Indonesian News Articles Summarization Using Genetic Algorithm

Nurul Khotimah and Abba Suganda Girsang

***Abstract*—Extractive text summarization consists of selecting the most important sentences from the original text. By summarizing the contents of the article, readers might be able to understand the article more easily and faster than reading the entire article. The process of summarizing involves gathering as much as possible of the information and presenting only the most important details as succinctly as possible. To solve that problem, a genetic algorithm will be adopted to extract sentences as a summary. The summarization process is considered as an optimization problem where the optimal summary is selected from a series of sentences from the original document. Genetic algorithm used to optimize sentence selection to obtain a summary that represents the main idea of the source document where the compression rate determines the number of sentences selected as summary. To represents the text and capture the interconnects between sentences, a graph will be constructed and given a weight with PageRank score. 60 news articles in Bahasa Indonesia from IndoSum are used as a dataset. To evaluate how good the results are, ROUGE-1 and cosine similarity are calculated to compare the summary generated by the system and reference summary. This study also set up 5 comparisons to other methods such as SumBasic, LexRank, LSA, TextRank, and KLSum. Evaluation results yield better summary results compare to other methods with the average ROUGE-1 score 0.641 on recall and cosine similarity 0.625 for compression rate of 30%.**

***Index Terms*— automatic text summarization, extractive summarization, genetic algorithm, news article.**

I. Introduction

THE amount of information on digital platforms such as e-Newspaper, journal articles, and data from social media are rapidly growing. For example, e-Newspaper as mass media that provides information about important and recent events automatically will increase day by day. Nowadays peoples can easily get information update through online news. However, read the whole news article needs quite a long time because sometimes the article consists of several pages that are difficult to get what the main idea of the news is given. Considering that, there is a process to present a short version of the original document computationally which contains the main idea of the document called automatic text summarization. By summarizing news articles, readers can be helped to obtain information more easily and determine whether they will read a whole article or not.

There are two types of approaches for automatic text summarization generally, extractive and abstractive. Extractive summarization method works by determining important sentences of the text and selecting them as a summary. That approach depends on sentences from the original text only. In contrast, abstractive summarization method expresses the ideas of the source documents using different words [1].

Automatic text summarization was introduced by P.Baxendale in 1958 with Positional's method which extracts the first and last sentence as a summary [2]. In the same year, Luhn’s method came up by selecting sentences with the highest concentrations of salient content terms [3]. Still from the statistical approach, Edmundson’s method came up with extract summary by scoring sentences using 4 features such as position of sentence, word frequency, cue words, and document structures [4]. After that, there are several methods appeared to extract the important sentences from document, for example TextRank [5] , LexRank [6], SumBasic [7], Latent Semantic Analysis [8], Term Frequency-Inverse Document Frequency [9], Graph [10], Centroid-Based [11]. Ant Colony [12] and Genetic Algorithm [13].

There are numerous studies in automatic text summarization. Most of them are investigating and exploring techniques for English language. However, there is little ongoing research in Indonesian language text summarization field. One research from Prasetyo et al., they implemented MEAD and modified IDF Dictionary method to summarize online news. MEAD uses centroid method to determine the importance of each sentence in a text document [14]. Later in 2012, Aristoteles et al., implemented genetic algorithm for text feature weighting using 11 features to investigate which are features have the best performance. In short, 4 features (positive keywords, sentence centrality, sentence resemblance to the title, and sentence semantics) show the best performance result [15]. The next research used semantic analysis approach to obtain the similarity between sentences by calculating the vector values of each sentence with the title [16]. Most of the research works focus on single document summarization used a statistical-based method for extracting sentences. The problem in extractive summarization is how the algorithm can extract important sentences that represent the contents of the document and in extractive summarization which

Manuscript received March 22, 2021; revised September 24, 2021.

Nurul Khotimah is a master student at Computer Science Department, BINUS Graduate Program-Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480 (email: nurul.khotimah001@binus.ac.id)

Abba Suganda Girsang is a Lecturer at Computer Science Department, BINUS Graduate Program-Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480 (corresponding author, email: agirsang@binus.edu).

often select sentences based on the highest score sometimes has sentence redundancy because most sentences that have the highest score tend to have the same information.

In this research, our work focused on Indonesian which is the official language of Indonesia. Research of automatic text summarization for Indonesian are still developing. Currently, reviewed papers of Indonesian summarization yet shows outstanding result. All the study shows different evaluation result according to research methodology conducted by the author, it means there is still room for a researcher to improve in automatic text summarization field.

To our knowledge, a combination of the graph method and a metaheuristic approach using a genetic algorithm has not been investigated for extractive summarization of a single Indonesian language document. In contrast for English documents, genetic algorithm has been done to solve automatic text summarization problem. Genetic algorithms were used to improve cohesion in extracted sentences in forming summaries [13] [17] . Genetic Algorithm (GA) was introduced by Holland wherein his research it was proven that GA was the most powerful optimization technique in finding solutions [18]. Genetic Algorithm (GA) is an algorithm based on biological evolutionary mechanisms. Genetics takes the best value out of a random selection of several possibilities. The summary results are obtained from the best individual scores. A sentence can be likened to a chromosome that will form an individual, then through a genetic selection process, the best individual will be taken as a summary sentence. Fitness in GA can find sentences with more optimal weights than the plain text summarization method.

Therefore, this study aims to propose a graph-based summarization algorithm method that used a genetic algorithm for optimizing the sentence selection to get a summary that represents a fair amount of the main idea from the source document where the number of sentences depends on the compression rate. Our proposed summarization method has several benefits. Firstly, this method is unsupervised learning which requires no training data. Secondly, construction of graph will capture relationship between sentences and redundant information.

The rest of this paper is structured as follows. Section 2 explains the related works. Section 3 explains the proposed method. Section 4 is about the results and discussion. Finally, the conclusions and future works are presented in Section 5.

II. Related Work

Automatic text summarization has become a popular research topic in the past few years. R. Alguliyev et al. [19] proposed a two-stages sentence selection model based on clustering and optimization techniques called COSUM. To find all topics in a text, k-means method is used to cluster the set of sentences. Then, an optimization model is applied to select important sentences from clusters. This model optimizes the harmonic mean of the objective functions of the sentences of the summary. An adaptive differential evolution algorithm with novel mutation strategy is developed to solve the optimization problem. This study shows that a combination of clustering and optimization approaches, also a combination of optimization with the graph-based approaches are more promising directions for automatic document summarization. In [20] proposed statistical method to perform an extractive summarization on single document. Weighted frequency of word is calculated by dividing frequency of the keywords by maximum frequency of the keywords. Summarizer will extract the high weighted frequency sentences in order to find summary of a document and the extracted summaries are converted into audio form.

Mohd et al. [21] proposed method based on the distributional hypothesis to capture the semantics of the text. Clustering algorithm is used for grouping semantic similar sentences. Next, top sentences from each group are extracted and retrieved by ranking algorithm as summary. This study shows that a semantic model can reduce redundancies in the input source.

S.Sehgal et al. [22] worked on extractive based summarization using TextRank algorithm. A graph is constructed with nodes for each sentence in a document and edges between sentences based on the number of words in common between two sentences, by calculating the number of words in common between two sentences. By considering the similarity between title and sentence, this study modified the function by adding a similarity score between sentence and title as cumulative score of each sentence. El-Kassas et al. [23] also proposed an extractive automatic text summarization based on graph-based method called EdgeSumm that combines a set of algorithms. The first algorithm creates a new graph model to represent the source document. The second and third algorithms search the resulting text graph for the sentences contained in the candidate summary. If the results show that the candidate summary still exceeds the limits required by the user, the fourth algorithm will be running to select the most important sentences. This study has shown that using the "sentence order" ranking criteria in the post-processing phase gives the best evaluation results and better summaries among the various ranking criteria. Similarly, [10] used graph-based method to represent sentence as node and relation between two sentences as edge by calculating the concept ratio derived from ontology of each sentence and combined with the distance from WMD score. Then to extract the summary, PageRank algorithm is applied for evaluating the valuable sentences. A.El-Refaiy et al. [24] proposed a a new unsupervised algorithm for extractive summarization for a single document. The algorithm uses Mean Shift Clustering algorithm to enhance the obtained summary, reduce redundancy and get more coherent sentences.

A genetic algorithm-based sentence extraction for text summarization method has been developed by the researchers. Nandhini and Balasundaram [17] used GA to extract the optimal combination of sentences while balancing the informative score and sentence similarity that increase readability through sentence cohesion. García-Hernández and Ledeneva [25] also proposed GA for optimizing step of sentences selection based on word frequency. They designed a fitness function based on two factors: most frequent words and sentence position.

Another method belongs to Meena and Gopalani [26], they proposed GA to determinant of the optimal weights on the text features. Those features are composed for fitness function calculation. This study showed, iterations of GA is the strength for finding optimal weights.

The methods mentioned above are the methods used for English text documents. However, the study of automatic text summarization for Indonesian is little ongoing research. Prasetyo et al. [14] created an application, namely SIDoBI, This application used MEAD which is centroid method based to determine the importance of each sentence in a text document. Aristoteles et al. [15] investigated text feature weighting using 11 features. All the features are used in training of GA model to obtain the appropriate weight combination for every feature. Similarly, [27] proposed a method based on sentence features scoring by Latent Dirichlet Allocation and GA for determining sentence feature weights.

Christian et al. [9] implemented TF-IDF algorithm to extract the summary. Sentences will be sorted in descending order by its value. Three to five sentences with the highest TF-IDF value are chosen as a summary. The other study by Gunawan et al. [28] They introduced TextTeaser algorithm for text in Indonesia language that calculates four elements, such as title feature, sentence length, sentence position and keyword frequency. This method will calculate word score based on its appearance in an article and selects sentences that possess best score among others. Further, Lucky and Girsang [12] implemented Multi-Objective Ant Colony Optimization for summarizing comments on Twitter. An undirected graph presented to build relation between sentences. The best solution to generate short and important comments determined by MOACO algorithm from the construction of undirected graph based on the required summary size.

III. PROPOSED METHOD

The proposed method of genetic algorithm for automatic news summarization consists of some steps which are described in Fig. 1.The number of documents used as many as 60 documents in Indonesian language from IndoSum dataset. The document summarizes by compression rate of 10%, 20% and 30%.

A. Data Collecting

The dataset of Indonesian news articles is retrieved from IndoSum dataset [29]. IndoSum is a corpus dataset for automatic text summarization of Indonesian documents taken from Indonesian news portal. In particular, 60 news articles of different length which grouped into 6 different topics (Entertainment, Inspiration, Sport, Showbiz, Headline, and Tech) are used for this study. Each document in the IndoSum data set is supplied with title, category, and gold standard summaries of human-generated. The IndoSum dataset is stored in the .jsonl (json line) file format, so the file is needed to be converted into a data frame format before is being used.

B. Data Preprocessing

Each sentence in IndoSum dataset has passed the

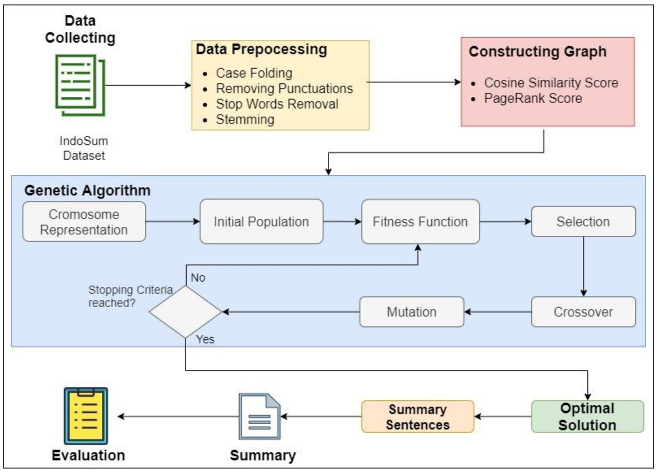


Fig. 1. Proposed method.

tokenization process. Furthermore, the words in each sentence will undergo a case folding process by replacing all letters in the document into lowercase. Then, continue for removing punctuation such as period (.) , exclamation points (!), question marks (?) and others. After that, top words removal is done by removing stop words or unimportant words using [30]. These words include article, prepositions, conjunction such as 'sebuah', 'dan', 'atau', 'di', etc. The last, the words in each sentence are stemmed into their stem, base, or root form using a Python library called Sastrawi.

C. Constructing Graph

Document to be summarized can be represented by a graph consisting of nodes and edges. Nodes represent the sentences, and the edges represent the similarity between those sentences. Cosine similarity measures is used in this study to determine how similar the documents are irrespective of their size. The cosine similarity is described as the division between the dot product of vectors and the product of the Euclidean norms or magnitude of each vector. In this paper, the sentences of the text are represented by vectors to get a similarity. The similarity of the sentence can be calculated using formula as follow in Eq. (1).

$$cos(A,B) = \frac{\sum_i A_i B_i}{\sqrt{\sum_i (A_i)^2} \cdot \sqrt{\sum_i (B_i)^2}} \quad (1)$$

where A_i is the attribute of vector A and B_i is the attribute of vector B. The cosine similarity of two sentences will range from 0 to 1. If the value closer to 0 indicates that the two sentences have less similarity and 1 indicate that both of sentences are same. Two sentences are linked if their similarity is above 0 and less than 0.8. The limit of 0.8 is used to ensure sentences that are similar not mutually connected to reduce redundancy in generated summaries. Meanwhile, if the value limit is too small it can cause fewer nodes to be connected so that the resulting solution is also less. The graph representation of a sample document is shown in Fig. 2.

After the graph is formed, each node or sentence in the graph is given the PageRank weight. PageRank was designed for web link analysis. PageRank determines the importance of a node within the graph, based on information drawn from the graph structure.

PageRank is used by Google to determine the level of importance of a web page. PageRank generated a matrix that user will move from page to another. In the case of text documents, PageRank calculations for each sentence can be

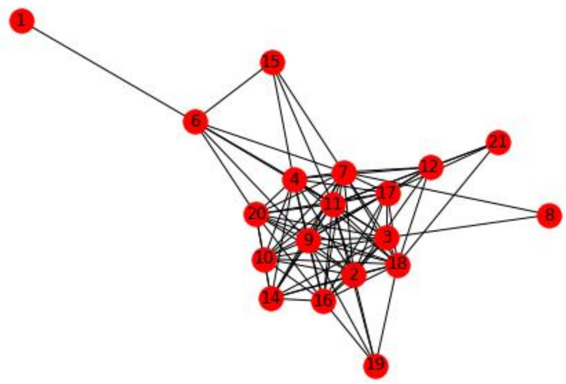


Fig. 2. Graph representation.

done by utilizing the graph and value of cosine similarity as the weight for the edge which connecting each node/sentence. The PageRank value is calculated by walking the graph randomly and then calculates the rank of a certain node by summing the PageRank value of nodes pointing to it, then divide it by the number of edges of its neighbors. The PageRank computations were carried out as iterations until the value was converged or didn't change anymore. The PageRank calculation formula is described in Eq. (2).

$$PR(S_i) = \frac{1-\alpha}{NodeCount} + \alpha \sum_{S_j} \in Neighbors\ S_i \frac{PR(S_j)}{CountEdge(S_j)} \quad (1)$$

In this formula, $PR(S_i)$is the PageRank score for sentence S1calculated by summing PageRank from each neighbor $PR(S_j)$ which divided by edge total from sentence S_j. α is damping factor (0.85). While $NodeCount$ is the number of nodes in a graph. By assuming sentences as a node, the PageRank algorithm used to rank each sentence that is composed in a graph. The ranking generated by PageRank can be used to ensure that the selected sentences from GA are an important sentence.

D. Text Summarization using Genetic Algorithm

After a graph is constructed and each of its nodes is given some weights. Then the desired summary size should be defined. The summary size is computed through a compression rate which is a manually fixed parameter to indicate the number of selected sentences. The number of selected sentences is computed as follows in Eq. (3).

$$Ns = N * R \quad (2)$$

$\boldsymbol{Ns}$ or Number of Sentence is the number of sentences that will be generated. N is the total number of sentences in 1 document and R is the compression rate that will determine the length of the resulting summary. For example, suppose the document consists of 21 sentences and the compression rate set up to 30% then the number of selected sentences will be equal to 6.

The stages of genetic algorithm process are given in Algorithm 1.

Chromosome Encoding. GA must encode each solution using a canonical way. One of the most used encodes for a chromosome is the binary (0,1).

(1) **Begin** Summary Extraction
(2) Set parameters
(3) Encode chromosome
(4) Generate Initial Population
(5) Compute fitness
(6) **while** (!stop Condition) do
(7) Evaluate initialize population by fitness calculation
(8) Select individual, by a tournament process
(9) Perform crossover with probability p_c
(10) Perform mutation with probability p_m
(11) **end while**
(12) Decode the individual with maximum fitness
(13) return sentences
(14) **End** Summary Extraction

Algorithm. 1. The pseudocode of genetic algorithm

In this genetic solution, the bits of given chromosome are the sentences of the document. 1 is used to denote the selected sentence, otherwise if not selected will be 0 as shown in Fig. 3.

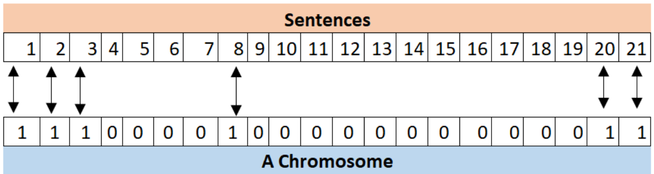


Fig. 3. The encoding of the genetic algorithm.

Initial Population. After chromosome encoding is setup, the population of 10 chromosomes is randomly generated in the beginning. Random function is applied to generate random floating-point array [0,1] as in Fig. 4. The number of selected sentences in each population based on compression rate.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21
P1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
P2	1	1	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
P3	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
P4	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
P5	0	0	1	1	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1
P6	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1
P7	1	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0
P8	1	0	0	0	0	1	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0
P9	0	1	0	0	1	0	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0
P10	0	0	0	1	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0

Fig. 4. Initial population.

Selection. Individuals with the best fitness values will be selected as parents for the next generation. There are several methods for selection process in genetic algorithm, but this study will use the tournament method with a tournament size equal to 4. This method is carried out by selecting individuals with the best value in a population that will compete with other individuals, the results of this competition will produce a winning individual who is selected to enter the next generation.

Fitness Function. Fitness function is needed to evaluate the quality of the chromosomes in a population, if the value of the subset on the chromosome is good there will be a higher probability chosen in the next population. In this study, the fitness function aims for finding the optimal combination of sentences as a solution by PageRank score.

To ensure that sentences chosen by GA are the important sentences from the source document, a constraint is set by the total PageRank score in the summary must be above the percentage of the summary size. It refers to the basic concept that the total PageRank value of all nodes in a graph must be 1, regardless of the number of nodes. If in a graph

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21
Parent 1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0
Parent 2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21
Child 1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0
Child 2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0

Fig. 5. Crossover Operation.

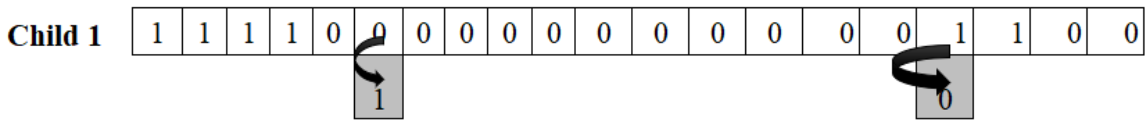


Fig. 6. Mutation Operation.

there are 100 nodes which level of importance is equal, then the value of PageRank of each node must be 0.01. Based on that, if the document wants to summarize by 30% of the original size, then the summary must have a minimum total PageRank 0.3 (0.01 * 30 nodes).

Crossover. In this stage, two individuals will be combined to get new individuals which expected to have better fitness. Crossover exchanges genetic information between two parent chromosomes selected from the selection operation to form a child chromosome, as in Fig. 5.

In our genetic algorithm the crossover operation is not completely random, the produced children must respect the compression rate of summary. The crossover operation is carried out with a crossover probability is equal to 0.8.

Mutation. Serves to replace missing genes from the population because of a selection process that allows the reappearance of genes that do not appear in population initialization.

In our genetic solution, mutation must respect the compression rate of summary as in Fig. 6. For this reason, mutation operation must affect two genes of chromosomes. Besides, these genes must be different ('0' and '1') and the mutation rate is 0.2.

Stopping Criteria. After a generation is created, stopping criteria is used to determine if the genetic algorithm should create another generation or need to stop. The stopping criteria of the algorithm can be either reaching a solution to the problem or reaching the maximum number of iterations [31]. The stopping criteria for implementation purpose is iteration will be stopped when there has been no improvement in the fitness values after *n* successive generations.

E. Evaluation Method

The proposed method in this study is evaluated by two standard evaluation metrics: ROUGE-N and cosine similarity. ROUGE-N (Recall Oriented Understudy of Gisting Evaluation-N) is a standard evaluation metric to test how good the quality of the system summary is. ROUGE calculates the number of n-grams of a word overlap between system summary and reference summary. It is claimed that ROUGE-1 consistently correlated highly with human summary statistically and has recall and precision in significance test with manual evaluation [32]. So, ROUGE-1 is chosen as the measurement of this study and the parameters analyzed are precision, recall, and F-Measure as in Eq. (4), Eq. (5), and Eq. (6).

$$Precision = \frac{Extracted\ summary \cap Reference\ summary}{Extracted\ summary} \quad (4)$$

$$Recall = \frac{Extracted\ summary \cap Reference\ summary}{Reference\ summary} \quad (5)$$

$$F-Measure = \frac{2*Precision*Recall}{Precision+Recall} \quad (6)$$

Another evaluation metric is cosine similarity. As content-based measures, it can determine if two sentences have the same information or not even they are written differently [33]. Cosine similarity compares the difference between reference summary from IndoSum dataset [29] and the summary generated from the proposed method. If the cosine similarity is closer to 1, it means that the two documents have similarities and can be considered as a good summary.

In addition, the results of the summarization will be compared with several other text summarization algorithms that have been implemented previously for the problem of summarizing text. These algorithms include SumBasic, LSA, LexRank, TextRank, and KLSum. The evaluating process uses the same dataset and preprocessing steps for a fair comparison between the proposed method (GA) and those methods. For text summarization, the benchmark algorithms are implemented using the Python library.

IV. Results And Discussions

A. Dataset and Setup

A collection of 60 news articles in Bahasa from IndoSum dataset are used for the experiment. The statistics of the data corpus used are tabulated in Table 1.

TABLE I
Statistic of dataset

Parameter	Size
Number of documents	60
Minimum number of sentences per docs	13
Average number of sentences. per docs	22
Maximum number of sentences. per docs	47

For evaluation purposes, the compression ratio is set up as 10%, 20%, and 30% respectively. The experiment is implemented in Python. There are some parameters need to be initialized. Some of them are specific for GA which are shown in Table II.

TABLE II
Parameter for GA

Parameter	Size
Tournament size	4
Crossover probability	0.8
Mutation rate	0.2
Maximum generation	100

Fig.7 reflects the example for the fitness value over generations for a GA on 1 article. Each generation is tested to calculate the average fitness value. From these experiments, it will be obtained where is the optimal solution for problem solving. Fig.7 showed that the generation that produces the best fitness is around on 40th generation above or as indicated by the arrow in the figure. In this generation, the fitness value of 0.0449 was obtained. From these results, it can be concluded that the most optimal number of generations is in that generation because there is no increase in the fitness value if the iteration is continued.

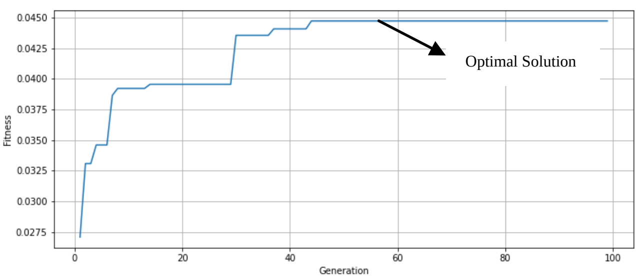


Fig. 7. Fitness evolution over generation using GA.

B. Results and Discussion

The results of the evaluation of the proposed method are presented in the following Table III. As can be seen in the table, the average of recall will be higher as the compression ratio is increased. For example, the average of recall for compression ratio of 30% is equal to 0.640, that means 64% of reference summary are also presented in the generated summary. And the lower recall on compression ratio of 10% with value equal to 0.406. To conclude, recall score is directly proportional to the number of summaries generated.

It means the higher of compression rate set up, the more sentences in the reference summary have been captured by the system summary.

TABLE III
Results of the proposed method

Evaluation Metrics	**ROUGE-1 score**		
	CR=10%	**CR=20%**	**CR=30%**
Avg. Precision	0.489	0.387	0.330
Avg. Recall	0.406	0.539	0.640
Avg. F1 measure	0.426	0.434	0.421

On the other hand, precision as a measure of how much information that the system returned is correct shows the average of precision decreased along with increased compression rate, though the decline is not significant. For example, precision value from compression ratio of 30% is equal to 0.330. The precision tells us that out of all the system summary bigrams, there is a 33% overlap with the reference summary. And the highest precision score is 48,9% contained in the summary results of compression ratio of 10%.

The average precision gets lower as the level of compression rate is increases. It means, the greater the degree of summarization, the smaller it is the proportion of the number of summaries generated by the system and seemed relevant. Precision explains that portion of sentences selected were part of the reference summary. In summary, the overlapping 1-gram is less to be found because the generated summary contains more sentences compared to the reference summary makes precision values is lower compared to the recall value.

Fig. 8a, 8b, and 8c represents the results obtained by comparing precision, recall and F1 measure on every topic based on compression ratio. From the figures it can be seen that documents with Headline's topic performed well compared to other topics achieving highest F-measure.

Graphically, Fig. 9 shows the comparison of minimum, average, and maximum cosine similarity based on compression ratio (10%, 20%, and 30%). Here, the graph showed the average of cosine similarity are 0,562, 0,602, 0,626, respectively. The result shows compression ratio of 30% got the most similar between summary generated and reference summary with value equal to 0,626 or degree of similarity is equal to 62,5%. To conclude, compression rate indicates that the higher compression ratio defined, the similarity with manual summaries higher also.

Table IV shows the generated summary of the sample input document from the IndoSum dataset generated by proposed method along with reference summary on a 30% compression ratio.

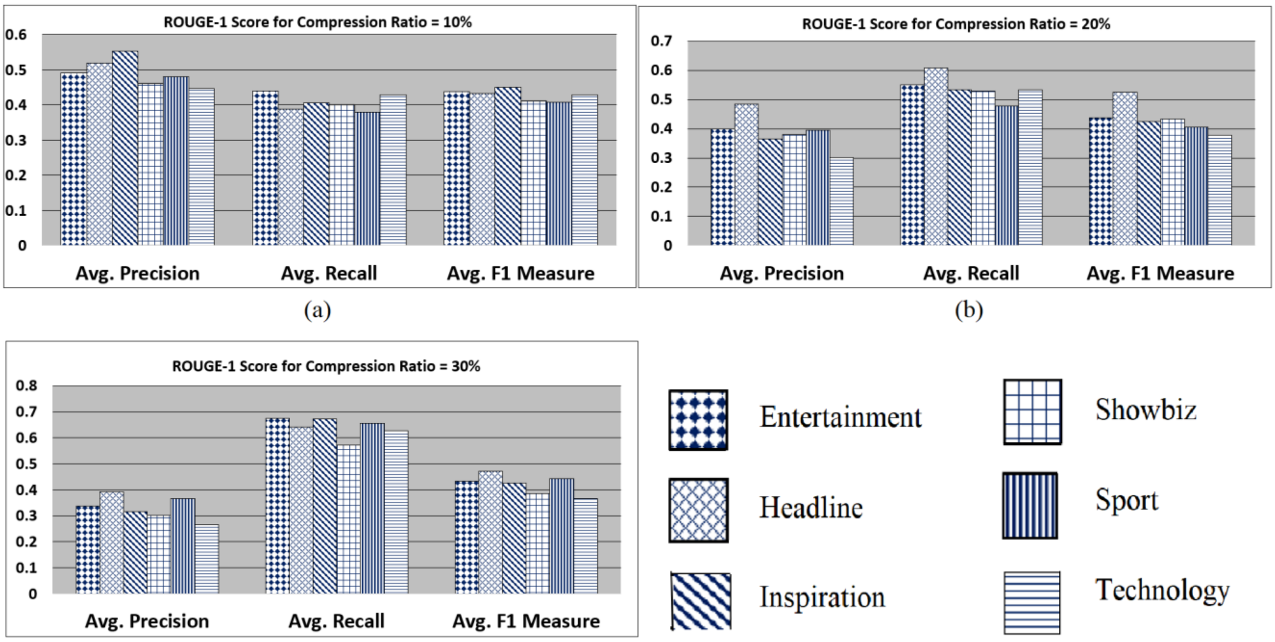
C. Comparison with Other Methods

Table V, Table VI, and Table VII show the comparison between the proposed method and other methods. Those tables contain three results from each compression ratio using five different algorithms SumBasic, LexRank, LSA, TextRank, and KLSum.

Table V presents average ROUGE-1 precision, recall, and F-measure for the proposed method with related methods for compression ratio of 10%. The result show that the proposed method in precision score with value equal to 0.489 and recall scores with value equal to 0.406 and TextRank method outperforms with value equal to 0.480. For f-measure scores, proposed method outperforms with values equal to 0.426.

Table VI presents average ROUGE-1 precision, recall, and F-measure for the proposed method with related methods for compression ratio of 20%. Result show that precision score outperforms all other methods in precision scores with value equal to 0.387. For f-measure scores, proposed method outperforms with values equal to 0.434.

Table VII presents average ROUGE-1 precision, recall, and F-measure for the proposed method with related methods for compression ratio of 30%.



(a) (b) (c)

Fig. 8. ROUGE-1 score for (a) compression ratio = 10%, (b) compression ratio = 20%, and (c) compression ratio = 30% group by topic

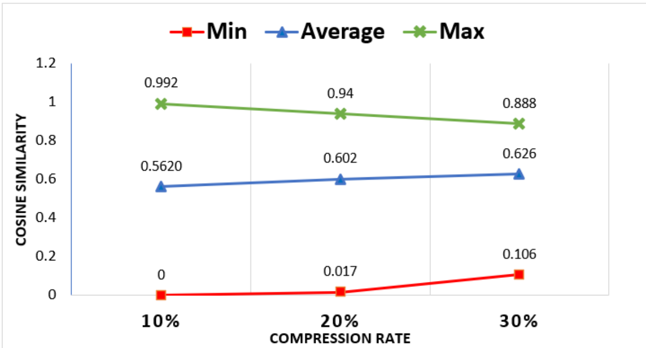


Fig. 9. Cosine similarity based on compression rate.

TABLE IV
A sample summary generated by system (top) and its reference summary (bottom)

Summary Generated by System	Diyakini bermanfaat buat tubuh, tidur sejenak sekitar 20-30 menit di siang hari cukup beralasan untuk dilakukan. Apalagi bagi mereka yang tidur di malam hari tak cukup. Selain membuat tubuh kembali ke kondisi lebih baik, tidur siang juga membuat seseorang lebih kreatif dan produktif. Banyak orang juga mengalami rasa kantuk setelah makan siang atau kurang waspada, sehingga jam ini terasa pas untuk tidur siang. Durasi tidur selama 20-30 menit saja, tak lebih. Hal ini disebut dengan inersia tidur atau sleep inertia.
Reference Summary	Sleep Council Inggris mengungkapkan waktu tepat untuk tidur siang adalah pukul 15 . Alasannya, melihat jam biologis tubuh, kondisi tubuh menurun di jam tersebut . Banyak orang mengalami rasa kantuk setelah makan siang, maka jam ini pas untuk tidur siang . Durasi tidur selama 20 -30 menit saja .Karena jika seseorang tidur melebihi 30 menit, ia akan tertidur lebih dalam dan pening saat terbangun .

Result show that F-measure score has the highest score compared to others compression ratio with values equal to 0.421. We also notice that F-measure values for the proposed method win over related methods. In all cases, increase the F-measure, which indicates that the generated summary covers most of the words in the reference summary. As a result, the summaries tend to be more similar to the reference summary.

TABLE V
Comparison of ROUGE-1 results of the proposed method with those of other methods by compression ratio of 10%

Method	Precision	Recall	F1 Measure
SumBasic [7]	0.501	0.234	0.307
LexRank [6]	0.445	0.337	0.368
LSA [8]	0.356	0.370	0.349
TextRank [5]	0.383	0.480	0.416
KLSum [34]	0.406	0.310	0.335
Proposed GA	**0.489**	**0.406**	**0.426**

TABLE VI
Comparison of ROUGE-1 results of the proposed method with those of other methods by compression ratio of 20%

Method	Precision	Recall	F1 Measure
SumBasic [7]	0.364	0.430	0.380
LexRank [6]	0.370	0.489	0.407
LSA [8]	0.303	0.512	0.367
TextRank [5]	0.317	0.603	0.407
KLSum [34]	0.377	0.354	0.349
Proposed GA	**0.387**	**0.539**	**0.434**

TABLE VII
Comparison of ROUGE-1 results of the proposed method with those of other methods by compression ratio of 30%

Method	Precision	Recall	F1 Measure
SumBasic [7]	0.350	0.469	0.388
LexRank [6]	0.331	0.612	0.416
LSA [8]	0.283	0.637	0.381
TextRank [5]	0.281	0.698	0.392
KLSum [34]	0.311	0.423	0.344
Proposed GA	**0.330**	**0.640**	**0.421**

TABLE VIII
Comparison of cosine similarity results of the proposed method with those of other methods

Method	Cosine Similarity (Average)		
	CR 10%	CR 20%	CR 30%
SumBasic [7]	0.449	0.492	0.515
LexRank [6]	0.488	0.552	0.596
LSA [8]	0.429	0.49	0.55
TextRank [5]	0.55	0.594	0.621
KLSum [34]	0.455	0.478	0.488
Proposed GA	**0.562**	**0.602**	**0.626**

The other comparison metric is cosine similarity score. Table VIII shows a comparison of cosine similarity score between the results obtained in this study and those obtained by other methods. The evaluation has been done by comparing the summaries generated with manual summaries to see how good and similar they are. As can be seen in the table, the proposed GA outperformed the other methods. The results show percentage similarity 56% for 10% CR, 60% for 20% CR, and 62% for 30% CR, respectively. In short, more than 50% of information in summary generated by the proposed method are exists also in manual summary.

Compared to TextRank method, the proposed GA has a similar result because our proposed method used graph concept likes TextRank method however genetic algorithm still outperform. The main strength of genetic algorithm is that it can find a larger number of solutions than other algorithms. By exploring more solutions, the chances of finding the best solution according to the objective function are greater than with other algorithms that use only the heuristic approach.

V. Conclusion And Future Work

This paper proposed the extractive text summarization using genetic algorithm to improve the quality of summary results based on the statistical method. The Genetic Algorithm is applied to generate summaries by selecting an important sentence from the graph based on the desired summary size. The PageRank algorithm is used to evaluate a statement to derive a representative sentences in the document.

The results are good enough to call this method efficient and even better than the existing method and show how the result of text summarization can be improved by integrating evolutionary algorithm techniques like genetic algorithm. Compared to other algorithms, the proposed algorithm gives promising results. For compression ratio of 10%, it produces results equal to 49%, 40%, and 43% for ROUGE-1 Precision, Recall, and F-measure, respectively. For compression ratio of 20%, it also produces results equal to 39%, 54%, and 43% for ROUGE-1 Precision, Recall, and F-measure, respectively. In addition, for compression ratio of 30%, it produces results equal to 33%, 64%, and 42% for ROUGE-1 Precision, Recall, and F-measure, respectively. The other comparison to see how similar the summary generated by the system with reference summary also showed outperform result. It produces results equal to 56%, 60%, and 62% for cosine similarity scores from compression ratios of 10%, 20%, and 30% respectively.

For future work, it is suggested to develop a score measuring formula for a better text feature to improve the readability and comprehension for Indonesian text summarization. The proposed method in this work can also be used as the basis for developing algorithms for multiple documents or an abstractive summarization approach for the Indonesian language.

References

[1] A. Elrefaiy, A. R. Abas, and I. Elhenawy, "Review of recent techniques for extractive text summarization," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 23, pp. 7739–7759, 2018.

[2] P. B. Baxendale, "Machine-Made Index for Technical Literature—An Experiment," *IBM J. Res. Dev.*, vol. 2, no. 4, pp. 354–361, Oct. 1958.

[3] H. P. Luhn, "The Automatic Creation of Literature Abstracts," *IBM J. Res. Dev.*, vol. 2, no. 2, pp. 159–165, Apr. 1958.

[4] H. P. Edmundson, "New Methods in Automatic Extracting," *J. ACM*, vol. 16, no. 2, pp. 264–285, Apr. 1969.

[5] P. Tarau, "Structure of the Felidae hemoglobins and response to 2,3-diphosphoglycerate," *Comp. Biochem. Physiol. -- Part B Biochem.*, vol. 45, no. 4, 1973.

[6] D. R. Radev and G. Erkan, "LexRank : Graph-based Centrality as Salience in Text Summarization," *J. Artif. Intell. Res.*, vol. 22, no. 1, pp. 457–479, 2004.

[7] L. Vanderwende, H. Suzuki, C. Brockett, and A. Nenkova, "Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion," *Inf. Process. Manag.*, vol. 43, no. 6, pp. 1606–1618, 2007.

[8] M. G. Ozsoy, F. N. Alpaslan, and I. Cicekli, "Text summarization using Latent Semantic Analysis," *J. Inf. Sci.*, vol. 37, no. 4, pp. 405–417, Aug. 2011.

[9] H. Christian, M. P. Agus, and D. Suhartono, "Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF)," *ComTech Comput. Math. Eng. Appl.*, vol. 7, no. 4, p. 285, Dec. 2016.

[10] K. Nandhini and S. R. Balasundaram, "Use of Genetic Algorithm for Cohesive Summary Extraction to Assist Reading Difficulties," *Appl. Comput. Intell. Soft Comput.*, vol. 2013, pp. 1–11, 2013.

[11] G. Algaphari, F. M. Ba-Alwi, and A. Moharram, "Text Summarization using Centrality Concept," *Int. J. Comput. Appl.*, vol. 79, no. 1, pp. 5–12, 2013.

[12] Lucky and A. S. Girsang, "Multi-Objective Ant Colony Optimization for Automatic Social Media Comments Summarization," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 3, pp. 400–408, 2019.

[13] N. Chatterjee, A. Mittal, and S. Goyal, "Single document extractive text summarization using Genetic Algorithms," in *2012 Third International Conference on Emerging Applications of Information Technology*, 2012, pp. 19–23.

[14] B. Prasetyo, T. Uliniansyah, and O. Riandi, "SIDoBI: Indonesian Language Document Summarization System," *Int. Conf. Rural Inf. Commun. Technol. 2009*, no. 8, pp. 378–382, 2009.

[15] A. Aristoteles, Y. Herdiyeni, and A. Ridha, "Text Feature Weighting For Summarization Of Document Bahasa Indonesia Using Genetic Algorithm," *Int. J. Comput. Sci. Issues*, vol. 9, no. 3, pp. 1–6, 2012.

[16] P. P. Tardan, A. Erwin, K. I. Eng, and W. Muliady, "Automatic text summarization based on semantic analysis approach for documents in Indonesian language," *Proc. - 2013 Int. Conf. Inf. Technol. Electr. Eng. "Intelligent Green Technol. Sustain. Dev. ICITEE 2013*, pp. 47–52, 2013.

[17] K. Nandhini and S. R. Balasundaram, "Use of Genetic Algorithm for Cohesive Summary Extraction to Assist Reading Difficulties," *Appl. Comput. Intell. Soft Comput.*, vol. 2013, pp. 1–11, 2013.

[18] J. H. Holland, "Even their creators do not fully understand Genetic Algorithms n," vol. 267, no. 1, pp. 66–73, 1992.

[19] R. M. Alguliyev, R. M. Aliguliyev, N. R. Isazade, A. Abdi, and N. Idris, "COSUM: Text summarization based on clustering and optimization," *Expert Syst.*, vol. 36, no. 1, pp. 1–17, 2019.

[20] S. A. Babar and P. D. Patil, "Improving Performance of Text Summarization," *Procedia Comput. Sci.*, vol. 46, pp. 354–363, 2015.

[21] M. Mohd, R. Jan, and M. Shah, "Text document summarization using word embedding," *Expert Syst. Appl.*, vol. 143, p. 112958, 2020.

[22] S. Sehgal, B. Kumar, Maheshwar, L. Rampal, and A. Chaliya, "A modification to graph based approach for extraction based automatic text summarization," *Adv. Intell. Syst. Comput.*, vol. 564, pp. 373–378, 2018.

[23] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, “EdgeSumm: Graph-based framework for automatic text summarization,” *Inf. Process. Manag.*, vol. 57, no. 6, p. 102264, 2020.

[24] Ahmed M. El-Refaiy, Ahmed R. Abas, and Ibrahim M. El-Henawy, "Determining Extractive Summary for a Single Document Based on Collaborative Filtering Frequency Prediction and Mean Shift Clustering," IAENG International Journal of Computer Science, vol. 46, no.3, pp494-505, 2019.

[25] R. A. García-Hernández and Y. Ledeneva, “Single Extractive Text Summarization Based on a Genetic Algorithm,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7914 LNCS, 2013, pp. 374–383.

[26] Y. K. Meena and D. Gopalani, “Evolutionary Algorithms for Extractive Automatic Text Summarization,” *Procedia Comput. Sci.*, vol. 48, no. C, pp. 244–249, 2015.

[27] Silvia, P. Rukmana, V. R. Aprilia, D. Suhartono, R. Wongso, and Meiliana, “Summarizing Text for Indonesian Language by Using Latent Dirichlet Allocation and Genetic Algorithm,” *Int. Conf. Electr. Eng. Comput. Sci. Informatics*, vol. 1, no. August, pp. 148–153, 2014.

[28] D. Gunawan, A. Pasaribu, R. F. Rahmat, and R. Budiarto, “Automatic Text Summarization for Indonesian Language Using TextTeaser,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 190, no. 1, p. 012048, Apr. 2017.

[29] K. Kurniawan and S. Louvan, “Indosum: A New Benchmark Dataset for Indonesian Text Summarization,” in *2018 International Conference on Asian Language Processing (IALP)*, 2018, pp. 215–220.

[30] GitHub Inc., “GitHub - masdevid/ID-Stopwords: Stopwords collection of Bahasa Indonesia collected from many sources.” [Online]. Available: https://github.com/masdevid/ID-Stopwords.

[31] Islam T. Elgendy, Moheb R. Girgis, and Adel A. Sewisy, "A GA-Based Approach to Automatic Test Data Generation for ASP.NET Web Applications," IAENG International Journal of Computer Science, vol. 47, no.3, pp557-564, 2020.

[32] L. Suanmali, N. Salim, and M. S. Binwahlan, “Genetic algorithm based sentence extraction for text summarization,” *Int. J. Innov. Comput.*, vol. 1, no. 1, p. 22, 2011.

[33] J. Steinberger and K. Ježek, “Evaluation Measures for Text Summarization,” *Comput. Informatics*, vol. 28, no. 2, pp. 251–275, 2009.

[34] L. Vanderwende, “Exploring Content Models for Multi-Document Summarization,” no. June, pp. 362–370, 2009.