**Indonesian Journal of Electrical Engineering and Computer Science**
Vol. 12, No. 3, December 2018, pp. 1239~1246
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v12.i3.pp1239-1246 

# Developing Corpora using Word2vec and Wikipedia for Word Sense Disambiguation

**Farza Nurifan, Riyanarto Sarno, Cahyaningtyas Sekar Wahyuni**
Department of Informatics, Institut Teknologi Sepuluh Nopember,
Jalan Raya ITS, Keputih, Sukolilo, Kota Surabaya, Jawa Timur 60111, Indonesia

**Article Info**

***Article history:***

Received May 25, 2018
Revised Aug 25, 2018
Accepted Sep 7, 2018

***Keywords:***

Lesk
Wikipedia
Word sense disambiguation
Word2vec
Wu palmer

**ABSTRACT**

Word Sense Disambiguation (WSD) is one of the most difficult problems in the artificial intelligence field or well known as AI-hard or AI-complete. A lot of problems can be solved using word sense disambiguation approach such as sentiment analysis, machine translation, search engine relevance, coherence, anaphora resolution, and inference. This research is done to solve WSD problem with two small corpora. The use of Word2vec and Wikipedia are proposed to develop the corpora. After developing the corpora, the similarity of the sentence with the corpora is measured using cosine similarity to determine the meaning of the ambiguous word. Lastly, to improve accuracy, Lesk algorithms and Wu Palmer similarity are used to deal with problems when there is no word from a sentence in the corpus. The results of the research show an 85.51% accuracy rate and the semantic similarity improve the accuracy rate by 8.02% in determining the meaning of ambiguous words.



***Corresponding Author:***

Riyanarto Sarno,
Department of Informatics,
Institut Teknologi Sepuluh Nopember,
Jalan Raya ITS, Keputih, Sukolilo, Kota Surabaya, Jawa Timur 60111, Indonesia.
Email: riyanarto@if.its.ac.id

## 1. INTRODUCTION

Natural languages contain a few words with a different meaning in a different context [1]. Human can easily distinguish it because we have the ability to see the context of the sentence and determine the meaning of the ambiguous words. To make computers understand the meaning of an ambiguous word, it requires a very difficult technique. Therefore, Word Sense Disambiguation (WSD) is existed to determine the meaning of ambiguous words [2]. For example, the ambiguous word is the word 'bank':

1. *"He sat down beside the Seine river bank"* [3].
2. *"He deposited the money at the Chase bank"* [3].

The word bank in both sentences has a different meaning. In the first sentence, it means a place near the river, while in the second sentence, it means a financial institution.

Word sense disambiguation is very important problem because it has many uses such as machine translation [4] or sentiment analysis [5]. In machine translation, translating a sentence containing the ambiguous words cannot be done directly without looking the context. Otherwise, it can be wrong. he accuracy of machine translation in translating words can be improved [6]. One of the examples is by using Word sense disambiguation.

Many researchers have proposed various approaches to solve word sense disambiguation problems, but none of it can handle inexistent words in a corpus. In [7], proposed the uses of adapted weighted graph to solve the problem. In [8], proposed the uses of machine learning to solve the problem. Another way to solve word sense disambiguation is by using corpus. Corpus is a set of structured text that has many uses. One of

them can be used to classify emotions from music [9], emotions from a text [10], and word sense disambiguation. In [3], proposed a word sense disambiguation solution using Skip-Gram corpora. In [3], Google Word Sense Disambiguation Corpora as the corpora and achieved a result with accuracy 42.12%. However, the existed method using corpus did not handle problem if there is no word from sentence that are in the corpus.

In this research, the use of Wikipedia and Word2vec is proposed to develop the corpora. Meanwhile, Lesk algorithm and Wu Palmer similarity are used to handle problem if there is no word from sentence that are in corpus. First, two corpora are developed using data from Wikipedia. The data obtained from Wikipedia then preprocessed to minimize the words variations [11]. After preprocessing the data, corpora are created using Word2vec. Second, the corpora are used to determine the meaning of an ambiguous word. To conduct this, the similarity of a sentence to the first and the second corpus is calculated using cosine similarity. If there are any words from the sentence that do not belong to corpora, Lesk algorithm [12] and Wu Palmer [13] are used to calculate the similarity. Then, the meaning of an ambiguous words is determined based on the value of similarity that has been calculated.

## 2. RESEARCH METHOD

The main objective of this research is to develop the corpora and to use it as a tool to solve word sense disambiguation problem. The proposed method is divided into three parts; the first part is developing the Wikipedia corpora; the second part is determining the result; the third part is performance measure.

### 2.1. Developing Wikipedia Corpora

Figure 1 shows the process of developing the corpora. The two datasets from Wikipedia, such as a word "bank (financial)" and "bank (geography)", are used as an input to be preprocessed. Then, the preprocessed data are used to create two corpora using word2vec. Corpus 1 and corpus 2 are the output of each dataset.

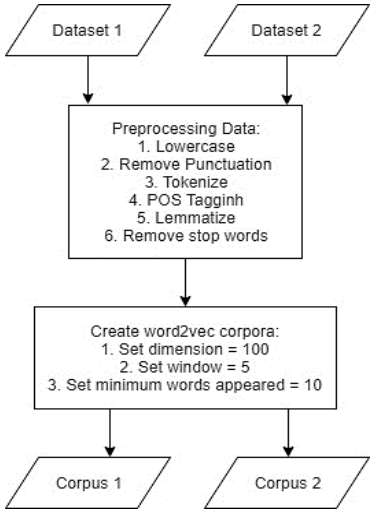


Figure 1. Corpora development

#### 2.1.1 Dataset

The Wikipedia article has many features including a table of contents, article references and category labels. In this paper, the use of category label feature from Wikipedia article is proposed to determine which article will be selected as a dataset to develop the corpus. For example, we develop corpora for word "bank", the first corpus is bank as a financial institution and the second corpus is bank as geography. The article selection is based on the category labels from Wikipedia. For bank as a financial institution, the Wikipedia articles that contains word "bank" with category labels related to financial institution are selected. The categories we found that related to financial institution are Banks, Banking, Legal Entities, Italian Inventions, and Economic History of Italy. For bank as geography, the category labels are Hydrology, Geomorphology, Limnology, Freshwater Ecology, Fluvial Landforms, Riparian Zone, Rivers, Water Streams, and Water and the Environment. After choosing the articles, then the

content of the article in the paragraph is obtained. Paragraphs taken from Wikipedia are then broken down into sentences based on period punctuation. The process can be seen in Figure 2.

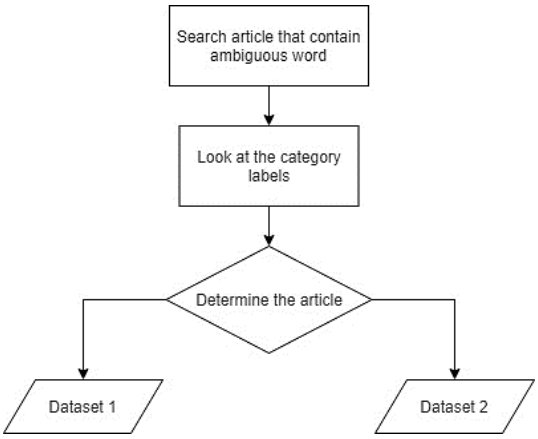


Figure 2. Article selection

### 2.1.2 Preprocessing

The sentences from the dataset have so many variations. This condition makes the process for creating the corpora will be less accurate. The preprocess itself has six steps to do.

a) Lowercase
This is the simple way to make the words variations to be less. For example, if there are two words "Money" and "money" it will be recognized as same word.

b) Remove punctuation
In building the corpora and testing it with our testing data, we only need the words. Therefore, the punctuations are removed.

c) Tokenize
We tokenize the input sentence to make it easier to be processed at the next step, which is POS tagging and Lemmatizing.

d) POS tagging
To make the data more accurate we use POS tagging. Part of Speech (POS) Tagging is commonly used to determine whether a word is a noun word, a verb, an adjective or an adverb.

e) Lemmatize
This is the part important process to make the data to have less variations. We will make words like "banks" to be same as the word "bank". We lemmatized the words based on the POS tagging of the words, so the lemmatized words will be more accurate.

f) Remove the stop word
Stop words are words that do not contain significant meaning when it is used to create a corpus. For example, are "the" and "to be" words, both do not provide a significant meaning to the context of the sentences. Table 1 shows the example of preprocessing result.

Table 1. Preprocessing Result

| Input | Output |
|---|---|
| My current bank deposit account interest rate has just been cut again. | current bank deposit account interest rate cut |
| Most people have a current account and most banks pay virtually no interest on this money. | people current account bank pays virtually interest money |

### 2.1.3 Create Word2vec Corpora

The corpora is developed using word2vec word embedding technique [14]-[16] on Google using data obtained from the content of Wikipedia articles. Word2vec is used because it has two layers of neural networks used to produce word embedding in a vector space. In vector spaces, words that share common contexts will converge in adjacent places [14].

There are two ways to create a corpus, such as the Continuous Bag of Words (CBOW) model and the Skip-Gram model [14]. The CBOW model predicts the words based on the given context whereas Skip-gram predicts words that surround the given word [14]. The preprocessed datasets are used as an input to Word2vec. Since this research does not have large datasets for training, Skip-gram model is used because it has a better solution in handling infrequent words than CBOW model. Skip-gram model is used with a hundred-dimensional vector and with window five words and minimum word appear ten times.

### 2.2. Determine the Result

Figure 3 shows the process of determining the result. We use testing data from Oxford English Dictionary and Yourdictionary.com to be preprocessed. Then, the sentence similarity with the corpora is calculated to determine the result.

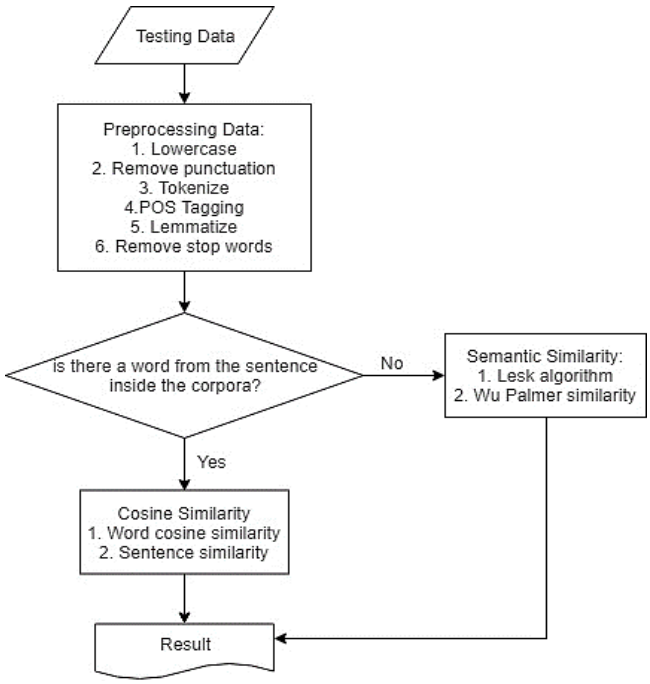


Figure 3. Determine the Result

#### 2.2.1 Testing Data

a) Oxford English Dictionary

The Oxford English Dictionary (OED) is the largest English dictionary widely used by people to search for word definitions or search for sentence examples from a word. Therefore, OED is used as testing data.

b) Yourdictiornary.com

Yourdictionary.com is a free online English dictionary that has many sample sentences, famous quotes, and audio pronunciations. In this dictionary, examples of sentences are made by internet users, so the data will have many sentence variations. Therefore, it is used to test the proposed method.

#### 2.2.2 Preprocessing Data

The preprocessing step for testing data is the same as preprocessing step for developing corpora.

#### 2.2.3 Similarity to Corpora

a) Cosine similarity

Cosine similarity is the calculation between two vectors with the result of an angle between them [17]. Cosine similarity produces results with intervals between -1 and 1. The formula for cosine similarity is;

$$cosine\ similarity = \cos(\theta) = \frac{A \cdot B}{\|A\|\ \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1)$$

where $A_i$ and $B_i$ are components of word2vec vectors $A$ and $B$, respectively.

b) Sentence similarity

Every word in a sentence except the ambiguous word itself are calculated using cosine similarity with the ambiguous word contained in the corpus [18]. The ambiguous word in the sentence and the word from sentence that is not in corpus will be given 0 value. The words from sentences that have been calculated using cosine similarity are then averaged.

$$sentence\ similarity = \frac{1}{n} * \sum_{i=1}^{n} x_i \quad (2)$$

where,

$n$ = number of words from sentence

$x$ = cosine similarity of the words from sentence with the ambiguous word

c) Determine the result

The meaning of the ambiguous word in a sentence is determined by the value of sentence similarity that has been calculated. If the value of sentence similarity to corpus one is higher than corpus two, then the meaning of the ambiguous word present in a sentence is as defined by corpus one and vice versa. For example, Table 2 shows the calculation with preprocessed input sentence "current bank deposit account interest rate cut" with the corpus 1 'bank' as a financial institution and the corpus 2 'bank' as geography.

Table 2. Cosine Similarity Result

| Word from sentence | Cosine similarity with word 'bank' in corpus 1 | Cosine similarity with word 'bank' in corpus 2 |
|---|---|---|
| Current | 0.838 | 0 (not in corpus) |
| Bank | 0 (ambiguous word) | 0 (ambiguous word) |
| Deposit | 0.949 | 0.983 |
| Account | 0.952 | 0 (not in corpus) |
| Interest | 0.925 | 0 (not in corpus) |
| Rate | 0.895 | 0.986 |
| Cut | 0 (not in corpus) | 0.992 |
| Sentence similarity | 0.651 | 0.423 |

### 2.2.4 Semantic Similarity

a) Lesk algorithm

Lesk algorithm is a classical algorithm for word sense disambiguation. In this paper, the simplified Lesk algorithm is used because it has a better performance [12]. This algorithm is shown in Figure 4. It calculates the overlapping words between the input sentence and the sentence from word definition and example in dictionary. In this case, Wordnet is used as the dictionary.

| | | |
|---|---|---|
| Input | | The bank can guarantee deposits will eventually cover future tuition cost because it invest in adjustable-rate mortgage securities. |
| bank[1] | Definition | a financial institution that accepts deposits and channels the money into lending activities |
| | Example | "he cashed a check at the bank"; "that bank holds the mortgage on my home" |
| bank[2] | Definition | sloping land (especially the slope beside a body of water) |
| | Example | "they pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents" |
| Output | bank[1] | |

Figure 4. Simplified Lesk Algorithm

Sentences that do not have a single word contained in the corpora are then used as the input into this algorithm. The output of this algorithm is one of the words in Wordnet and will be used in the next step.

b) Wu Palmer Similarity

Wu Palmer similarity [13] is one of many algorithm that measures the semantic similarity of two words based on the Wordnet tree.

The formula for calculating similarity using Wu Palmer is

$$wu\ palmer\ similarity = \frac{2 * Depth(LCS)}{(Depth(a) + Depth(b))} \quad (3)$$

where,
$LCS$ = Least Common Subsumer (parent of the two words searched)
$a$ = the first word
$b$ = the second word

The word resulted from Lesk algorithm then measured with the real meaning of ambiguous word in Wordnet using Wu Palmer similarity and the output score will be used to determine the result. For example, the output from Lesk algorithm is "slope", then the word "slope" measured with the word "bank" as a financial institution and word "bank" as geography in Wordnet.

c) Determine the result

The meaning of the ambiguous word in a sentence is determined by the value of Wu Palmer similarity that has been calculated. If the value of Wu Palmer similarity to corpus one is higher than corpus two, then the meaning of the ambiguous word present in a sentence is as defined by corpus one and vice versa

### 2.3. Performance Measure

To evaluate the proposed method, these following formulas are used

$$Precision = \frac{\left(\frac{TS_1}{(TS_1 + FS_1)} + \frac{TS_2}{(TS_2 + FS_2)}\right)}{2} \quad (4)$$

$$Recall = \frac{\left(\frac{TS_1}{(TS_1 + FS_2)} + \frac{TS_2}{(TS_2 + FS_1)}\right)}{2} \quad (5)$$

$$F1Score = \frac{(Precission\ x\ Recall)}{(Precission + Recall)}\ x\ 2 \quad (6)$$

$$Accuracy = \frac{TS_1 + TS_2}{(Total\ Data)} \quad (7)$$

where,
$TS_1$ = True prediction of the first sense
$FS_1$ = False prediction of the first sense
$TS_2$ = True prediction of the second sense
$FS_2$ = False prediction of the second sense

## 3. RESULTS AND ANALYSIS

In this paper, Python programming language is implemented to propose the method. To get the articles from Wikipedia, we use content function from Wikipedia python library. The nltk python library is used to preprocess the data from Wikipedia and gensim python library is used to create the word2vec corpora. The amount of the testing data we used can be seen in Table 3. Table 4 shows the experiment result without semantic similarity. Since there is no word from the sentence inside both corpora, the sentence similarity will have 0 value. Therefore, the precision, recall, and F1score value cannot be calculated. We can only calculate the accuracy.

Table 3. Testing Data

| Ambiguous words | Senses | Wikipedia Dataset (sentences) | Testing Data (sentences) |
|---|---|---|---|
| Bank | Financial Institution & Geography | 335 | 138 |
| Plant | Factory & Biology | 298 | 80 |
| Heart | Feeling & Organ | 369 | 40 |
| Average | - | 334 | 86 |

Table 4. Experiment Results of Cosine Similarity Without Semantic Similarity

| Ambiguous word | Unknown sentences | Accuracy (%) |
|---|---|---|
| Bank | 29 | 73.72 |
| Plant | 10 | 81.25 |
| Heart | 2 | 77.50 |
| Average | 13,6 | 77.49 |

Table 5 is the second results that presents the experiment result with semantic similarity. Since there is no 0 value of the semantic similarity, we can calculate the precision, recall, and F1score. As can be seen in Table 5, if we use semantic similarity when there is no word from the sentence inside both corpora, the accuracy result is improved by 8.02%.

Table 5. Experiment Result of Cosine Similarity with Semantic Similarity

| Ambiguous word | Precision (%) | Recall (%) | F1Score (%) | Accuracy (%) |
|---|---|---|---|---|
| Bank | 88.21 | 89.33 | 88.76 | 89.05 |
| Plant | 85.00 | 85.00 | 85.00 | 85.00 |
| Heart | 82.50 | 82.58 | 82.54 | 82.50 |
| Average | 85.23 | 85.63 | 85.43 | 85.51 |

## 4. CONCLUSION

This research proposes the use of Wikipedia and Word2vec to develop the corpora. The additional algorithm such as Lesk algorithm and Wu Palmer similarity are used to handle inexistent words in a corpus. The results of our proposed method to solve word sense disambiguation problems show an accuracy rate of 85.51% and the semantic similarity can improve the accuracy rate by 8.02%. For further research, the process for handling words from a sentence that are not in the corpora with cosine similarity is still lacking so that it can be developed to achieve better accuracy.

## ACKNOWLEDGEMENTS

The authors would like to thank to Institut Teknologi Sepuluh Nopember, *Direktorat Riset dan Pengabdian Masyarakat, Direktorat Jenderal Penguatan Riset dan Pengembangan*, the Ministry of Research, Technology, and Higher Education of Indonesia for financing the research.

## REFERENCES

[1] A. R. Pal, D. Saha, and S. K. Naskar, "Word sense disambiguation in Bengali: A knowledge based approach using Bengali WordNet," in *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 2017, pp. 1–5.

[2] N. Bouhriz, F. Benabbou, E. Habib, and B. Lahmar, "Word Sense Disambiguation Approach for Arabic Text," *IJACSA) Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 4, pp. 381–385, 2016.

[3] S. Gupta, A. Namavari, and T. O. Smith, "Word Sense Disambiguation Using Skip-Gram and LSTM Models," 2017.

[4] Q.-P. Nguyen, A.-D. Vo, J.-C. Shin, and C.-Y. Ock, "Effect of Word Sense Disambiguation on Neural Machine Translation: A Case Study in Korean," *IEEE Access*, vol. 6, pp. 38512–38523, 2018.

[5] B. S. Rintyarna, R. Sarno, and C. Fatichah, "Enhancing the performance of sentiment analysis task on product reviews by handling both local and global context," *Int. J. Inf. Decis. Sci.*, vol. 11, 2018.

[6] H. Sujaini, K. Kuspriyanto, A. Akhmad Arman, and A. Purwarianti, "A Novel Part-of-Speech Set Developing Method for Statistical Machine Translation," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 12, no. 3, p. 581, Sep. 2014.

[7] B. S. Rintyarna and R. Sarno, "Adapted weighted graph for Word Sense Disambiguation," in *2016 4th International Conference on Information and Communication Technology (ICoICT)*, 2016, pp. 1–5.

[8] N. Sharma, S. Kumar, and Dr. S. Niranjan, "Using Machine Learning Algorithms for Word Sense Disambiguation: A Brief Survey," *ISSN Int. J. Comput. Technol. Electron. Eng.*, vol. 2, no. 1, pp. 2249–6343.
[9] F. Hastarita Rachman, R. Sarno, and C. Fatichah, "Music Emotion Classification based on Lyrics-Audio using Corpus based Emotion," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 3, pp. 1720–1730, 2018.
[10] F. H. Rachman, R. Sarno, and C. Fatichah, "CBE: Corpus-based of emotion for emotion detection in text document," in *2016 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, 2016, pp. 331–335.
[11] S. Vijayarani, M. R. Janani, and A. Professor, "TEXT MINING: OPEN SOURCE TOKENIZATION TOOLS – AN ANALYSIS," *Adv. Comput. Intell. An Int. J.*, vol. 3, no. 1, pp. 37–47, 2016.
[12] P. Basile, A. Caputo, and G. Semeraro, "An Enhanced Lesk Word Sense Disambiguation Algorithm through a Distributional Semantic Model," *Proc. COLING 2014, 25th Int. Conf. Comput. Linguist. Tech. Pap.*, pp. 1591–1600, 2014.
[13] P. Sharma, R. Tripathi, and R. C. Tripathi, "Finding Similar Patents through Semantic Query Expansion," *Procedia Comput. Sci.*, vol. 54, pp. 390–395, Jan. 2015.
[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, 2013.
[15] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. Curran Associates Inc., pp. 3111–3119, 2013.
[16] T. Mikolov, W. Yih, and G. Zweig, "Linguistic Regularities in Continuous Space Word Representations," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 746–751.
[17] A. Huang, "Similarity Measures for Text Document Clustering," in *Proceedings of the New Zealand Computer Science Research Student Conference 2008*, 2008, pp. 49–56.
[18] D. Wali and N. Modhe, "Word Sense Disambiguation Algorithms in Hindi," 2015.

## BIOGRAPHIES OF AUTHORS

Farza Nurifan is now fourth year student of Informatics Department at Institut Teknologi Sepuluh Nopember. His current interests are in Text Mining and Iternet of Things.
E-mail: farzanurifan@gmail.com

Riyanarto Sarno received M.Sc and Ph.D in Computer Science from the University of Brunswick Canada in 1988 and 1992. In 2003 he was promoted to a Full Professor. His teaching and research interests includes Internet of Things, Process Aware Information Systems, Intelligent Systems and Business Process Management.
E-mail: riyanarto@if.its.ac.id

Cahyaningtyas Sekar Wahyuni received her bachelor degree from Information System of Universitas Brawijaya in 2018. She actively joined in many organizations, committees, and competitions. She has won several competitions in business plan, english speech, and english debate. Now, she is joining magister study at Department of Informatics in Institut Teknologi Sepuluh Nopember, Surabaya. Her current interests are in Process Mining and Business Process Management.
E-mail: cahyaningtyas.sekar.w@gmail.com