IMPLEMENTATION OF DIFFERENTIAL EVOLUTION METHOD ON INDONESIAN TEXT SUMMARIZATION

Firdanto Prasydani[1], Nelly Indriani W., S.Si., M.T.[2]

[1,2]Teknik Informatika – Universitas Komputer Indonesia
Jalan Dipatiukur 112-116 Bandung
E-mail : [1]prasydani@yahoo.com, [2]nelly.indriani@email.unikom.ac.id

ABSTRACT

In several studies of text summarization, some of the main issues that arise in the field of automated text summarization are the relevance and subject matter of the summary.
The results of a good summary should contain important information without neglecting the relevance and subject matter of the summary. Differential Evolution method is then proposed to optimize the features extracted from a document based on informative and cohesive scores in order to obtain a good summary. Differential Evolution Method is an optimization method that recently developed and proved to have good performance.
Based on these results, Differential Evolution is used in this study to generate a summary of Indonesian texts that have the best relevance structure. In this study extracted 5 pieces of features on the document which are the position of sentence, word frequency, keywords, similarity with the title and centrality. Testing performed is a test of accuracy which is done by comparing the results of system summaries with the results of manual summaries obtained from Indonesian experts. System testing yields an average accuracy of 47.02% with the best result 83.33%. While based on ROUGE-1 test obtained average f-measure 57.22% with the best result 91,46, ROUGE-2 test obtained average f-measure 49.66% with the best result 92.31%.
Keywords : Differential Evolution, Automatic Text Summarization, NLP, Algorithm, Optimization.

1. INTRODUCTION

Automatic text summarization is making a shorter form from a text by using aplication that run and operated on a computer [1]. While according to N. Bhatia and A. Jaiswal, summary is a text that generated from a text or many texts, which contain information from the original text and has less half the length of the original text [2]. Automatic text summarization research generally do using 2 ways in the classification technique which is extractive and abstractive [2]. Extractive summarization are not focused on understanding the text. But focused on extracting the most important part based on statistical features and linguistics like cue words, position and a word frecuency [3]. While abstractive summarization put paraphrase part from the source document. In general, abstarctive summarization can summarize a text better than extractive summarization. But the system that can do it are more difficult to build because it needs the use of natural language generation technologies, which is still a growing field.

Some major issues that arise on automatic text summarization field are relevance and the main subject from the summary which can still be developed again [4][5]. A good summary should contain the main information without neglecting the relevance and the subject of the summary [4][5].

Based on research from K. Nandhini [6] has been done comparison between Differential Evolution (DE) and Genetic Algorithm (GA) in optimizing extracted features from a document based on informative and cohesive score as the fitness function. The result of the comparison lead to a conclusion that DE produce better summary than GA. Based on that result, Differential Evolution is used on this research to determine it performance on producing Indonesian text summary which has good relevance structure [6]. Other result gained from implementing GA on text summarization are research conducted by Zulkifli which results 53% in summary accuracy [7], research conducted by Aristototeles produces an accuracy of 40%[8], research conducted by Jaradat produces 54,8% of f-measure value[9], While result gained by K. Nandhini on implementing GA produce average accuracy 60% on testing[6][10]. Other testing result on DE implementations are research conducted by R. Alguliev who produce 43% of ROUGE-1 testing value[4], research conducted by Shweta with their Discrete DE produce 36,33% average f-measure value [11], While Fajarwati and their VRP-DP problem manage to lower 18% from the total of traveling cost with DE [12], Nurdiansyah with their flowshop multi objective scheduling with many machines using DE manage to get better average relative presentation compared to other testing which is always under 1%[13], Mohanty manage to build a LFC system for multi source with DE as the control parameter tuning which produce a consistent system with no disturbance from loading codition changes and better control performance [14], while research conducted by K. Nandhini in using DE

for text summarization yields 71% average accuracy score[6].

This research use Differential Evoltuion (DE) which is a population based method who optimized a problem iteratively develop candidate solutions with quality weight measurement. Basically DE is an algorithm similar to GA. But the differences is GA depends on its crossover while DE depends on its mutation [4]. Beside that, DE is a efficient and easy to use algorithm compared to other evolutionary algorithm [15].

2. METHOD

This research use Differential Ecolution for Indonesian text summarization, where Differential Evolution is a optimization method [16]. The start from this research are input data analysis where inputted data must be on text form. After that, inputted data will be processed in Preprocessing phase, then features from the data will be extracted, then on the final phase those features will be processed using Differential Evolution to produce a text summary.

System architecture on this automatic text summarization has 4 main phase which is preprocessing, feature extraction, counting word weight and implementing Differential Evolution. For the explanation on every phase in summarization process refer to image 1 below:

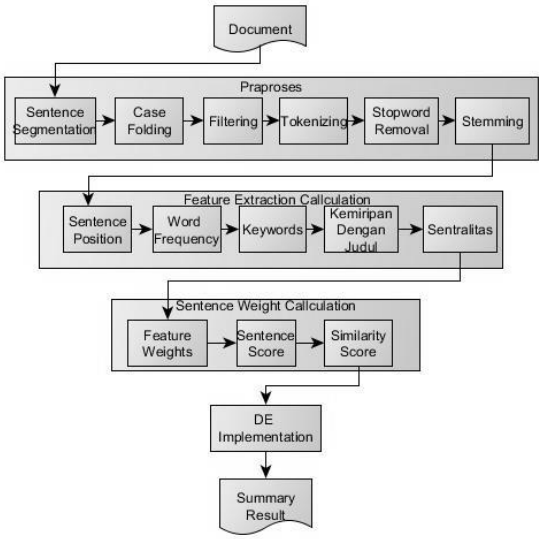


Image 1. System Illustration

2.1 Preprocessing

Preprocessing is a phase where selected document, cleaned from noise. The Preprocessing done on this phase are: sentence segmentation, case folding, filtering, word tokenizing, stopword removal and stemming. Below is the Preprocessing phase on image 2:

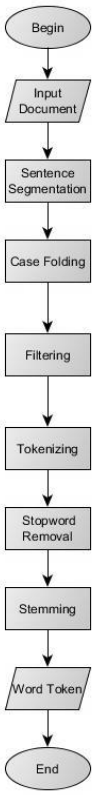


Image 2. Preprocessing

2.2 Features Extraction

This section will explain features extraction process from preprocessed document. Extracted features are sentence position, word/term frequency, keyword, title similarity and centrality.

2.2.1 Sentence Position

This feature is under assumption that the firs sentence and the last sentence on a paragraph is an important part [6]. This feature is gained based on the following formula (1):

$$Sentence\ Position = Max(\frac{1}{i}, \frac{1}{n-i}) \quad (1)$$

Where:
n = Number of sentences on a paragraph
i = Sentence position

2.2.2 Term Frequency

The meaningful for a term can be determinded from the frequency of its appearences on a paragraph. For that purpose, TF-isf are used to extract term frequency. Tf-isf is a stastitical method which is used to evaluate the meaningful of a term in a document paragraph. This method can extract the weights from every term after stopwords removed from the text. Tf-isf can be calculated using the following formula (2) and (3) [6]:

$$isf = log(N/ni) \quad (2)$$

Where :
N = Number of sentence in document
n = Number of sentence that contain ith term

$$Term\ Frequency = Tf * isf \quad (3)$$

2.2.3 Keywords

At first, keyword is defined as a non-gesture term with measured highest weigth, and keyword term used become appearance frequency in a document [17]. On this research, extracted keywords feature gained from ≤ 10 most often appear term. After keywords obtained, every same token on the sentence with the keywords will be given 0,1 and 0 otherwise [18].

2.2.4 Title Similarity

Title containt a set of term which gives an important lead about the concept of a pharagraph [19]. The similarity of every term on the title can be measured with the follwoing formula (4) below:

$$Similarity = \frac{|Term\ in\ title \cap Term\ in\ sentence(i)|}{|Term\ in\ title \cup Term\ in\ sentence(i)|} \quad (4)$$

Following is a example of similarity value on a document with every sentence on table 1 and 2 below:

Table 1. Title Token

No	Token
1	Implementasi
2	Metode
3	histograms
4	Of
5	Oriented
6	gradients
7	optimal
8	algoritma
9	frei-chen
10	deteksi
11	citra
12	manusia

Tabel 2. Title Similarity Example

Kalimat	Bobot
S1	$\frac{8}{24} = 0{,}3333$
S2	$\frac{4}{21} = 0{,}1905$
S3	$\frac{2}{19} = 0{,}1053$
S4	$\frac{4}{26} = 0{,}1538$
S5	$\frac{1}{30} = 0{,}0333$
S6	$\frac{2}{21} = 0{,}0952$
S7	$\frac{2}{24} = 0{,}0833$
S8	$\frac{2}{25} = 0{,}08$
S9	$\frac{3}{22} = 0{,}1364$
S10	$\frac{4}{25} = 0{,}16$

2.2.5 Centrality

Centrality of a sentence states its similarity to other sentence [20]. Also mentioned on journal [21] that centrality is a overlapping of a vocabulary between 2 sentences. If a sentence have higher centrality, then that sentences is a best candidate to be included in summary result. And so, on this research to measure centrality value from a sentence, done matching between every sentence to other sentences until found the highest centrality value from a sentence. Below is the formula used to measure centrality:

$$Centrality = \frac{|Term\ in\ sentence(i) \cap Term\ in\ sentence(j)|}{|Term\ in\ sentence(i) \cup Term\ in\ sentence(j)|} \quad (5)$$

Below is the example of centrality calculation on table 3:

Table 3. Centrality Calculation Example

Kalimat	Bobot
S2	$\frac{\lvert Kata\ di\ S1 \cap kata\ di\ S2\rvert}{\lvert Kata\ di\ S1 \cup kata\ di\ S2\rvert} = \frac{4}{21} = 0{,}1905$
S3	$\frac{\lvert Kata\ di\ S1 \cap kata\ di\ S3\rvert}{\lvert Kata\ di\ S1 \cup kata\ di\ S3\rvert} = \frac{3}{19} = 0{,}1579$
S4	$\frac{\lvert Kata\ di\ S1 \cap kata\ di\ S4\rvert}{\lvert Kata\ di\ S1 \cup kata\ di\ S4\rvert} = \frac{3}{26} = 0{,}1154$
S5	$\frac{\lvert Kata\ di\ S1 \cap kata\ di\ S5\rvert}{\lvert Kata\ di\ S1 \cup kata\ di\ S5\rvert} = \frac{1}{30} = 0{,}0333$
S6	$\frac{\lvert Kata\ di\ S1 \cap kata\ di\ S6\rvert}{\lvert Kata\ di\ S1 \cup kata\ di\ S6\rvert} = \frac{2}{21} = 0{,}0952$

S7	$\frac{\lvert Kata\ di\ S1 \cap kata\ di\ S7\rvert}{\lvert Kata\ di\ S1 \cup kata\ di\ S7\rvert} = \frac{3}{24} = 0{,}125$
S8	$\frac{\lvert Kata\ di\ S1 \cap kata\ di\ S8\rvert}{\lvert Kata\ di\ S1 \cup kata\ di\ S8\rvert} = \frac{3}{25} = 0{,}12$
S9	$\frac{\lvert Kata\ di\ S1 \cap kata\ di\ S9\rvert}{\lvert Kata\ di\ S1 \cup kata\ di\ S9\rvert} = \frac{10}{22} = 0{,}4545$
S10	$\frac{\lvert Kata\ di\ S1 \cap kata\ di\ S10\rvert}{\lvert Kata\ di\ S1 \cup kata\ di\ S10\rvert} = \frac{3}{25} = 0{,}12$

2.3 Sentence Weight Calculation

Sentence weight calculation is done as it is used as input data for objective function in Differential Evolution. To calculate it it must pass 3 phase wihich is feature weight, sentence score and similarity score.

2.3.1 Feature Weight

To calculate Feature weight, used a comparison matrix to compare every feature value with other feature value. Below is the illustration on the matrix [22]:

Tabel 4. Feature Weight Matrix

Feature	F_1	F_2	...	F_n
F_1	1	F_2/F_1	...	Fn/F_1
F_2	F_1/F_2	1	...	Fn/F_2
.	.	.	...	..
F_n	F_1/Fn	F_2/Fn		1
$\sum$	W_1	W_2	...	W_n

On this research, extracted features from a document are:
F1 = Sentence Position
F2 = Term Frequency
F3 = Keyword
F4 = Title Similarity
F5 = Centrality

Those features then averaged and inputted to the comparison matrix using the following formula (6) below:

$$F_i = \frac{\sum_{j=1}^{n} Fij}{n} \quad (6)$$

Dimana:
Fi = ith Feature
j = jth Sentence
n = number of sentence

After that, the value from feature weight then normalized by difiding every Fi/Fj value with every W beneath it. The value then summed down vertically to obtain the feature weight value.

2.3.2 Sentence Score

Every sentences score in the document are used as one of the fitness function that will be used in Differential Evolution. Every sentences score can be measured using formula (7) below[6][10]:

$$Sentence\ Score = \sum_{i=1}^{n} Wi * Fi \quad (7)$$

Where :
Wi = ith feature weight
Fi = ith Feature
n = Number of feature

2.3.2 Similarity Score

Similarity measurement is the most important thing on text related research like text summarization, document clustering and information retrieval [6]. Cosine similarity is one of the best document similarity measurement because its sensitivity on vector document pattern. Cosine measure calculate cosine from the angle between two vector features. Cosine similarity can be calculated using following formula (8) below:

$$Sim(Si, Sl) = \frac{\sum_{j=1}^{n} W_{ij} * W_{lj}}{\sqrt{\sum_{j=1}^{n} (W_{ij})^2} * \sqrt{\sum_{j=1}^{n} (W_{lj})^2}} \quad (8)$$

Dimana :
S = Sentence
W = jth Feature on sentence i and l
i,l = sentence index
j = feature index
n = number of feature

Through Cosine similarity, obtained similarity matrix between sentence which is later going to be used to calculate fitness function in Differential Evolution.

2.4 Differential Evolution

Differential Evolution (DE) is a recently developed algorithm, simple and efficient, heuristic population based and has a better performance on convergence and its operation [6]. Below is the main flow of DE on Image 3:

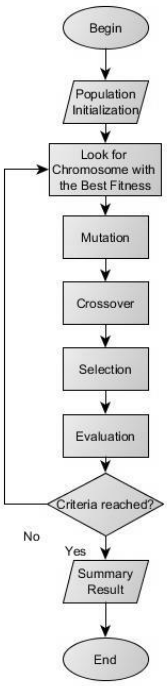


Image 3. DE Flow

2.4.1 Individual Representation

DE use chromosome term to represent it solution candidate. In DE, these chromosomes are composed from a few array of real or decimal numbers. On this research, decimal number is used which is sentence position as allele on the chromosome. To represent 1 solution candidate, a chromosome needs a few value which are later going to be decoded into a valid and measureable fitness solution. Below is the chromosome design used on this research:

a. Number of gene/allele on chromosome are match with the compression rate and number of sentence on the source document. For example the compression rate is 50% and the number of sentences is 13 then the gene on the chromosome is 7. Every genes is a integer number with value higher or equals with 0.
b. Values on this genes will be decoded with random values between 1 to the number of sentence and ordered based on the sentence order. After decode, the gene value will determine which sentence that will be the part of summary generation for a chromosome. Because the text document used on this research is a scientific document, the first and the last loci of the chromosome are positioned for the first and the last sentence on the document.

Below is the example of a chromosome that will be generated based on a document with 10 number of sentence with compression rate 50% on image 4:

Allele	1	4	8	9	10
Urutan Allele	1	2	3	4	5

Gambar 4. Contoh Kromosom DE

2.4.2 Population Initialization

When system start, system will generate other initial chromosomes randomly based on the first chromosome generated on individual decode. For example, a chromosome is generated like on image 4, then other individual will be generated with the same ways. The number of candidate solution is an observed value. Because in DE, the smaller the population, the smalled its diversity and so otherwise [4].

2.4.3 Fitness Function

Fitness function must be able to measure the quality of a chromosome in the population accurately. Fitness function used to obtain the combination from a sentence who have highest informative and cohesive score. Below is the formula used to obtain the informative and cohesive score of a chromosome on formula (9) and (10):

$$Fitness1 = \frac{1}{1+\sum_{i=1}^{N}\sum_{j=1}^{n} Wj*Fij} \quad (9)$$

Where:
N = number of sentence
n = number of feature
W = Feature Weight
F = Feature
i = Sentence index
j = feature index

$$Fitness2 = \frac{1}{\sum_{i=1}^{n}\sum_{j=i+1}^{n-1} Sim(Si,Sj)} \quad (10)$$

Where:
i,j = Sentence index
n = number of sentence
Sim = similarity value

These both values are used to obtain the fitness value of a chromosome with formula (11) below:

$$Fitness = Fitness1 * Fitness2 \quad (11)$$

2.4.4 Mutation

DE is a mutation based algorithm which is done after populations are initialized to make new population. Mutation process on DE is done by finding difference between a few individual selected randomly, Differs from most of evolutionary algorithm [4]. DE mutation process on this research is done by using modified mutation process on image 4 below[6]:

```
Procedure Mutasi_DE
1:    BangkitkanIndeks(indeksA, indeksB)
2:    for i ← 2, n – 1 do
3:      M[i] = Absolute(X_indeksA[i] – X_indeksB[i])
4:    Endfor
5:    for i ← 2, n – 1 do
6:      if (M[i] <> 0) then
7:        if ((X_j[i+1] - X_j[i-1]) > 2) then
8:          BangkitkanBilanganAcak(R)
9:          if((R > X_j[i-1]) AND (R < X_j[i+1])
          AND (R <> X_j[i])) then
10:           trial[i] = R;
11:         endif
12:       endif
13:     endif
14:   endfor
end procedure
```

Gambar 4. Prosedur Mutasi

On the start of the procedure, 2 chromosome index which is different from each other will be randomly generated. M[i] will then be counted to determine mutation process. R is a integer number which is randomly generated between allele Xj[i-1] and Xj[i+1]. Trial array is an array which has a same length and value with chromosome j but used to contain R value. After a chromosome pass mutation process, a trial arrays is then obtained which is a combination between the chromosome's allele values and the new allele values.

2.4.5 Crossover

To improve diversity of a mutated vector, a crossover is then done[23]. On this research, crossover used is a modified crossover which is done by Nandhini [6]. Below is the procedure which passed by the trial array to do the Crossover [6]:

```
Procedure Crossover_DE
1:  k = 0
2:  for i ← 2, n – 1 do
3:    if((X_j[i] >= trial[i]) AND (X_j[i] <
      trial[i+1])) then
4:      tC[k] = i
5:      k++
6:    endif
7:  endfor
8:  for i ← 1, getLength(tC) do
9:    BangkitkanBilanganAcak(randCR)
10:   if(randCR <= cR)then
11:     trial[tC[i]] = trial[tC[i]]
12:   else
13:     trial[tC[i]] = X_j[tC[i]]
14:   endif
15: endfor
end procedure
```

Gambar 5. Prosedur Crossover

Crossover on DE is done between the original chromosome and its mutated chromosome [4][6][16]. Crossover can be done by finding allele who have potention to crossover. That allel is finded by looking at its Sentence number, If its sentence number is same or higher than the previous allel and smaller than the next allele on the trial array. This are done to keep the sentence ordering and redudancy on trial array. randCR is a float number which is randomly generated to determine the crossover probabilities. cR is a input variable which is inputed by user to determine the crossover probabilites.

2.4.6 Selection

DE use the principal of "Survival of the fittest" on its selection process. After mutation and crossover process, every new generated chromosome will be compared with its old chromosome [23]. Between the two values, chromosome which has a higher value will be the new chromosome to go to the next generation.

2.4.7 Stopping Criteria

Stopping criteria is a solution which is used to limit DE so that there is no infinite loop. Stopping criteria generally used to stop DE is when there is no significant chanes on the last few generations (local converge) or when the maximum iteration of population is reached [6]. On this research is used the maximum iteration on the stopping criteria. But on the testing, founded 3 document which has convergence point near the maximum iteration. Because of that, the three document's testing is done by adding another stopping criteria which is if there is no significant changes for (0,2 * maximum generations), the loop will be stopped.

2.4.7 Output DE

After stopping, there is going to be a population of chromosomes which containt combination of alleles generated on the initial DE phase. Chromosome with the best fitness it then searched on this population. The allele value of this chromosome is then used as the summary solution. Below is the example of final phase of DE towards a document that has 10 sentences and 50% compression rate on table 5:

Table 5. DE Result

Gene	Sentence
1	Deteksi manusia pada suatu citra digital dapat dilakukan dengan menggunakan metode dari Dalal dan Triggs pada tahun 2005 yaitu Histograms of Oriented Gradients (HOG)
6	Pada citra yang banyak terdapat noise, hasil perhitungan gradien

	masih kurang maksimal sehingga dapat mempengaruhi akurasi pendeteksian
8	Mereka menggabungkan teknik segmentasi sebelum tahap penghitungan gradien pada metode HOG dan menghasilkan perkembangan akurasi deteksi sebesar 4,47%
9	Berdasarkan penelitian oleh Frei dan Chen [4], dihasilkan sebuah algoritma deteksi tepi baru yang disebut algoritma Frei-Chen
10	Dengan demikian akurasi pendeteksian manusia menggunakan metode Histograms of Oriented Gradients pun diharapkan akan meningkat.

3. Study

Methode testing is done to measure the performance and implementation of Differential Evolution method in generating summary sentences. On this testing is used 50 Indonesian prolusion text document on thesis research. The summarys result generated by system are then compared with the manual summary produced by Indonesian experts, to keep in mind that the study case used on this research is Indonesian text. After that, ROUGE-1 and ROUGE-2 testing is also tested on the system summary and reference summary. But before all of that, a parameter testing is done to look for the best paramete. On this research there are a few parameter used which is crossover rate, population and generation. Parameters who produce best fitness with minimum iteration will be used to process other documents. After the best parameters obtained, accuracy testing is done by comparing the system result and the manually produced summary. 50 text documents are used to test the system on this research.

After the testing on 1 document which has the average document length, obtained crossover rate and population parameters 0.6 and 60. With these values, other text documents are also tested to get the best fitness values and summary results by using same generation parameter 1500. After the testing, obtained longest convergence point on every document on finding the best fitness value. Below is the longest convergence point based on the testing on 47 text document with crossover rate 0.6 and 60 population on image 6:

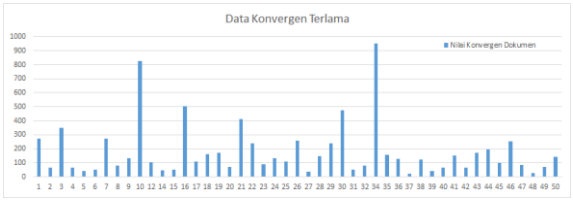


Image 6. Document Convergence point chart

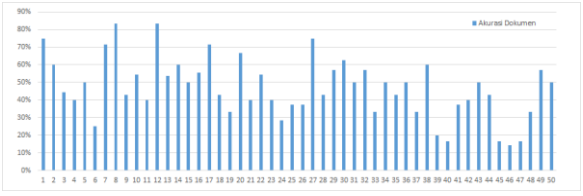
Based on the results, document 11, 13 and 33 are not included on the testing. This is done because those 3 have a convergence point near 1500 generation. And so, for those 3 documents is decided to look for the best fitness values using stopping criteria proposed before.

Best fitness are obtained and the chormosome values will be compared with the reference summaries produced by Indonesian experts. The comparison is done to measure the proximity between analysis and the real result. On this accuracy testing, the quality of a summary is determined by accuracy which is obtained with formula(12) below [24]:

$$Accuracy = \frac{\sum Relevan\ System\ Sentence\ With\ Manuel\ Sentence}{\sum Number\ of\ System\ Sentence} \quad (12)$$

System testing is done with summary sentence compressed with 50% compression rate. Those sentences then counted the similarity with the reference summaries which produced by Indonesian experts which also asked to choose half sentences of the document. Below is the accuracy testing result on Image 7:

Image 7. Documents Accuracy



ROUGE-1 and ROUGE-2 testing is then done on those 50 documents. Below is the average recall, pre*cision* and *f-measure* obtained from ROUGE-1 and ROUGE-2 testing towards the 50 documents on table 6:

Table 6. ROUGE testing results

N	Recall	Precision	F-Measure
ROUGE-1	51,54%	66,61%	57,22%
ROUGE-2	44,74%	57,99%	49,66%

4. CONCLUSION

4.1 Conclusion

Based on the research result which have been done, it could be concluded that Differential Evolution yields average accuracy values 47,02% with best testing result 83,33%. ROUGE-1 has average recall 51,54%, precision 66,61% and f-measure 57,22% with the best testing result recall 92,59%, precision 90,36%, and f-measure 91,46%. While ROUGE-2 yields average recall 44,47%, precision 57,99% and f-measure 49,66% with best testing results recall 93,51%, precision 91,14% and f-measure 92,31%. These average testing results are a low results because its still far from 80%. Crossover rate, generation and population becomes the determinant in how big the sentence combination who have the highest fitness can be found. Crossover rate determine on how often a chromosome do crossovers with its trial array. Population determine how fast or slow a chromosome with highest fitness can be found. While generation determinds how fast or slow DE process stop.

4.2 Suggestion

Based on a few problems obtained on the work of this research, thus the suggestion do develop this research are::

1. Using different mutation and crossover. Because DE is an algorithm which its main process depends on its mutations and crossovers. On its development, its been found different mutation and crossover process other than the one that proposed by Storn and Price when developing the original DE.
2. Adding other stopping criteria for DE. Because, like the testing result on the document by adding the stopping criteria, could be achieved maximum fitness value on the average 21,03% of the maximum generation.
3. Using a better dataset especially on the use of Indonesian words and sentences. Because based ont the testing done on 50 dataset document on this research, obtained various kinds of convergence point which is the difference between each of them are vary.

DAFTAR PUSTAKA

[1] M. Mustaqhfiri, Z. Abidin, and R. Kusumawati, "Peringkasan Teks Otomatis Berita Berbahasa Indonesia Menggunakan Metode Maximum Marginal Relevance," *Matics*, no. March 2012, 2012.

[2] N. Bhatia and A. Jaiswal, "Trends in Extractive and Abstractive Techniques in Text Summarization," *Int. J. Comput. Appl.*, vol. 117, no. 6, pp. 975–8887, 2015.

[3] F. Kyoomarsi, H. Khosravi, E. Eslami, P. K. Dehkordy, and A. Tajoddin, "Optimizing Text Summarization Based on Fuzzy Logic," *Seventh IEEE/ACIS Int. Conf. Comput. Inf. Sci. (icis 2008)*, vol. 5, pp. 347–352, 2008.

[4] R. M. Alguliev, R. M. Aliguliyev, and N. R. Isazade, "DESAMC+DocSum: Differential evolution with self-adaptive mutation and crossover parameters for multi-document summarization," *Knowledge-Based Syst.*, vol. 36, pp. 21–38, 2012.

[5] J. P. Mei and L. Chen, "SumCR: A new subtopic-based extractive approach for text summarization," *Knowl. Inf. Syst.*, vol. 31, no. 3, pp. 527–545, 2012.

[6] K. Nandhini and S. R. Balasundaram, "Extracting easy to understand summary using differential evolution algorithm," *Swarm Evol. Comput.*, vol. 16, pp. 19–27, 2014.

[7] A. T. Wibowo and G. Septiana, "Pembobotan Fitur Ekstraksi Pada Peringkasan Teks Bahasa Indonesia Menggunakan Algoritma Genetika," pp. 1–8.

[8] Aristoteles, Y. Herdiyeni, A. Ridha, and J. Adisantoso, "Text Feature Weighting for Summarization of Documents in Bahasa Indonesia Using Genetic Algorithm," *IJCSI Int. J. Comput. Sci.*, vol. 9, no. 3, pp. 1–6, 2012.

[9] Y. A. Jaradat and A. T. Al-Taani, "Hybrid-based Arabic single-document text summarization approach using genatic algorithm," *2016 7th Int. Conf. Inf. Commun. Syst. ICICS 2016*, pp. 85–91, 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] S. Karwa and N. Chatterjee, "Discrete differential evolution for text summarization," *Proc. - 2014 13th Int. Conf. Inf. Technol. ICIT 2014*, vol. 1, pp. 129–133, 2014.

[12] I. A. Fajarwati and W. Anggraeni, "Penerapan Algoritma Differential Evolution untuk Penyelesaian Permasalahan Vehicle Routing Problem with Delivery and Pick-up," *J. Tek. ITS*, vol. 1, no. 2301–9271, pp. A391–A396, 2012.

[13] R. Nurdiansyah, J. T. Industri, and F. T. Industri, "Pengembangan Algoritma Differential Evolution Untuk Penjadwalan Flow Shop Multi Obyektif," pp. 41–49, 2011.

[14] B. Mohanty, S. Panda, and P. K. Hota, "Controller parameters tuning of differential evolution algorithm and its application to load frequency control of multi-source power system," *Int. J. Electr. Power Energy Syst.*, vol. 54, pp. 77–85, 2014.

[15] A. Abuobieda, N. Salim, M. S. Binwahlan, and A. H. Osman, "Differential evolution cluster-based text summarization methods," *Proc. - 2013 Int. Conf. Comput. Electr.*

Electron. Eng. 'Research Makes a Differ. ICCEEE 2013, pp. 244–248, 2013.

[16] S. Das and P. N. Suganthan, "Differential Evolution : A Survey of the State-of-the-Art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, 2011.

[17] H. P. Edmundson, "New methods in automatic extracting," *J. Assoc. Comput. Mach.*, vol. 16, no. 2, pp. 264–285, 1969.

[18] I. N. Sari, A. P. Kurniati, and I. L. Hakim, "ANALISIS DAN IMPLEMENTASI PERINGKAS TEKS OTOMATIS MENGGUNAKAN METODE MAXIMAL MARGINAL IMPORTANCE (MMI)," no. Mmi, 2011.

[19] J. Pedersen, X. Palo, and P. Alto, "A Trainable Document Summarizer."

[20] D. R. Radev, "LexRank : Graph-based Centrality as Salience in Text Summarization," vol. 22, pp. 457–479, 2004.

[21] R. Ferreira *et al.*, "Assessing sentence scoring techniques for extractive text summarization," *Expert Syst. Appl.*, vol. 40, no. 14, pp. 5755–5764, 2013.

[22] S. Tofighy Mohsen, R. Gopal Raj, and H. Seyyed Haj Javadi, "AHP Techniques for Persian Text Summarization," vol. 26, no. 1, pp. 1–8, 2013.

[23] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[24] B. Zaman and E. Winarko, "Analisis Fitur Kalimat Untuk Peringkas Teks Otomatis Pada Bahasa Indonesia," *Ijccs*, vol. 5, no. 2, pp. 60–68, 2011.