JITK (JURNAL ILMU PENGETAHUAN
DAN TEKNOLOGI KOMPUTER)

VOL. 6. NO. 1 AUGUST 2020
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v6i1.1344

BALINESE AUTOMATIC TEXT SUMMARIZATION USING GENETIC ALGORITHM

Cokorda Gde Abimanyu[1]; Ngurah Agus Sanjaya ER[2]; A. A. Istri Ngurah Eka Karyawati[3]

Department of Informatics
Udayana University
www.unud.ac.id
cokdabi@gmail.com, agus_sanjaya@unud.ac.id, eka.karyawati@unud.ac.id

Abstract— A summary contains the important idea of a text. However, summarizing a text requires one to read its entire content. In this study, text summarization is done automatically by applying a genetic algorithm to optimize the weight of five sentence features. The features include positive and negative keywords, the similarity between sentences and titles, the similarity between sentences, and cosine similarity. The collection of documents in this study are Balinese text stories. The summarization technique used is the extraction technique which eliminates unnecessary sentences, without changing the structure of the original sentence. The score of a sentence is generated by multiplying the feature value of each sentence by the weight of the feature. Summarization of the text is done by sorting the sentences based on the score. At the training stage, the best weight combination is chosen based on the average fitness value. Evaluation of the proposed method is carried out using 50 test data in the form of Balinese text stories. From the test results, it can be concluded that the fitness value of the feature weights is affected by the crossover and mutation rate of the genetic algorithm. Furthermore, accuracy is also influenced by the compression parameters used.

Keywords: Summarization, Genetic Algorithm, Balinese, Compression.

Intisari— *Intisari dari suatu teks dapat dibaca dari ringkasannya. Akan tetapi, peringkasan dokumen mengharuskan seseorang untuk membaca keseluruhan isi dokumen. Pada penelitian ini, peringkasan teks dilakukan secara otomatis dengan menggunakan algoritma genetika untuk mengoptimasi limat bobot fitur kalimat. Fitur-fitur tersebut meliputi kata kunci positif dan negatif, kemiripan antara kalimat dengan judul, kemiripan antar kalimat dan cosine similarity. Kumpulan dokumen pada penelitian ini adalah teks cerita berbahasa Bali. Teknik peringkasan yang digunakan adalah teknik ekstraksi dengan menghilangkan kalimat yang tidak penting, tanpa mengubah struktur kalimat aslinya. Skor suatu kalimat dihasilkan dengan cara mengalikan nilai fitur masing-masing kalimat dengan bobot fitur. Peringkasan teks dilakukan dengan mengurutkan kalimat berdasarkan nilai skornya. Pada tahap pelatihan, kombinasi bobot terbaik dipilih berdasarkan rata-rata nilai fitness. Pengujian terhadap metode yang diusulkan dilakukan menggunakan 50 data uji berupa teks cerita berbahasa Bali. Dari hasil pengujian dapat disimpulkan bahwa nilai fitness bobot fitur dipengaruhi oleh parameter crossover rate dan mutation rate. Disamping itu akurasi ringkasan juga dipengaruhi oleh parameter kompresi yang digunakan.*

Kata Kunci*: Peringkasan, Algoritma Genetika, Bahasa Bali, Kompresi.*

INTRODUCTION

We can understand the main idea of a document by simply reading a summary of the document. However, the conventional process of summarizing documents requires a lot of time because the reader must first read the entire content of the document [1]. Moreover, the summary result of this manual process will certainly be greatly influenced by the background knowledge of the reader. Both of these problems can be overcome if the summarization process can be done automatically. This would eliminate the need of spending time reading the entire text and also the knowledge requirement of the reader.

Previously, there have been many approaches developed for automatic text summarization systems on English [2] [3] and Indonesian [4] [5] documents. Automatic text summarization can be applied in various fields including scientific articles [4] and news [5], [6]. The techniques used to summarize the documents vary greatly, from artificial neural networks [3], deep learning [2], latent semantic analysis [5], or latent self-allocation [7]. In addition, the Genetic algorithm has been also applied to summarize Indonesian [8] as well as English [9] documents. In the case of text summarization, a Genetic algorithm can assist in optimizing the weights of the sentence features. The weight of each feature is necessary for the score calculation of the sentence. To find the optimal weight combination, the Genetic algorithm performs the selection, crossover, and mutation process. Once the summarization process is carried out, the Recall-Oriented Understanding for Gisting Evaluation (ROUGE-N) [10] is calculated to evaluate the result. ROUGE-N automatically compares the generated summary with the manual summary.

The automatic summarization of Balinese text documents has not attracted much attention. One of the reasons is the lack of reliable stemming [11], [12] for Balinese. In this research, automatic text summarization for Balinese texts is proposed using a Genetic algorithm. In the testing phase, the accuracy of the summary results will be tested using the ROUGE method. In addition, testing is also carried out to investigate the effect of the crossover rate and mutation rate parameters on the accuracy of the resulting summary results.

MATERIALS AND METHODS

Preprocessing (Tokenizing, Stopwords Removal, Stemming)

The text summarization process starts with preprocessing of the corpus which includes tokenizing, stop words removal, and stemming. The preprocessing phase begins with reading the Balinese text documents as input. Each document is then broken up into smaller parts (tokens) used for further processing. In this study, a token is a word in the Balinese language. The title and the content of the document will go through the same process and both will generate a collection of words.

The words which have been introduced as the result of the previous process will then be compared with the words in a stop words list. This list consists of predefined words that have been considered as "not important". The words are considered not important because their frequency of occurrences in the document are very high. Words that fall into this category include conjunctions, personal pronouns, and so forth. Words that are found in the stop words list eventually can be deleted and are not used for further processing.

Stemming is the process of finding the root of a word by eliminating all of the affixes which are attached to the word [11]. Affixes can consist of a prefix, a suffix, an infix, and a combination of prefixes (confix). Although there is a lemmatizer for Balinese [12], a complete Balinese vocabulary is required to get the best results. Therefore, in this study, we apply the Bastal algorithm [11] for stemming of the Balinese words. Stemming is achieved by applying a rule-based method in search of the root word. This algorithm uses a dictionary consisting of the Balinese words as the reference. Bastal algorithm classifies affixes (wewehan), i.e. prefixes (pangater), suffixes (pangiring), and nasal sounds (anusuara) based on Balinese morphological rules. The following are the steps of the Bastal algorithm for stemming Balinese:

1. Check the number of letters then determine whether the word currently processed is found in the root word dictionary. If the word consists of 2 (two) or fewer letters, and/or is found in the root word dictionary, then it is assumed that the word is a root word. The algorithm can be stopped because the root word has been found.
2. Delete the prefix ("*a-*") in the word and if the word is found in the dictionary then stop the process. Otherwise, the previously deleted prefix is put back to the word.
3. Delete the following prefixes ("*ka-*", "*sa-*", "*di-*", "*pa-*", "*pi-*", "*ma-*") on the word and if the word is found in the dictionary then stop the process. If not then do step 4 and all the deleted prefixes are put back to the word.
4. Remove the following nasal sound ("*ng-*", "*ny-*", "*n-*", "*m-*") on the word and replace it according to the rules. If the word is found in the dictionary then stop the process. If not, the previously replaced nasal should be put back to the word.
5. Delete the suffix ("*-a*") in the word and if the word is found in the dictionary then stop the algorithm. Otherwise, the deleted suffix is put back and proceed to step 3.
6. Delete the suffix ("*-an*", "*-in*", "*-ne*") or "*-ng*" in the word and if the word is found in the dictionary then stop the algorithm. If not, then do steps 2 until 4 and proceed to step 6a.
    a) If "*-ng*" has been deleted and the last letter is "*a*", then delete "*a*". If the word is found in the dictionary then stop the process. If not, then do steps 2 until 4 and proceed to step 6b.
    b) All deleted suffixes ("*-an*", "*-in*", "*-ne*") or "*-ng*" and "*a*" are returned.
7. If all steps (1-6) have been carried out but the root word has not been found, then the initial word will be assumed as the root word.

Sentence Scoring

As have been previously discussed, the features of a sentence used in this study include positive keywords (f_1), negative keywords (f_2), words in sentences that appear in the document title (f_3), similarities between sentences ($f4$) and cosine similarity (f_5).

$$f_1(s_i) = \frac{No.\ of\ occurrences\ of\ positive\ keywords}{Total\ number\ of\ words\ in\ s_i} \quad \text{..........} \quad (1)$$

$$f_2(s_i) = \frac{No.\ of\ occurrences\ of\ negative\ keywords}{Total\ number\ of\ words\ in\ s_i} \quad \text{.......} \quad (2)$$

Positive keywords and negative keywords are defined as the most frequent and least words which appear in a document. In cases where a document has more than one word which have the most or least frequencies of occurrences, the positive and negative keywords are randomly selected from those words. After the positive and negative keywords for a document have been determined, the value of positive keywords (f_1) and negative keywords (f_2) features of a sentence (s_i) can be calculated using equations (1) and (2).

$$f_3(s_i, s_j) = \left| \frac{Words\ in\ s_i \cap words\ in\ s_j}{Words\ in\ s_i \cup words\ in\ s_j} \right| \quad \text{..............................} \quad (3)$$

Feature f_3 of a sentence states the number of words that are found both in the sentence and also in the title of the document. $f3$ is calculated using equation (3) where s_j is the words in the title.

$$f_4(s_i) = \left| \frac{Words\ in\ s_i \cap_{j \in D-i} Words\ in\ s_j}{\cup_{j \in D} Words\ in\ s_j} \right| \quad \text{........................} \quad (4)$$

Feature f_4 expresses the similarity between the sentence s_i with other sentences in the same document D and is calculated using equation (4). From equation (4), it can be seen that the similarity between the sentence s_i with other sentences in document D is expressed as the number of words that appear on s_i and also feature in other sentences in document D.

$$f_5 = sim(s_i, s_j) = \frac{\sum_m t_{im} t_{jm}}{\sqrt{\sum_m t_{im}^2} * \sqrt{\sum_m t_{jm}^2}} \quad \text{........................} \quad (5)$$

Feature f_5 is used to calculate the similarity between sentences based on the cosine similarity in the vector space model. Equation (5) is used to calculate the similarity between the sentence s_i and s_j. s_j here is a collection of all sentences in document D other than s_i (s_j = D - s_i). t_{im} and t_{jm} state the weight value of each word in s_i and s_j. The weight of each word is calculated using all the words contain document D.

After each feature of a sentence is calculated using equation (1) - (5), the score for the sentence is then obtained from equation (6).

$$Skor\ (S) = \sum_{i=1}^{n} w_i * f_i \quad \text{..} \quad (6)$$

w_i in equation (6) shows the weight given to each feature f_i. If each feature is considered to have an equal contribution to the score of a sentence, then the weight of each feature can be ignored. The contribution of each feature is of course not equal, thus determining the weight of the feature will greatly affect the score of each sentence. The score of each sentence will ultimately affect the results of the summarization process. The weight of each feature corresponds directly to the level of accuracy of the summary. Therefore, determining the weight combination for the sentence features cannot be done manually and must be carefully chosen to optimize the accuracy. The problem of finding the appropriate combination of weight for the sentence features while optimizing the accuracy of the summary can be solved by leveraging the Genetic algorithm.

Generating Combinations of Feature Weights with Genetic Algorithm

Genetic Algorithm is a searching algorithm. The algorithm is based on natural system mechanisms, namely genetics and natural selection. The Genetic algorithm is a computer program that simulates the process of evolution, by producing chromosomes from each population randomly. These chromosomes are allowed to multiply following the laws of evolution and are expected to produce prime or better chromosomes.

Terms such as genes/individuals, chromosomes, and populations that are used in the Genetic algorithm must be defined first before the process of finding the most optimal feature weights can be done. Genes are the smallest unit in the Genetic algorithm which in this study represents the weight of features (w_i for i = 1, ..., n). Chromosomes are a combination of genes that form certain values that represent solutions to a problem. In this study, a chromosome is the weight representation of the five features used to calculate the score of a sentence. Chromosomes are expressed by a collection of 5 genes namely w_i for i = 1, ..., 5) and are expressed as w_1: w_2: w_3: w_4: w_5. Given $w_1 = 0.11$; $w_2 = 0.21$; $w_3 = 0.32$; $w_4 = 0.12$; $w_5 = 0.24$ then the chromosome is expressed as 0.11: 0.21: 0.32: 0.12: 0.24 where the ":" sign is used to separate genes from one another. The population is a collection of individuals/chromosomes which must be generated at the beginning of the process. The technique used to generate the initial population in this study is random generation.

A distinctive characteristic that distinguishes the Genetic algorithm from other algorithms is the selection process carried out within the initial population to determine the individuals used to produce the next generation. Selected individuals are of course individuals who can maximize the fitness value in the training data used. In this research, the selection method used is the roulette wheel. At the beginning of the selection process with the roulette wheel method, the total fitness (t_f) of the population is calculated by adding up the fitness values (f_i, where i indicates the *1*st to n^{th} individuals) of all individuals. Each individual then has a selection probability (P_i) which is the fitness value being normalized by t_f. After that, each individual is then sorted (from the smallest to the largest) based on the probability values. A random number R is used to separate the order of individuals in the population which satisfies $C[k\text{-}1] < R < C[k]$, where C is the cumulative probability value of P_i while k denotes the number of iteration. For k-th iteration when generating new chromosomes, chromosomes are selected from the sorted chromosome list that satisfies the cumulative probability value $C[k] > R$.

After the best chromosome is determined for the k-th iteration, then it is used to produce new chromosomes. The new chromosome can be produced in two ways, namely crossover and mutation. Crossover can only be done if a generated random number produces a value smaller than a parameter called the crossover rate (CR). The type of crossover used in this research is the one-point crossover. Given two chromosomes (p_1 and p_2), to produce a new chromosome (p_3), a cut-off point k is determined first. p_3 is then produced by taking p_1 genes (from the 1st through k-1th position) and p_2 genes (from k^{th} position through n^{th}). The alternative way of producing new chromosomes is through mutation. In mutation, some changes are applied in the value of one or several genes in a chromosome. The position of the gene whose value will be changed as well as the replacement value for the gene is determined randomly. The number of chromosomes which undergo the process of mutation in a population is controlled by the mutation rate (MR) parameter. One of the experiments in this research is aimed at finding out the CR and MR values which produce the best combination of feature weights so that the summarization process achieves the maximum accuracy.

The final result of the feature weight generation process leveraging Genetic algorithm is the production of new chromosomes through the process of crossover or mutation. Chromosomes that are produced either by crossover or mutation are called offspring.

Determining the Optimal Feature Weight

The implementation of a Genetic algorithm to assist in finding the combination of feature weights will produce many solutions that are referred to as offsprings. To determine the best offspring, i.e. the combination of feature weights that provides the optimal level of accuracy of the summary results, it is necessary to define the fitness of each offspring.

Fitness is a measure of goodness of a solution which has been generated using the Genetic algorithm. To calculate the fitness value of an offspring, we must first test the combination of feature weights stated by it to summarize each document in the corpus. When the process is completed, the summary results will then be compared with the references (manual summary created by an expert which are prepared beforehand) to get the ROUGE value. From both summaries, i.e. the result of the process explained earlier as well as the reference summary created by the expert, we first generate the n-grams. We can then calculate ROUGE (Recall-Oriented Understanding for Gisting Evaluation) as the number of n-grams contains in both summaries. This ROUGE-N value is calculated using equation (7).

$$ROUGE-N = \frac{Number_{Matched}(gram_n)}{Number(gram_n)} \quad\text{.........................}(7)$$

In equation (7), n expresses the length of n-gram, the $Number_{Matched}(gram_n)$ indicates the number of n-grams that are contained in the reference summary and the automatically generated summary while the $Number(gram_n)$ denotes the n-gram in the reference summary. In this research, the value of n in *ROUGE-N* is equal to 1 (unigram) because it has been proven that the unigram version of *ROUGE-N* has a positive correlation with the evaluation of manually created summaries. In this article, we refer to *ROUGE* as the unigram version of *ROUGE-N*.

The fitness value of the offspring is expressed by the average *ROUGE* value for a set of documents summarized using a combination of the weight of the features stated by the offspring. The fitness value is calculated using equation (8), where $ROUGE_i$ states the *ROUGE* value for the i-th document, and n states the number of documents in the corpus.

$$Fitness = \frac{\sum_{i=1}^{n} ROUGE_i}{n} \quad\text{...}(8)$$

In this research, the process of searching for the best offspring in the corpus is then performed using the *k-fold* cross-validation method. *k-fold* cross-validation is a method generally used to find out the average success of a system by conducting repeated trials but with different input [13].

The Summarization Process

Summarizing a document begins by extracting the five features (f_1, f_2, f_3, f_4, and f_5) for each sentence. The score for a sentence is then calculated as the sum of the five features multiplied with its corresponding weight. The sentences are then sorted based on the score in descending order. Next, the summary of a document is automatically obtained by taking the top n sentences. The value of n here is determined using equation (9).

$$n = |JKD| - (COMP * |JKD|) \quad \text{.....................................} (9)$$

JKD shows the number of sentences in the original document (before summarization) and $COMP$ denotes the summary compression percentage. From the result of research conducted by Radev et al. [14], it is believed that the number of sentences contains in the summary of a document may not be larger than 50% of the number of sentences in the original document. From this statement, the value of n in equation (9) must be greater than half of the number of sentences in the original document. Therefore, the number of sentences to be omitted, which is stated by $COMP$ in equation (9), must be in the range of 50% < $COMP$ <100%. The effect of $COMP$ on the accuracy value of the resulting summary is one of the experiments conducted in this study.

RESULT AND DISCUSSION

The data used in the performance evaluation of this research are in the form of Balinese folklore stories. The number of documents collected is 150. To be able to measure the accuracy of the summaries that are generated automatically by our proposed method, each of the 150 documents has to be summarized manually by a Balinese linguist and are used as the reference summaries. The manual process of summarization conducted by the expert also applies the extractive method. In this approach, the expert only removes sentences that are considered as not important without changing the structure of the sentence. This is done to ensure a fair comparison between the automatically generated summaries and the references. The data is then divided into two, i.e. 100 and 50 documents as training and testing data respectively. The training data is then used to find the best combination of feature weights (the first experiment), while the test data is used to determine the effect of the compression level on the summary results (the second experiment).

Tabel 1. Testing Results for Combination of *CR* and *MR*

No	*CR*	*MR*	*Fitness Average*	*Variance*	*Standard Deviation*
1.	0.4	0	0.841	0.0014	0.0357
2.	0.35	0.05	0.873	0.0002	0.0159
3.	0.3	0.1	0.887	0.0001	0.011
4.	0.25	0.15	0.867	0.0004	0.0199
5.	0.2	0.2	0.822	0.0022	0.0446
6.	0.15	0.25	0.836	0.0011	0.032
7.	0.1	0.3	0.846	0.0013	0.0347
8.	0.05	0.35	0.839	0.0006	0.0244
9.	0	0.4	0.814	0.0026	0.0492

The first experiment conducted in this study was to find the combination of CR and MR parameters that produced the best fitness average. From the first experiment, we choose the most optimal features of weight combination. The best combination of CR and MR is then used in the second experiment. Table 1 shows the result of the first experiment. In the first experiment, we determine 9 combinations of CR and MR parameters. For each combination of parameters, we conduct ten trials. In each trial, a different amount of offspring is generated (10, 20, 30, ..., 100). The best offspring from each trial is then used to summarize 50 Balinese folklore documents and calculate the fitness value. The average fitness, variance, and standard deviation for each combination of CR and MR parameters are shown in columns 3, 4, and 5 of Table 1 respectively.

From Table 1 we find that the best combination of CR and MR is 0.3 and 0.1 respectively with an average fitness value of 0.887. The standard deviation and variance values from the combination of these parameters are close to 0, thus it can be concluded that the error rate is very small.

Tabel 2. The Best Combination of Features Weight

w_1	w_2	w_3	w_4	w_5	*Fitness*
0.008	0.373	0.321	0.234	0.060	0.901

Aside from searching for a combination of CR and MR parameters that produces the best average fitness values, in each of the trials in the first experiment we also take note of the combination of features with the best fitness. Table 2 shows the combination of features weight that produces the highest fitness values. From Table 2 it can be seen that the best weights for features f_1, f_2, f_3, f_4 and f_5 are 0.008; 0.373; 0.321; 0.234; 0.06 which yield a fitness value of 0.901. This weight combination is then used in the second experiment.

The purpose of the second experiment is to determine the effect of the compression level on the accuracy of the automatically generated summary of documents. In this second experiment, we use four levels of compression ranging from 60% to 90% with an increase of multiple 10. The accuracy of the summary results obtained based on changes in the compression rate is calculated using equation (7). The average accuracy for the compression rate of 60% up until 90%, can be seen in Table 3.

Tabel 3. Average Accuracy for Different Levels of Compression Rate

No.	Compression Rate	Average Accuracy
1.	60%	77.3%
2.	70%	62.7%
3.	80%	41.5%
4.	90%	21.5%

From the experiment result, it can be seen that increasing the level of compression causes the average accuracy of the summary results to decrease. Table 3 shows that when the compression rate is at 60%, it produces the largest average accuracy of the summary results (77.3%) while at a 90% compression rate the accuracy of the summary results is only 21.5%. This is understandable because with a high level of compression (90%), the number of sentences in the summary results will be increasingly low. The reference summaries produced by the expert have at least 2-3 sentences for each paragraph, whereas at a high level of compression the number of sentences in each paragraph for the automatically generated summaries is only a few sentences. Moreover, equation (7) is based on the number of *n*-gram contain in both summaries (the automatically generated summary and the reference summary). At high compression levels, this number is increasingly smaller while the number of *n*-grams in the reference summary is fixed. The decrease in the number of *n*-grams that are found in both summaries will naturally cause the accuracy of the summary results to decrease.

CONCLUSION

From the results of experiments that have been carried out, several conclusions can be drawn. The proposed methodology can generate summaries for Balinese documents automatically. Summarizing is carried out with an extractive method where we first extract five features from each sentence on the document. We then calculate the score of the sentence where the weight of each feature is generated using the Genetic algorithm. Next, we sort the sentences based on their score and finally determine the number of *n* sentences that have the highest fitness value. These sentences are then used as the result of the summarization process. The combination of *CR* and *MR* parameters that produces the best average fitness is 0.3 and 0.1. Furthermore, the feature weights (w_1, w_2, w_3, w_4, w_5) that produce the highest fitness value (0.901) are 0.008; 0.373; 0.321; 0.234; 0.06. The compression value has a negative effect on the average value of the accuracy of the summary results. The increase in the compression rate (the number of sentences omitted), the average value of the accuracy of the summary will decrease. In this research, the best compression rate is 60% with an average accuracy of 77.3% for 50 test data.

REFERENCES

[1] J. S. Saputra and M. Fachrurrozi, “Peringkasan Teks Berita Berbahasa Indonesia Menggunakan Metode Latent Semantic Analysis (LSA) dan Teknik Steinberger & Jezek,” 2017.

[2] M. Yousefi-Azar and L. Hamey, “Text summarization using unsupervised deep learning,” *Expert Syst. Appl.*, vol. 68, pp. 93–105, 2017, doi: https://doi.org/10.1016/j.eswa.2016.10.017.

[3] R. Nallapati, B. Xiang, B. Zhou, W. Question, A. Algorithms, and Y. Heights, “Sequence-To-Sequence Rnns For Text Summarization,” *Iclr*, pp. 4–7, 2016.

[4] D. Anggraini and L. Wulandari, “Peringkasan Teks Artikel Ilmiah Berbahasa Indonesia Menggunakan Teknik Ekstraktif dan Fitur Kalimat Untuk Dokumen Tunggal,” *Semin. Nas. Rekayasa Komput. dan Apl.*, pp. 126–130, 2015.

[5] M. Y. Saputra, Jerry. Fachrurrozi, “Peringkasan Teks Berita Berbahasa Indonesia Menggunakan Metode Latent Semantic Analysis (LSA) dan Teknik Steinberger & Jezek,” *Comput. Sci. ICT*, vol. 3, no. 1, pp. 215–219, 2017.

[6] M. Mustaqhfiri, Z. Abidin, and R. Kusumawati, “Peringkasan Teks Otomatis Berita Berbahasa Indonesia Menggunakan Metode Maximum Marginal Relevance,” *Matics*, 2012, doi: 10.18860/mat.v0i0.1578.

[7] E. Y. Hidayat, F. Firdausillah, K. Hastuti, I. N. Dewi, and Azhari, “Automatic text summarization using latent drichlet allocation (LDA) for document clustering,” *Int. J. Adv. Intell. Informatics*, vol. 1, no. 3, pp. 132–139, 2015, doi: 10.26555/ijain.v1i3.43.

[8] Aristoteles, “Penerapan Algoritma Genetika pada Peringkasan Teks Dokumen Bahasa Indonesia,” *Semirata FMIPA Univ. Lampung*, pp. 29–33, 2013, [Online]. Available: http://jurnal.fmipa.unila.ac.id/index.php/semirata/article/download/703/523.

[9] K. Jezek and J. Steinberger, “Automatic Text Summarization (The state of the art 2007 and new challenges),” *Proc. Znalosti*, pp. 1–12, 2008.

[10] I. P. G. H. Suputra, “Peringkasan teks otomatis untuk dokumen bahasa Bali berbasis metode ektraktif,” *J. Ilm. Komput.*, vol. X, no. 1, pp. 1–6, 2017.

[11] I. P. M. Wirayasa, I. M. A. Wirawan, and I. M. A. Pradnyana, “Algoritma Bastal: Adaptasi Algoritma Nazief & Adriani Untuk Stemming Teks Bahasa Bali,” *J. Nas. Pendidik. Tek. Inform.*, vol. 8, no. 1, p. 60, 2019, doi: 10.23887/janapati.v8i1.13500.

[12] I. G. A. P. Arimbawa and N. A. S. ER, “Lemmatization in Balinese Language,” *JELIKU - J. Elektron. Ilmu Komput. Udayana*, vol. 8, no. 3, pp. 235–242, 2020, [Online]. Available: https://ojs.unud.ac.id/index.php/JLK/article/view/51892.

[13] E. S. Y. Pandie, “Sistem Informasi Pengambilan Keputusan Pengajuan Kredit Dengan Algoritma K-Nearest Neighbour (Studi Kasus: Koperasi Simpan Pinjam),” 2012.

[14] D. R. Radev, E. Hovy, and K. McKeown, “Introduction to the Special Issue on Summarization,” *Comput. Linguist.*, vol. 28, no. 4, pp. 399–408, 2002, doi: 10.1162/089120102762671927.

Deliberately left blank