pada penelitian ini, dapat ditemukan beberapa hasil, yaitu:

1. Pengujian dengan visual menghasilkan secara penglihatan mata tidak adanya perbedaan pada gambar baik yang belum disisipi pesan maupun sudah.
2. Penyisipan pesan pun dilakukan untuk menguji kinerja dari kriptografi dan penyisipan pada citranya. Hasil tersebut membuktikan jika proses yang dilakukan sebanyak 25 kali menemukan keberhasilan sebanyak 18 kali dan gagal sebanyak 7 kali. Hal tersebut dipengaruhi jumlah pesan, kunci dan ukuran citra berpengaruh pada proses enkripsi dan dekripsi karena dipengaruhi oleh file gambar yang terlalu kecil *size* sekitar sampai ±10 KB dengan nilai lebar dan tinggi (dimensi) gambar yang terlalu kecil. Namun penyisipan simbol dan karakter sudah dapat dilakukan dengan metode CBC Vigenere. Ketika enkripsi gambar memiliki pengubahan *size* dan proses dekripsi teknik Steganografi *Least Significant Bit* (LSB) mampu disisipi pesan dan dapat mengembalikan pesan ke semula.

## DAFTAR PUSTAKA

Abomhara, M. and Koien, G. (2015) 'Cyber Security and the *Internet* of Things : Vulnerabilities , Threats , Intruders Cyber Security and the *Internet* of Things : Vulnerabilities , Threats , Intruders', *Journal of Cyber Security*, 4(May), pp. 65–88. doi: 10.13052/jcsm2245-1439.414.

Anderson, R. (1989) 'Cryptanalytic Properties Of Short Substitution', *Taylor & Francis*, XIII(1), pp. 61–72. doi: 10.1080/0161-118991863772.

Atzori, L., Iera, A. and Morabito, G. (2010) 'The *Internet* of Things : A survey', *Computer Networks ELSEVIER*. Elsevier B.V., 54(15), pp. 2787–2805. doi: 10.1016/j.comnet.2010.05.010.

Dashti, A., Kheradmand, H. A. and Jazi, M. D. (2016) 'Comparison Of Three Modes Of Cryptography

Operation For Providing Security and Privacy Based on Important Factors', *Information Technology & Electrical Engineering*, 5(3), pp. 7–12.

Goutam, R. K. (2015) 'Importance of Cyber Security', *International Journal of Computer Applications*, 111(7), pp. 14–17.

Handoko, L. B. *et al.* (2019) 'Digital Signature Pada Citra Menggunakan Rsa Dan Vigenere Cipher Berbasis Md5', *SIMETRIS*, 10(1), pp. 357–366.

Hidayat, M. H., Gerhana, Y. A. and Syaripudin, U. (2018) 'Kombinasi Algoritme Kriptografi Vigenere Chiper dan Hill Cipher untuk Penyandian Pesan Rahasia pada Metode Steganografi', *INSIGHT*, 1(1), pp. 125–131.

Humendru, F. and Zebua, T. (2018) 'Implementation of Triple Transposition Vegenere Cipher Algorithm and Cipher Block Chaining for Encoding Text', *International Journal of Informatics and Computer Science*, 2(1), pp. 26–31.

Ijemaru, G. K. *et al.* (2018) 'Security Challenges of Wireless Communications Networks : A Survey Security Challenges of Wireless Communications Networks : A Survey', *International Journal of Applied Engineering Research*, 13(8), pp. 5680–5692.

ITU (2017) 'The world in 2017: ICT facts and figures', *International Telecomummication Union*, July.

Johnson, N. F. and Mason, G. (no date) 'Exploring Steganography: Seeing the Unseen', *IEEE, Computing Practices*, 31(2), pp. 26–34.

Kavitha *et al.* (2012) 'Steganography Using Least Signicant Bit Algorithm', *International Journal of Engineering Research and Applications (IJERA)*, 2(3), pp. 338–341.

Kester, Q. (2012) 'A Cryptosystem Based on Vigenère Cipher with Varying Key', *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 1(10), pp. 108–113.

Kizza, J. M. (2013) *Guide to Computer Network Security*. Springer.

Laskar, S. A. and Hemachandran, K. (2012) 'Secure Data Transmission Using Steganography And Encryption', *International Journal on Cryptography and Information Security (IJCIS)*, 2(3), pp. 161–172. doi: 10.5121/ijcis.2012.2314.

Lestiawan, H. and Purnama, R. D. O. (2016) 'Pengamanan Dokumen Teks Menggunakan Algoritme Kriptografi Mode Operasi Cipher Block Chaining ( CBC ) Dan Steganografi Metode End Of File ( EOF )', *Techno.com*, 15(1), pp. 22–31.

Li, B. *et al.* (2011) 'A Survey on Image Steganography and Steganalysis', *Journal of Information Hiding and Multimedia Signal Processing*, 2(2), pp. 142–172.

Maruf, F., Riadi, I. and Prayudi, Y. (2015) 'Merging of Vigenére Cipher with XTEA Block Cipher to Encryption Digital Merging of Vigenére Cipher with XTEA Block Cipher to Encryption Digital Documents', *International Journal of Computer Applications*, 132(1), pp. 27–33. doi: 10.5120/ijca2015907262.

Permana, A. A. (2018) 'Penerapan Kriptografi Pada Teks Pesan dengan Menggunakan Metode', *Jurnal Al-Azhar Indonesia Seri Sains Dan Teknologi*, 4(3), pp. 110–115.

Raphael, A. J. and Sundaram, D. V. (2010) 'Cryptography and Steganography – A Survey', *Int. J. Comp. Tech. Appl*, 2(3), pp. 626–630.

Senthil, K., Prasanthi, K. and Rajaram, R. (2013) 'A Modern Avatar of Julius Ceasar and Vigenere Cipher', *IEEE International Conference on Computational Intelligence and Computing Research*, pp. 13–15. doi: 10.1109/ICCIC.2013.6724170.

Sinkov, A. (2009) *Elementary Cryptanalysis: A Mathematical Approach*. Second Edi. United States of America: The Mathematical Association of Amerika.

Song, S. *et al.* (2011) 'A Novel Secure Communication Protocol Combining Steganography and Cryptography', *Elsevier Inc, Advanced in Control Engineering and Information Science*, 15, pp. 2767–2772. doi: 10.1016/j.proeng.2011.08.521.

Syawal, M. F., Fikriansyah, D. C. and Agani, N. (2016) 'Implementasi Teknik Steganografi Menggunakan

Algoritme Vigenere Cipher Dan Metode LSB', *Jurnal TICOM*, 4(3), pp. 91–99.

Taneja, M. (2013) 'An Analytics Framework to Detect Compromised IoT Devices using Mobility Behavior', *International Conference on ICT Convergence (ICTC) IEEE*, pp. 38–43.

Tiwari, A., Yadav, S. R. and Mittal, N. K. (2014) 'A Review on Different Image Steganography Techniques', *International Journal of Engineering and Innovative Technology (IJEIT)*, 3(7), pp. 121–124.

Wikibooks (2020) *Visual Basic for Applications*.

Zebua, T. (2015) 'Pengamanan Data Teks Dengan Kombinasi Cipher Block Chaining dan LSB-1', *Seminar Nasional Inovasi dan Teknologi (SNITI)*, 2015(September), pp. 85–89. Available at: sniti.info.

Zou, Y. *et al*. (2016) 'A Survey on Wireless Security : Technical Challenges , Recent Advances , and Future Trends', *Proceedings of the IEEE*, 104(9), pp. 1727–1765.

Terbit *online* pada laman web jurnal :
http://ejournal.amikompurwokerto.ac.id/index.php/telematika/

**Telematika**

Terakreditasi Sinta "3" KEMENRISTEKDIKTI, No. 21/E/KPT/2018

UNIVERSITAS AMIKOM PURWOKERTO

# Implementasi Data Mining Menggunakan Algoritme *Naive Bayes Classifier* dan *C4.5* untuk Memprediksi Kelulusan Mahasiswa

**Endang Etriyanti[1], Dedy Syamsuar[2] dan Yesi Novaria Kunang[3]**

[1,2,3]Program Studi Magister Teknik Informatika, Fakultas Ilmu Komputer
Universitas Bina Darma
Email : endang.etriyanti@gmail.com[1], dedy_syamsuar@binadarma.ac.id[2],
yesinovariakunang@binadarma.ac.id[3]

**INFO ARTIKEL**

***Sejarah Artikel:***
Menerima 8 Agustus 2019
Revisi 8 Oktober 2019
Diterima 24 Februari 2020
*Online* 28 Februari 2020

***Keywords*:**
*Naïve Bayes Classifier*
*C4.5 Algorithm*
*Student Graduation*
*RapidMiner*

**Kata kunci:**
*Naive Bayes Classifier*
Algoritme *C4.5*
Kelulusan Mahasiswa
RapidMiner

**Korespondensi:**
Telepon: +6281996312599
E-mail:
endang.etriyanti@gmail.com

## *ABSTRACT*

*The inability of students to complete their studies on time is faced by most of higher education institution. STMIK Bina Nusantara Jaya Lubuklinggau is one of those which is experienced with this matter. In most cases, the students could complete their studies longer than the expected duration. From 162 students of Sistem Informasi study program in the year 2013 and 2014, there were 117 students completed their studies on time, while 45 students were late. As a result, it could prevent new students from joining the institution since the limited student capacity. This study deploys data mining technique in predicting the graduation status of students on time. First, preprocessing is used to obtain a good dataset. Secondly, the data is processed to obtain a set of prediction. In this step, two mining algorithm were applied – Naive Bayes classifier and C4.5 algorithm to be knowing the performance of the two methods, the method has a greater accuracy value will be recommended to solving the problem of prediction of students graduation at STMIK Bina Nusantara Jaya Lubuklinggau. Thirdly, the result then was validated using K-Fold Cross Validation technique. Finally, the Confusion Matrix is deployed to ensure the accuracy of the prediction. The results indicate that the C4.5 Algorithm method can be used to predict student graduation status with an accuracy rate of 79,08% while the accuracy rate of the Naive Bayes Classifier method is only 78,46%. The dominant factor is IPK-S4 variable.*

## ABSTRAK

Ketidakmampuan mahasiswa untuk menyelesaikan studi tepat waktu dialami oleh sebagian besar Lembaga Pendidikan Tinggi. STMIK Bina Nusantara Jaya Lubuklinggau adalah salah satu perguruan tinggi yang mengalami hal tersebut. Dalam banyak kasus para mahasiswa menyelesaikan studi mereka lebih lama dari rentang waktu yang diharapkan. Dari 162 mahasiswa program studi Sistem Informasi tahun angkatan 2013 dan 2014 terdapat 117 mahasiswa yang menyelesakan studinya tepat waktu, sedangkan 45 mahasiswa terlambat. Akibatnya hal tersebut dapat menghambat mahasiswa baru untuk bergabung dengan lembaga karena kapasitas mahasiswa yang terbatas. Penelitian ini menggunakan teknik data mining dalam memprediksi status kelulusan mahasiswa. Pertama, preprocessing digunakan untuk mendapatkan dataset yang berkualitas. Kedua, data diproses untuk mendapatkan serangkaian prediksi. Pada langkah ini, dua Algoritme data mining diterapkan - Algoritme Naive Bayes Classifier dan Algoritme C4.5 dengan tujuan untuk mengetahui kinerja dari kedua algoritme dengan tingkat akurasi yang lebih besar akan direkomendasikan untuk menyelesaikan masalah prediksi kelulusan mahasiswa pada STMIK Bina Nusantara Jaya Lubuklinggau. Ketiga, hasilnya kemudian divalidasi menggunakan teknik K-Fold Cross Validation. Terakhir, Coffusion Matrix digunakan untuk memvalidasi nilai akurasi hasil prediksi. Hasil penelitian menunjukkan bahwa metode Algoritme C4.5 dapat digunakan untuk memprediksi status kelulusan mahasiswa dengan tingkat akurasi 79,08%

sedangkan metode Naive Bayes Classifier hanya 78,46%. Dengan faktor dominan adalah variabel IPK-S4.

**PENDAHULUAN**

Mahasiswa merupakan aspek penting yang harus diperhatikan dengan serius dalam evaluasi program studi. Salah satu indikator keberhasilan program studi dapat dilihat dari lama studi mahasiswa. Lama studi mahasiswa adalah rentang waktu bagi mahasiswa untuk menyelesaikan studinya. Selain itu lama studi mencerminkan tingkat pencapaian mahasiswa dalam studinya. Dalam persfektif yang lebih luas rata-rata lama studi mahasiswa mempengaruhi kualitas program studi dan oleh karena itu lama studi mahasiswa dijadikan salah satu kriteria untuk menentukan penilaian akreditasi oleh Badan Akreditasi Nasional Perguruan Tinggi (Zainuddin, 2019). Untuk alasan ini setiap lembaga pendidikan perlu memberikan perhatian terhadap lama studi mahasiswa.

Ketidakmampuan mahasiswa untuk menyelesaikan studi tepat waktu dihadapai oleh sebagian besar lembaga pendidikan tinggi. STMIK Bina Nusantara Jaya Lubuklinggau adalah salah satu perguruan tinggi yang mengalami hal tersebut. Dalam banyak kasus, para mahasiswa menyelesaikan studi mereka lebih lama dari rentang waktu yang diharapkan. Dari 162 data mahasiswa program studi Sistem Informasi tahun angkatan 2013 dan 2014 terdapat 117 mahasiswa yang dapat menyelesaikan studinya tepat waktu sedangkan 45 mahasiswa tidak tepat waktu atau terlambat. Akibatnya hal tersebut dapat menghambat mahasiswa baru untuk bergabung dengan lembaga karena kapasitas mahasiswa yang terbatas. Untuk itu daya tampung mahasiswa baru dan lama studi mahasiswa perlu diperhatikan (Bisri, 2015). Sehingga untuk mengantisipasi hal tersebut maka prediksi perlu dilakukan untuk mengetahui status kelulusan mahasiswa. Jika status kelulusan mahasiswa dapat diprediksi, maka bagian program studi perlu memberi perhatian serius kepada mahasiswa yang diprediksi terlambat untuk dapat meningkatkan IPK pada setiap semester agar dapat dapat menyelesaikan studinya sesuai rentang waktu yang diharapkan.

Prediksi menurut Salmu & Solichin (2017) merupakan proses keilmuan untuk mendapatkan *knowledge* secara berurutan berdasarkan bukti-bukti. Ada berbagai macam cara untuk menyelesaikan masalah prediksi, salah satunya adalah teknik penambangan data (*data mining*). Teknik *data mining* merupakan cara yang mudah dan relatif cepat untuk memperoleh pengetahuan secara otomatis (Suyanto, 2017) dan pengetahuan abstrak dari sebuah *database* yang besar (Mulya, 2019) yang meliputi bentuk dan/atau hubungan antar data. Data mining menurut Juliansa (2019) adalah proses untuk mendapatkan ilmu pengetahuan dari sebuah informasi yang berasal dari gudang basis data.

Bagian penting dalam *data mining* adalah teknik klasifikasi, yaitu cara yang digunakan untuk mempelajari data set agar didapatkan hubungan antar data yang membentuk *pattern* (pola) sehingga dapat diperoleh *knowledge.* Ada banyak metode *data mining* yang bisa diterapkan untuk klasifikasi. Algoritme yang populer antara lain *Artificial Neural Network,* Algoritme *C4.5*, *Nearest Neighbour Rule, Fuzzy Logic, Naive Bayes, K-Mean, Support Vector Machine,* dan lain-lain. Penelitian yang mengangkat topik tentang klasifikasi dan penerapan *data mining* telah banyak dilakukan sebelumnya (Prakoso & Tutik, 2017; Anam & Santoso, 2018; Amalia, 2017; Risqiati & Ismanto, 2017; Septiani, 2017; Zainuddin, 2019).

Beberapa penelitian sebelumnya juga mengukur tingkat akurasi masing-masing metode *data mining.* Penelitian Anam & Santoso (2018) membandingkan kinerja antara Algoritme *Naive Bayes Classifier* dengan Algoritme C4.5 dalam mengklasifikasikan data penerima beasiswa. Temuan dari penelitian tersebut menunjukan tingkat akurasi Algoritme C4.5 (96, 40%) lebih baik dibandingkan dengan *Naive Bayes*

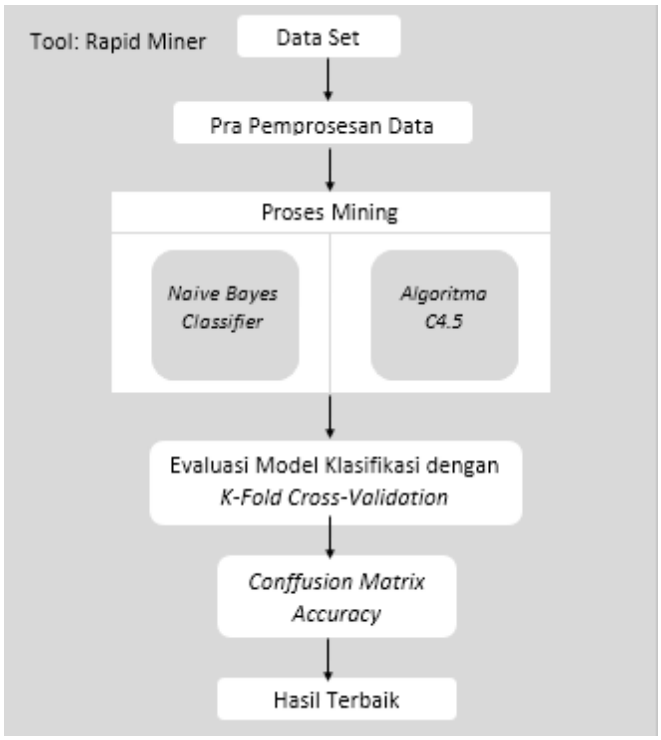
*Classifier* (95,11%). Hasil yang sama diperoleh pada penelitian selanjutnya (Prakoso & Tutik, 2017; Risqiati & Ismanto, 2017; Zainuddin, 2019) dimana tingkat akurasi C4.5 lebih baik dibandingkan dengan *Naive Bayes Classifier* dengan perbedaan berkisar antara 1-7%. Namun, hasil yang berbeda menjadi temuan dari Septiani (2017) dan Amalia (2017). Pada penelitiannya, Septiani (2017) memprediksi penyakit hepatitis, metode yang digunakan adalah komparasi metode Algoritme C4.5 dan *Naive Bayes Classifier*, dengan hasil penelitian *Naive Bayes Classifier* memiliki nilai akurasi 83,71% dan Algoritme C4.5 yaitu 77,29%. Amalia (2017) membandingkan metode *data mining* untuk memprediksi proses bersalin seorang ibu dengan menggunakan tiga metode yaitu *Neural Network*, *Naive Bayes Classifier* dan Algoritme C4.5. Secara berturut-turut diperoleh tingkat akurasi sebesar 93%, 94% dan 90%.

Berdasarkan uraian diatas, penelitian ini bertujuan untuk melakukan prediksi kelulusan mahasiswa STMIK Bina Nusantara Jaya dengan 2 metode yaitu *Naive Bayes Classifier* dan Algoritme C4.5. Data yang digunakan pada penelitian ini berjumlah 162 data mahasiswa program studi Sistem Informasi tahun angkatan 2013 dan 2014 yang sudah lulus. Secara teoritis penelitian ini berkontribusi dalam penerapan metode data mining untuk memprediksi kelulusan seorang mahasiswa. Manfaat selanjutnya adalah institusi dapat menentukan strategi dengan memberikan perhatian lebih bagi mahasiswa yang diprediksi akan terlambat.

## METODE PENELITIAN

Tahap pertama yang dilakukan adalah pengumpulan data. Data yang diperoleh adalah sebanyak 227 data set mahasiswa yang telah menyelesaikan studinya yaitu data set mahasiswa tahun angkatan 2013 dan 2014 dengan 11 atribut. Tahap kedua dilakukan pra-pemrosesan data atau pengolahan data awal untuk mendapatkan data yang baik sebelum data diolah menggunakan menggunakan metode Algoritme C4.5 dan *Naive Bayes Classifier*. Setelah dilakukan pra-pemrosesan data, maka data set yang akan digunakan pada proses mining adalah 162 data mahasiswa dengan 9 atribut. Tahap ketiga dilakukan proses mining menggunakan metode Algoritme C4.5 dan *Naive Bayes Classifier* pada tools RapidMiner. Untuk memvalidasi nilai akurasi kedua metode yang digunakan diterapkan tehnik *K-Fold Cross Validation* dan hasil akurasi dapat dilihat berdasarkan *Confusion Matrix*. Tahap selanjutnya hasil pengujian dari metode Algoritme C4.5 dan *Naive Bayes Classifier* akan dibandingkan, dengan tujuan untuk mengetahui metode yang terbaik dengan tingkat akurasi yang paling tinggi.

Agar lebih jelas desain penelitian yang penulis gunakan dapat dilihat seperti pada Gambar 1.




Gambar 1. Desain Penelitian

## 1. Pengumpulan Data

Pengumpulan data dilakukan langsung dilapangan yaitu data mahasiswa program studi Sistem Informasi tahun angkatan 2013 dan 2014 yang sudah lulus yang diperoleh dari bagian akademik. Data yang diperoleh yaitu 227 data set mahasiswa dengan 11 atribut atau variabel. Variabel yang digunakan antara lain adalah Jenis Kelamin, Status Sekolah, Asal Sekolah, IP semester 1, IP semester 2, IP semester 3, IP semester 4, IPK semester 4 dan Status Kelulusan. Adapun contoh data yang digunakan dapat dilihat pada Tabel 1.

Tabel 1. Contoh Data

| *No* | *NIM* | *Nama* | *Jenis Kelamin* | *Status Sekolah* | *Asal Sekolah* | *IP-S1* | *IP-S2* | *IP-S3* | *IP-S4* | *IPK-S4* | *Status Kelulusan* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2013.01.0001 | Ahmad Shalihin | Laki-laki | Swasta | SMA | 3,41 | 3,05 | 3,09 | 3,18 | 3,18 | Tepat Waktu |
| 2 | 2013.01.0003 | Irma Tilawati | Perempuan | Negeri | SMA | 3,23 | 3,05 | 2,89 | 3,2 | 3,1 | Tepat Waktu |
| 3 | 2013.01.0004 | Muhammad Hidayatullah | Laki-laki | Negeri | SMK | 3,27 | 3,14 | 3,14 | 3 | 3,14 | Tepat Waktu |
| 4 | 2013.01.0005 | Nurhidayah | Perempuan | Negeri | SMK | 3,32 | 2,95 | 2,78 | 3,1 | 3,05 | Tepat Waktu |
| 5 | 2013.01.0006 | Sutrisno Raja Guk Guk | Laki-laki | Negeri | SMA | 3,18 | 3,05 | 3 | 3 | 3,06 | Tepat Waktu |
| 6 | 2013.01.0008 | Duwi Santoso | Laki-laki | Swasta | SMK | 2,91 | 3,05 | 2,18 | 2,29 | 2,34 | Terlambat |
| 7 | 2013.01.0009 | Hesti Kurnia | Perempuan | Negeri | SMK | 3,05 | 2,95 | 2,65 | 3 | 2,92 | Terlambat |
| 8 | 2013.01.0010 | Edi Lianto | Laki-laki | Negeri | SMA | 3,73 | 3,23 | 3,82 | 3,55 | 3,58 | Tepat Waktu |
| 9 | 2013.01.0011 | Rina | Perempuan | Swasta | SMA | 3,59 | 3,68 | 3,91 | 3,82 | 3,75 | Tepat Waktu |
| 10 | 2013.01.0012 | Dayang Sejoli | Perempuan | Swasta | SMA | 3,73 | 3,64 | 3,86 | 3,55 | 3,69 | Tepat Waktu |

## 2. Alat dan Bahan

Alat dan bahan yang digunakan dalam penelitian ini antara lain adalah:

1) *Ms. Excel* digunakan untuk pengolahan data mentah atau data awal.
2) RapidMiner merupakan *tool* yang dimanfaatkan untuk mengimplementasikan metode *data mining* yang digunakan.

3) *Naive Bayes Classifier* dan Algoritme C4.5 sebagai Algoritme perhitungan untuk menyelesaikan masalah prediksi status kelulusan mahasiswa.
4) Teknik *K-Fold Cross-Validatio*n diterapkan untuk memvalidasi nilai akurasi model yang dibangun.

## 3. Pra Pemrosesan Data

Pada penelitian ini prediksi dilakukan berdasarkan data-data yang sudah terjadi, maksudnya adalah data yang penulis gunakan berupa data mahasiswa yang sudah menyelesaikan waktu studinya. Jadi data yang akan diolah telah memiliki variabel tujuan yaitu status kelulusan yang dikategorikan tepat waktu dan terlambat. Hal ini dimaksudkan agar dapat diketahui nilai akurasi hasil prediksi berdasarkan penerapan dari dua metode *data mining* yang digunakan. Penelitian ini sejalan dengan penelitian Risqiati & Ismanto (2017) penelitian tersebut menggunakan data kelulusan mahasiswa sebagai data set yang diimplementasikan dengan metode Algoritme C4.5 dan *Naive Bayes Classifier* pada tool RapidMiner.

Hasil dari pengumpulan data didapatkan *record* sebanyak 227 data set mahasiswa yang telah menyelesaikan studinya yaitu data set mahasiswa tahun angkatan 2013 dan 2014 dengan 11 atribut. Mahasiswa yang dikategorikan lulus tepat waktu yakni mahasiswa yang dapat menyelesaikan studinya selama 7 semester (3,5 tahun) atau 8 semester (4 tahun) untuk program sarjana. Sedangkan mahasiswa yang menyelesaikan pendidikannya lebih dari 8 semester, maka dikategorikan terlambat. Namun dari hasil pengumpulan data, data *record* dan atribut tidak seluruhnya bisa digunakan karena perlu dilakukan pra pemrosesan data atau pengolahan data awal untuk mendapatkan data yang baik. Adapun rincian 11 atribut yang belum dilakukan pra pemrosesan data terlihat seperti dalam Tabel 2 berikut:

Tabel 2. Atribut Sebelum Pra Pemprosesan Data

| *No* | *Nama* | *Jenis Data* |
|---|---|---|
| 1 | NIM | Karakter |
| 2 | Nama | Karakter |
| 3 | Jenis Kelamin | Kategorikal |
| 4 | Status Sekolah | Kategorikal |
| 5 | Asal Sekolah | Kategorikal |
| 6 | IP-S1 | Numerik |
| 7 | IP-S2 | Numerik |
| 8 | IP-S3 | Numerik |
| 9 | IP-S4 | Numerik |
| 10 | IPK-S4 | Numerik |
| 11 | Status Kelulusan | Kategorikal |

Beberapa penelitian yang telah dilakukan menyatakan bahwa pra pemrosesan data perlu dilakukan untuk mendapatkan data set dengan kualitas baik. Seperti penelitian yang dilakukan oleh Zainuddin (2019) teknik *preproccesing* dilakukan untuk mendapatkan data dengan kualitas baik. Cara yang dilakukan antara lain *validation* data yaitu untuk menghilangkan pencilan, derau, data yang kosong dan yang inkonsisten, serta *discretization* data yaitu dilakukan seleksi atribut kelulusan. Selanjutnya penelitian yang dilakukan oleh Septiani (2017) untuk mendapatkan data dengan kualitas baik beberapa teknik yang dapat dilakukan antara lain *validation, integration and transformation, size reduction/discritization*. Dan dalam penelitian yang dilakukan oleh Prakoso & Tutik (2017) menyatakan pentingnya *preproccesing* data sebelum data set diproses menggunakan teknik *data*

*mining*. *Preproccesing* meliputi: memeriksa dan membuang data yang inkonsisten, data ganda, data yang perlu diperbaiki dan atau menambah data sesuai dengan kebutuhan.

Berdasarkan pada beberapa penelitian di atas, maka pada penelitian ini pra pemrosesan data dilakukan untuk mendapatkan data dengan kualitas baik. Pra pemrosesan data yang penulis gunakan antara lain:

a. Pembersihan Data, yaitu menghilangkan data yang kosong dan tidak lengkap. Misalnya, *record* mahasiswa dengan status berhenti dan non aktif dihapuskan karena mengandung data nilai mata kuliah yang tidak lengkap. Sehingga data yang awalnya berjumlah 227 menjadi 162 data set saja, jadi sebanyak 28,63% data yang kosong dan tidak lengkap dibersihkan/dihapus pada tahap pra pemrosesan data untuk menghindari adanya *missing value* dalam data set.
b. Reduksi Data, dilakukan guna mendapatkan data set dengan *record* dan jumlah atribut yang bersifat informatif saja. Sebagai contoh atribut NIM dan Nama tidak digunakan pada proses mining karena tidak relevan. Jadi atribut yang digunakan pada proses mining hanya atribut yang bersifat informatif saja yaitu jenis kelamin, status sekolah, asal sekolah, IP-S1, IP-S2, IP-S3, IP-S4, IPK-S4 dan Status Kelulusan.
c. Transformasi Data, digunakan untuk mengubah IP-S1, IP-S2, IP-S3, IP-S4 dan IPK-S4 yaitu nilainya dibuatkan interval yang lebar dan kedalamannya sama. Implementasi dilakukan pada *tool* RapidMiner, pra pemrosesan data dilakukan menggunakan operator *Discretize*.

Setelah dilakukan pra pemrosesan data, maka data set yang digunakan pada proses mining adalah 162 data mahasiswa dengan 9 atribut yang telah dinormalisasi dan *missing value* tidak terdapat pada data set tersebut. Adapun rincian atribut yang digunakan pada proses mining terlihat seperti pada Tabel 3:

Tabel 3. Atribut Data Setelah Pra Pemprosesan Data

| *No* | *Nama* | *Jenis Data* |
|---|---|---|
| 1 | Jenis Kelamin | Kategorikal |
| 2 | Status Sekolah | Kategorikal |
| 3 | Asal Sekolah | Kategorikal |
| 4 | IP-S1 | Numerik |
| 5 | IP-S2 | Numerik |
| 6 | IP-S3 | Numerik |
| 7 | IP-S4 | Numerik |
| 8 | IPK-S4 | Numerik |
| 9 | Status Kelulusan | Kategorikal |

## 4. RapidMiner

Pada penelitian ini *tool* RapidMiner yang merupakan *tool data mining* Selain itu *tool* ini menampilkan visualisasi hasil olahan data. *Tool* RapidMiner adalah sebuah *tool* yang bersifat *open source*. RapidMiner menurut Mulya (2019) adalah sebuah alat *data mining* yang digunakan untuk menganalisa informasi. Pada penelitiannya Supriyanti, Kusrini, & Armadyah (2016) menyatakan bahwa RapidMiner merupakan sebuah tool yang dapat digunakan untuk membantu menyelesaikan masalah prediksi, proses *data mining* dan text mining. *Tool* RapidMiner mempunyai banyak operator *data mining* (lebih dari 500 operator), termasuk operator untuk *input*, *output*, data *preproccesing* dan lain-lain.

### *5. Naive Bayes Classifier*

Dalam penelitian ini metode *Naive Bayes Classifier* digunakan sebagai Algoritme perhitungan untuk menyelesaikan masalah prediksi. Metode ini menggunakan teorema Bayes, yang bekerja berdasarkan probabilitas sederhana dinyatakan dengan persamaan berikut:

$$P\ (H|E) = \frac{P(E|H)P(H)}{P(E)} \qquad (1)$$

Keterangan persamaan:

E = Bukti

H = Hipotesis

P(H|E) = Hipotesis H benar untuk bukti E

P(E|H) = Kemungkinan sebuah bukti E terjadi akan memengaruhi hipotesis H atau dengan kata lain kemungkinan bahwa bukti E benar untuk hipotesis H

P(H) = Probabilitas awal (priori) hipotesis H terjadi tanpa memandang bukti apapun

P(E) = Probabilitas awal (priori) bukti E terjadi tanpa memandang hipotesis/bukti yang lain

Maksud dari aturan Bayes yakni berdasarkan bukti-bukti (E) yang diamati maka hasil hipotesis (H) dapat diprediksi.

### 6. Algoritme C4.5

Metode kedua yang penulis gunakan sebagai Algoritme perhitungan untuk menyelesaikan masalah prediksi kelulusan mahasiswa adalah Algoritme C4.5. Algoritme C4.5 menurut Prakoso & Tutik (2017) yaitu metode yang bisa diterapkan untuk menyelesaikan masalah klasifikasi data dengan atribut kategorial. Sedangkan Anam & Santoso (2018) berpendapat bahwa Algoritme C4.5 diterapkan guna membentuk sebuah pohon keputusan yang mempresentasikan aturan dalam klasifikasi.

Elemen penting yang harus dipahami dalam Algoritme C4.5 yaitu Enteropy dan Gain. Tahapan dalam membangun pohon keputusan antara lain adalah:

1. Memilih atribut untuk dijadikan node/akar
2. Membuat cabang dari setiap nilai
3. Membagi kasus pada setiap cabang
4. Ulangi proses dalam setiap cabang sampai seluruh kasus pada cabang berada pada kelas yang sama.

Pemilihan atribut sebagai node/akar yakni berdasarkan nilai Gain tertinggi dari dari semua atribut. Berikut adalah persamaan untuk menghitung nilai Gain:

$$Gaint\ (S, A) = Entropy\ (S) \sum_{i=i}^{n} \frac{|\ S_i\ |}{|\ S\ |} * \qquad (2)$$

Keterangan:

S = Himpunan Kasus

A = Atribut

n = Jumlah Partisi Atribut A

$|\ S_i\ |$ = Jumlah Kasus Pada Partisi Ke - i

| S | = Jumlah Kasus Dalam S

Nilai Enteropy dapat dihitung menggunakan persamaan berikut:

$$Entropy\ (S) = \sum_{i=i}^{n} Pi * Log_2\ Pi \qquad (3)$$

Keterangan:

S = Himpunan Kasus

A = Fitur

n = Jumlah Partisi S

$P_i$ = Proporsi Dari $S_i$ Terhadap S



http://dx.doi.org/10.35671/telematika.v13i1.881

**7. Evaluasi Metode Klasifikasi**

Evaluasi metode klasifikasi bertujuan untuk menganalisa perbandingan kinerja dari metode klasifikasi yang digunakan. Dalam penelitiannya Anam & Santoso (2018) menjelaskan bahwa evaluasi kinerja model klasifikasi dimaksudkan untuk mengetahui kinerja model klasifikasi berdasarkan hasil pengujian moedel yang diterapkan. Dalam penelitian ini hasil implementasi metode Algoritme C4.5 akan dibandingkan dengan *Naive Bayes Classifier*. Untuk memvalidasi nilai akurasi model yang dibangun digunakan metode *K-Fold Cross Validation* dan hasil akurasi dapat dilihat berdasarkan *Confusion Matrix*.

***a. K-Fold Cross Validation***

*K-Fold Cross Validation* menurut Anam & Santoso (2018) adalah teknik untuk memvalidasi nilai akurasi metode yang diterapkan berdasarkan data set. Suyanto (2017) menyatakan bahwa metode *K-Fold Cross Validation* membagi himpunan data D secara acak menjadi *k-fold* (sub himpunan) yang saling bebas $f_1$, $f_2$, ..., $f_k$, sehingga setiap *fold* berisi 1/k bagian data. Selanjutnya dapat membangun k himpunan data: $D_1$, $D_2$, ..., $D_k$, yang masing-masing berisi (k-1) *fold* untuk *training* data, 1-fold untuk *testing* data. Pada umumnya metode *k-fold cross validation* menggunakan 10 kali iterasi (k=10) dengan tujuan untuk memperoleh akurasi dengan bias dan variansi yang cukup rendah.

***b. Confusion Matrix***

Tujuan *Confusion Matrix* menganalisa kualitas kinerja model klasifikasi dalam mengenali variabel dari seluruh kelas. *Confussion Matrix* berisi informasi mengenai kelas sebenarnya dan kelas prediksi dari suatu proses klasifikasi (Anam & Santoso, 2018). Tabel matrix digunakan untuk mempresentasikan hasil evaluasi model klasifikasi. Misalnya data set terbagi menjadi kelas A dan kelas B, maka kelas A diasumsikan sebagai variabel positif dan kelas B diasumsikan sebagai variabel negatif. Nilai *accuracy*, *reccal* dan *precission* dapat diperoleh dari hasil evaluasi menggunakan *Confusion Matrix*. Gambar 2 merupakan contoh *Confusion Matrix*:

| | | Kelas Hasil Prediksi | | |
|---|---|---|---|---|
| | | Ya | Tidak | Jumlah |
| Kelas Aktual | Ya | TP | FN | P |
| | Tidak | FP | TN | N |
| | Jumlah | P | N | P + N |

Gambar 2. *Confusion Matrix*

Perhitungan nilai akurasi, *precission* dan *reccal* dinyatakan dalam persamaan berikut:

$$Accuracy = \frac{TP+TN}{P+N} \quad (4)$$

$$Precission = \frac{TP}{TP+FP} \quad (5)$$

$$Reccal = \frac{TP}{P} \quad (6)$$

Keterangan:

TP *(True Positive)* : Jumlah variabel positif yang dilabeli dengan benar oleh *classifier*, sebagai contoh variabel dengan label status kelulusan = tepat waktu
TN *(True Negative)* : Jumlah variabel negatif yang dilabeli dengan benar oleh *classifier*
FP *(False Positive)* : Jumlah variabel negatif yang salah dilabeli oleh *classifier*
FN *(False Negative)* : Jumlah variabel positif yang salah dilabeli oleh *classifier*
P : Jumlah sampel positif

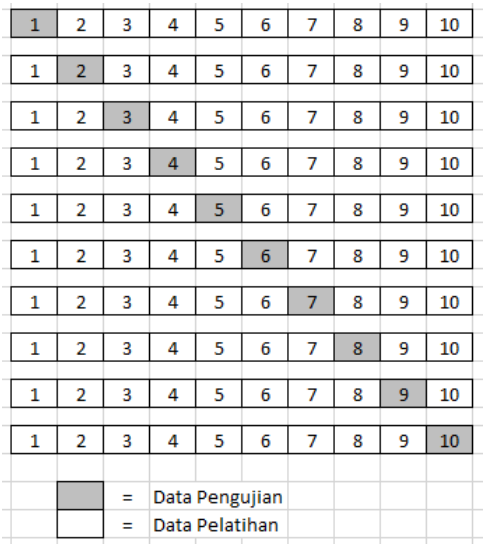

http://dx.doi.org/10.35671/telematika.v13i1.881

N : Jumlah sampel negatif

## HASIL DAN PEMBAHASAN

### 1. Analisa Data

Data yang digunakan pada proses *training* dan *testing* telah melewati proses prapemrosesan data sehingga data telah siap diolah. Dataset yang digunakan dalam proses *training* maupun *testing* sebanyak 162 *record.* Atribut yang digunakan dalam proses *training* dan *testing* terdiri dari atribut Jenis Kelamin, Status Sekolah, Asal Sekolah, IP-S1, IP-S2, IP-S3, IPK-S4 dan Status Kelulusan. Proses *training* dan *testing* diterapkan pada *tool* RapidMiner 5.3 menggunakan metode perhitungan yaitu *Naive Bayes Classifier* dan Algoritme C4.5. Proses pengujian dilakukan dengan cara menambahkan operator *X-Validation (k-Fold Cross-Validation)* dengan melakukan 10 kali iterasi atau pengulangan (k=10 *fold)*. Dalam 10 *Fold Cross-Validation*, data dibagi menjadi 10 subset data. Dari 10 subset data tersebut *Cross-Validation* akan menggunakan 9 *fold* untuk pelatihan dan 1 *fold* untuk untuk pengujian seperti pada Gambar 3.




Gambar 3. Skema 10 *Fold Cross Validation*

Selanjutnya nilai akurasi dari hasil pengujian dapat dilihat berdasarkan *Confussion Matrik* seperti pada Gambar 4 dan Gambar 5.

### 2. Hasil Implementasi Metode *Naive Bayes Classifier* dan Algoritme C4.5

Dari hasil implementasi metode *Naive Bayes Classifier* didapatkan nilai akurasi sebesar 78,46% yang dievaluasi menggunakan *Confussion Matrix.* Gambar 4 menampilkan detail hasil yang diperoleh dari proses mining yang dilakukan dengan menggunakan *Naïve Bayes Classifier*.

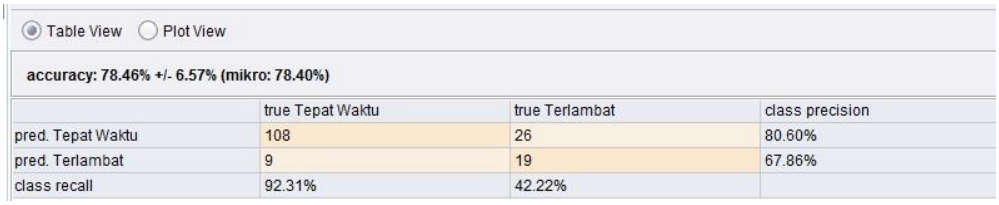


Table View Plot View

accuracy: 78.46% +/- 6.57% (mikro: 78.40%)

| | true Tepat Waktu | true Terlambat | class precision |
|---|---|---|---|
| pred. Tepat Waktu | 108 | 26 | 80.60% |
| pred. Terlambat | 9 | 19 | 67.86% |
| class recall | 92.31% | 42.22% | |

Gambar. 4 Nilai Akurasi Metode *Naive Bayes Classifier*

Gambar 4 menampilkan hasil perhitungan akurasi data set dengan metode *Naive Bayes Classifier* berdasarkan *confussion matrix*. Dari gambar tersebut dapat dilihat bahwa nilai *Accuracy* Rata-rata sebesar 78,46%, nilai *Reccal* kelas tepat waktu sebesar 92,31%, Nilai *Reccal* kelas terlambat sebesar 42,22%, nilai

*Preccision* kelas tepat waktu sebesar 80,60% dan nilai *Preccision* kelas terlambat sebesar 67,86%. Dari 162 data set, terdapat 108 data yang sesuai prediksi yaitu "tepat waktu", dan 26 yang diprediksi "tepat waktu" ternyata "terlambat". Dan sebanyak 9 data yang diprediksi "terlambat" tenyata termasuk kalsifikasi "tepat waktu" dan sebanyak 19 data sesuai prediksi yaitu "terlambat".

Selanjutya hasil implementasi metode Algoritme C4.5 pada *tool* RapidMiner diperoleh nilai *Accuracy* sebesar 79,08% berdasarkan *Confussion Matrix* seperti pada Gambar berikut:

Table View Plot View

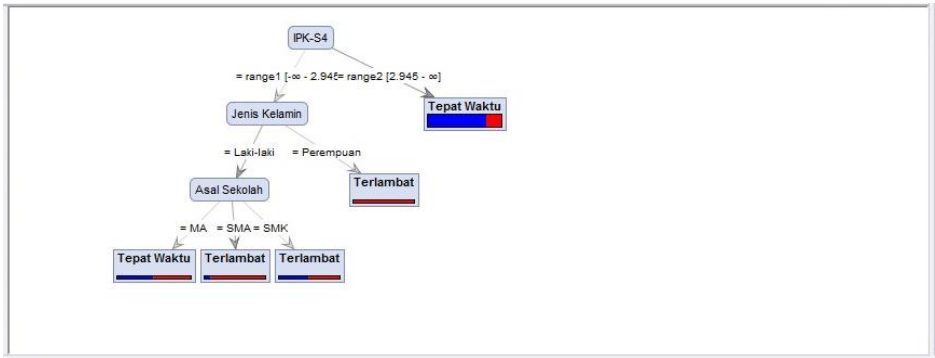
**accuracy: 79.08% +/- 7.61% (mikro: 79.01%)**

| | true Tepat Waktu | true Terlambat | class precision |
|---|---|---|---|
| pred. Tepat Waktu | 114 | 31 | 78.62% |
| pred. Terlambat | 3 | 14 | 82.35% |
| class recall | 97.44% | 31.11% | |

Gambar 5. Nilai Akurasi Metode Algoritme C4.5

Gambar 5 menampilkan hasil dari perhitungan akurasi data set dengan metode Algoritme C4.5 berdasarkan *confussion matrix*. Dari gambar tersebut dapat dilihat bahwa nilai *Accuracy* dari metode ini sebesar 79,08%, Nilai *Reccal* kelas tepat waktu sebesar 97,44%, Nilai *Reccal* kelas terlambat sebesar 31,11%, nilai *Preccision* kelas tepat waktu sebesar 78,62% dan nilai *Preccision* kelas terlambat sebesar 82,35%. Dari 162 data set, terdapat 114 data yang sesuai prediksi yaitu "tepat waktu", dan 31 yang diprediksi "tepat waktu" ternyata "terlambat". Dan sebanyak 3 data yang diprediksi "terlambat" tenyata termasuk kalsifikasi "tepat waktu" dan sebanyak 14 data sesuai prediksi yaitu "terlambat".

Dari hasil implementasi Algoritme C4.5 menggunakan *tool* RapidMiner maka terbentuk pohon keputusan berdasarkan nilai gain tertinggi adalah sebagai berikut:




Gambar 6. Pohon Keputusan Berdasarkan Informatif Gain

Dari Gambar 6 dapat dilihat bahwa variabel atau kriteria yang berpengaruh dalam prediksi kelulusan mahasiswa di STMIK Bina Nusantara Jaya Lubuklinggau adalah IPK-S4, Jenis Kelamin dan Asal Sekolah. Dari seluruh variabel yang digunakan, variabel IPK-S4 menjadi simpul akar, hal tersebut dikarenakan nilai gain tertinggi ada pada variabel IPK-S4. Variabel nilai IPK-S4 adalah nilai komulatif terakhir yang diambil selama mahasiswa mengikuti proses belajar di STMIK Bina Nusantara Jaya Lubuklinggau. Sehingga dengan IPK-S4 bisa menggambarkan kemajuan proses perkuliahan mahasiswa dan kendala atau hambatan yang dihadapi masing-masing mahasiswa. Data tersebut merupakan data paling dekat yang menggambarkan data prediksi kelulusan mahasiswa dibandingkan variabel-variabel yang lain. Hal ini sejalan dengan penelitian yang sudah dilakukan oleh Romadhona, suprapedi & himawan (2017) dan Priati (2016), dalam penelitiannya salah satu faktor dominan yang berpengaruh dalam prediksi kelulusan mahasiswa adalah IPK-S4.

Selain IPK-S4 yang menarik dari hasil pohon keputusan Algoritme C4.5 adalah jenis kelamin yang menjadi salah satu variabel dominan dalam penelitian ini. Dari pohon keputusan terlihat bahwa jenis kelamin perempuan diprediksi terlambat sedangkan jenis kelamin laki-laki diprediksi tepat waktu jika asal sekolah = MA. Jadi, dapat disimpulkan bahwa mahasiswa dengan jenis kelamin perempuan dan laki-laki yang asal sekolahnya SMK atau SMK perlu diberikan perhatian dan bimbingan yang lebih serius agar dapat dapat memperbaiki IPK pada setiap semester sehingga dapat lulus tepat waktu.

**3. Perbandingan Hasil Akurasi Metode *Naive Bayes Classifier* dan Algoritme C4.5**

Hasil dari implementasi yang telah dilakukan, perbandingan tingkat akurasi antara metode Algoritme C4.5 dan *Naive Bayes Classifier*:

Tabel 5. Perbandingan Nilai Akurasi Metode Naive Bayes Classifier dan Algoritme C4.5

| *No* | *Metode* | *Nilai Akurasi* |
|---|---|---|
| 1 | *Naive Bayes Classifier* | 78,46% |
| 2 | Algoritme C4.5 | 79,08% |

Berdasarkan tabel diatas, prediksi kelulusan mahasiswa menggunakan metode Algoritme C4.5 memiliki nilai akurasi yang lebih tinggi dibandingkan dengan nilai akurasi metode *Naive Bayes Classifier* yaitu 79,08%. Selisih nilai akurasi antara kedua metode tersebut adalah sebesar 0,62%. Hal ini sejalan dengan penelitian yang telah dilakukan oleh Anam & Santoso (2018) dan penelitian yang dilakukan oleh Risqiati & Ismanto (2017) dimana nilai akurasi metode Algoritme C4.5 lebih besar dari metode *Naive Bayes Classifier*.

## KESIMPULAN DAN SARAN

Kesimpulan dari penelitian ini bahwa hasil prediksi kelulusan mahasiswa pada STMIK Bina Nusantara Jaya Lubuklinggau berdasarkan data set yang diimplementasikan dengan metode *Naive Bayes Classifier* menunjukan nilai *Accuracy* 78,46% dan prediksi menggunakan metode Algoritme C4.5 diperoleh nilai *Accuracy* yang lebih besar yaitu 79,08%. Karena Algoritme C4.5 memiliki nilai akurasi yang lebih besar dibandingkan dengan nilai akurasi metode *Naive Bayes Classifier* maka metode Algoritme C4.5 direkomendasikan untuk digunakan dalam menyelesaikan masalah prediksi kelulusan mahasiswa pada STMIK Bina Nusantara Jaya Lubuklinggau. Dan dari pohon keputusan hasil implementasi Algoritme C4.5 dapat disimpulkan bahwa variabel atau kriteria yang berpengaruh dalam prediksi kelulusan mahasiswa di STMIK Bina Nusantara Jaya Lubuklinggau adalah IPK-S4, Jenis Kelamin dan Asal Sekolah.

Untuk pengembangan dan penelitian selanjutnya, penulis memberikan beberapa saran, yang pertama sebaiknya jumlah data perlu ditambah guna meningkatkan nilai *Accuracy*. Yang kedua, yaitu bukan hanya faktor intern atau faktor akademik saja yang dijadikan sebagai variabel atau kriteria namun faktor eksternal misalnya status bekerja, status pernikahan, faktor pembiayaan, dll perlu dijadikan sebagai variabel atau kriteria. Yang ketiga, penerapan *fitur selection* perlu dilakukan untuk pengembangan penelitian ini, atau untuk penelitian sejenis yang akan dilakukan. Selanjutnya penelitian sejenis dapat dilakukan dengan menerapkan metode *data mining* yang berbeda dengan metode yang telah penulis gunakan. Dan untuk pengembangan penelitian dapat dilakukan dengan mengadopsi hasil prediksi untuk dijadikan sebagai pendukung dalam proses pengambilan keputusan oleh para pemangku keputusan.

**DAFTAR PUSTAKA**

Amalia, H. E. (2017). Komparasi Metode Data Mining Untuk Penentuan Proses Persalinan Ibu Melahirkan, *13*, 103–109.

Anam, C., & Santoso, H. B. (2018). Perbandingan Kinerja Algoritma C4 . 5 dan Naive Bayes untuk Klasifikasi Penerima Beasiswa, *8*(1), 13–19.

Bisri, A. (2015). Penerapan Adaboost untuk Penyelesaian Ketidakseimbangan Kelas pada Penentuan Kelulusan Mahasiswa dengan Metode Decision Tree, *1*(1).

Juliansa, H. (2019). Data Mining Rought Set Dalam Menganalisa Kinerja Dosen STMIK Bina Nusantara Jaya Lubuklinggau, *4*(1), 11–17.

Mulya, D. P. (2019). Analisa dan Implementasi Association Rule Dengan Algoritma FP-Growth, *1*(1), 47–57.

Prakoso, S. A., & Tutik, E. T. (2017). Komparasi Algoritma C4.5 Dengan Naive Bayes Untuk Klasifikasi Kelulusan Mahasiswa Tepat Waktu Di PTS "KZX," *3*(1).

Priati. (2016). Kajian Perbandingan Teknik Klasifikasi Algoritma C4 . 5 , Naive Bayes Dan Cart Untuk Prediksi Kelulusan Mahasiswa (Studi Kasus : STMIK Rosma Karawang), (July 2016). https://doi.org/10.5281/zenodo.1184054

Risqiati, & Ismanto, B. (2017). Analisis Komparasi Algoritma Naive Bayes Dan C4-5 Untuk Waktu Kelulusan Mahasiswa, *XII*(1), 33–38.

Romadhona, Agus; suprapedi; himawan, H. (2017). Prediksi Kelulusan Tepat Waktu Mahasiswa Stmik-Ymi, *13*, 917.

Salmu, S., & Solichin, A. (2017). Prediksi Tingkat Kelulusan Mahasiswa Tepat Waktu Menggunakan Naïve Bayes : Studi Kasus UIN Syarif Hidayatullah Jakarta Prediction of Timeliness Graduation of Students Using Naïve Bayes : A Case Study at Islamic State University Syarif Hidayatullah Jakarta, (April), 701–709.

Septiani, W. D. (2017). Komparasi Metode Klasifikasi Data Mining Algoritma C4.5 Dan Naive Bayes Untuk Prediksi Penyakit Hepatitis. *Jurnal Pilar Nusa Mandiri*, *13*(1), 76–84.

Supriyanti, W., Kusrini, & Armadyah, A. (2016). Perbandingan kinerja algoritma c4.5 dan naive bayes untuk ketepatan pemilihan konsentrasi mahasiswa, *1*(2012).

Suyanto. (2017). *Data Mining Untuk Klasifikasi dan Klasterisasi Data*. Informatika Bandung.

Zainuddin, M. (2019). Perbandingan 4 Algoritma Berbasis Particle Swarm Optimization ( pso ) Untuk Prediksi Kelulusan Tepat Waktu Mahasiswa, *13*(1), 1–12.