

USULAN TUGAS AKHIR

1. IDENTITAS PENGUSUL

NAMA : Afif Fauzi Aulia Ur Rahman
NRP : 5110100055
DOSEN WALI : Dr. Ir. Siti Rochimah, M.T.
DOSEN PEMBIMBING : 1. Diana Purwitasari, S.Kom., M.Sc.
2. Arya Yudhi Wijaya, S.Kom., M.Kom.

2. JUDUL TUGAS AKHIR

“Aplikasi Pembangkit Ringkasan Dokumen yang Mudah Dipahami dengan Menggunakan Algoritma *Differential Evolution* yang Dimodifikasi”

3. LATAR BELAKANG

Proses meringkas memiliki tujuan utama untuk mengurangi kompleksitas. Judul, kata kunci, daftar isi, dan abstraksi bisa dikategorikan sebagai salah satu bentuk dari ringkasan. Ringkasan dokumen adalah pengurangan panjang dan kompleksitas suatu dokumen yang utuh. Ringkasan mempunyai bentuk sebagai abstraksi hasil dari ringkasan suatu dokumen tersebut adalah dokumen yang pendek namun lebih banyak pemaparan daripada judul dokumen tersebut [1].

Permasalahan yang sering terjadi dalam proses peringkasan dokumen secara otomatis adalah banyaknya informasi yang *redundant* yang ada dalam hasil ringkasan dokumen. Permasalahan lain adalah rendahnya tingkatnya kemudahan membaca suatu hasil ringkasan [2].

Melalui Tugas Akhir ini penulis akan melakukan implementasi proses meringkas dengan pendekatan statistika dasar sebagaimana ditunjukkan dengan *term frequency-inverse sentence frequency* (TF-ISF). TF-ISF adalah kombinasi dari TF dan ISF. TF adalah frekuensi suatu *term*/kata pada suatu kalimat. ISF adalah frekuensi kalimat mengandung *term* tersebut. Kemudian diukur tingkat kemiripan antar kata dan kalimat dengan *Cosine Similarity*.

Dengan ini penulis mengusulkan penggunaan algoritma *Differential Evolution* (DE) yang sudah dimodifikasi digunakan untuk membentuk pilihan kalimat yang akan dibentuk menjadi suatu ringkasan yang mudah dibaca. Modifikasi pada algoritma *Differential Evolution* ada pada *crossover* dan *mutation*.

4. RUMUSAN MASALAH

Tugas akhir ini mempunyai beberapa macam rumusan masalah meliputi hal-hal seperti di bawah ini:

- a. Meringkas dokumen dengan proses TF-ISF dan *Cosine Similarity* sehingga bisa digunakan untuk menghasilkan ringkasan yang mudah dibaca.
- b. Merancang metode *Differential Evolution* yang dimodifikasi yang dapat digunakan untuk mengoptimalkan hasil ringkasan.
- c. Melakukan implementasi *Differential Evolution* yang dimodifikasi dalam optimasi hasil ringkasan dokumen.

5. BATASAN MASALAH

Batasan masalah yang ada pada Tugas Akhir ini antara lain:

- a. Data teks yang digunakan dalam Tugas Akhir ini menggunakan bahasa Inggris.
- b. *Dataset* didapatkan dari http://www.samacheerkalvi.in/samacheerkalvi_10th_Textbook.php dan menggunakan dokumen mengenai ilmu sains dan ilmu sosial.
- c. Parameter yang digunakan dalam proses optimasi adalah *mutation* (F) dan *crossover* (CR).
- d. Perangkat lunak yang digunakan dalam proses pengerjaan Tugas Akhir ini menggunakan Scilab.

6. TUJUAN PEMBUATAN TUGAS AKHIR

Tujuan dari pembuatan Tugas Akhir ini antara lain:

- a. Mempelajari proses DE dalam menyelesaikan permasalahan peringkasan dokumen.
- b. Mengimplementasikan ekstraksi ringkasan suatu dokumen yang dioptimasi dengan menggunakan algoritma DE.

- c. Menganalisis kinerja proses ringkasan suatu dokumen yang dioptimasi dengan menggunakan algoritma DE dengan melakukan proses uji coba.

7. MANFAAT TUGAS AKHIR

Manfaat yang diharapkan dari hasil pembuatan Tugas Akhir ini adalah membantu membuat ringkasan yang mudah dipahami suatu dokumen yang berguna bagi orang yang mengalami dalam kesulitan membaca.

8. TINJAUAN PUSTAKA

a. *Cosine Similarity*

Cosine Similarity sendiri adalah salah satu metode yang terkenal dan banyak digunakan dalam proses *clustering* dokumen teks. Di mana dengan menghitung kemiripan dua kalimat dengan menghitung kosinus dari sudut dua vektor/kalimat. Untuk mendapatkan nilai dari bobot tiap kalimat, digunakan *term frequency-inverse sentence frequency* (TF-ISF) terlebih dahulu [3].

$$\text{Cosine}(S_a, S_b) = \frac{\sum_{w_i \in S_a, S_b} (w_i, S_a) \cdot (w_i, S_b)}{\sqrt{\sum_{w_i \in S_a} (w_i, S_a)^2} \sqrt{\sum_{w_i \in S_b} (w_i, S_b)^2}} \quad (1)$$

S = Sentence/kalimat.

S_a = kalimat ke- a .

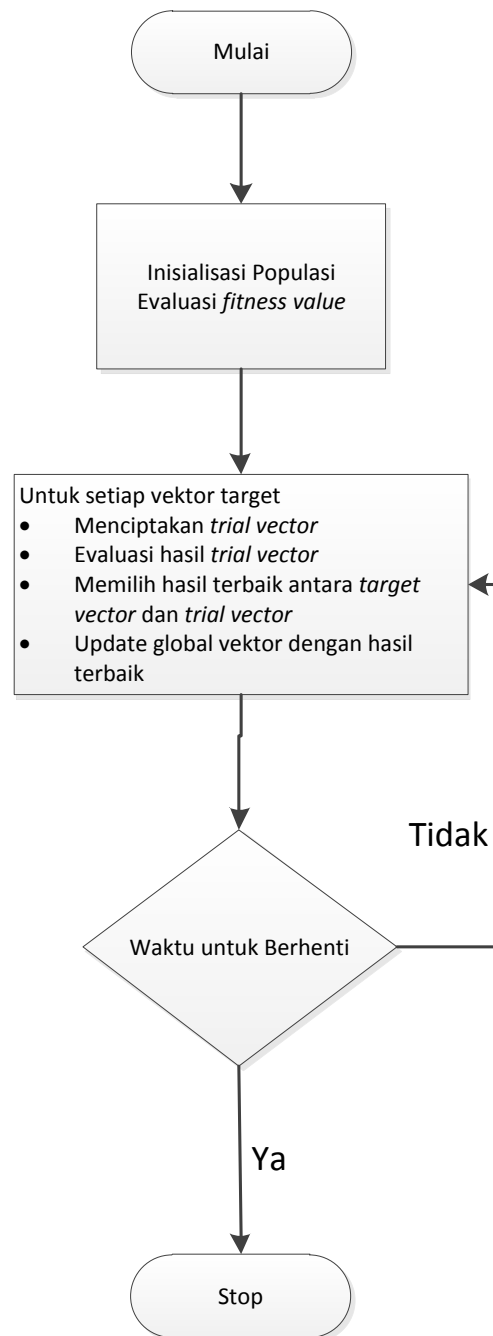
(w_i, S_a) = bobot *term*/kata di S_a .

b. *Differential Evolution*

Differential Evolution (DE) termasuk dalam keluarga *Evolutionary Algorithm*. DE diperkenalkan oleh Storn dan Price pada tahun 1995 [4]. DE merupakan evolusi dari *Genetic Algorithm* (GA) dengan mengganti operator logika dengan operator matematis.

Perbedaan utama antara DE dan GA (algoritma serupa) adalah mekanisme untuk menghasilkan solusi baru dengan menggabungkan beberapa solusi dengan kandidat solusi. Solusi-solusi dari DE didapat dari siklus yang berulang dari 3 operator utama DE, yaitu *mutation*, *crossover* dan *selection*. Proses bisa dilihat di Gambar 1.

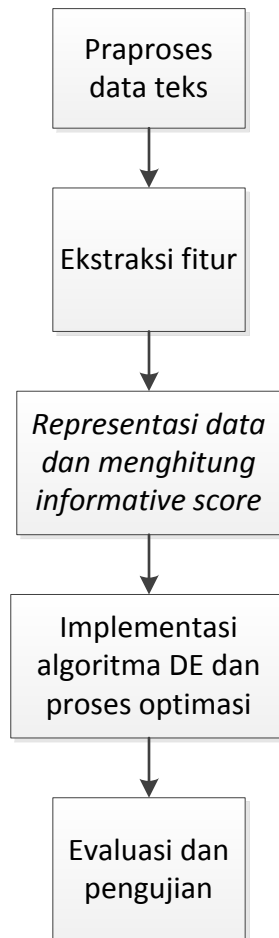
Proses utama pada DE adalah menciptakan *mutant vector* dari 3 vektor populasi yang dirumuskan $V = X_1 + F(X_2 - X_3)$ yang langkah tersebut disebut *mutation*. Langkah selanjutnya adalah menciptakan *trial vector* dari *crossover* antara *mutant vector* dan *target vector*. Kemudian dilakukan proses *selection* dimana bila *trial vector* lebih bagus hasilnya daripada *target vector* [5].



Gambar 1. Bagan Proses Algoritma *Differential Evolution*

9. RINGKASAN ISI TUGAS AKHIR

Proses pengerjaan Tugas Akhir ini digambarkan dalam Gambar 2 dan uraian singkat di bawah ini:



Gambar 2. Bagan Proses Pengerjaan Tugas Akhir

a. Praproses Data Teks

Praproses data dilakukan untuk mengolah data agar dapat diproses dengan baik selama implementasi metode DE.

1. *Sentence Segmentation*: membagi dokumen teks menjadi tiap kalimat.
2. *Tokenization*: membagi tiap kalimat dalam dokumen menjadi tiap *term*/kata.

3. *Stopword Removing*: menghilangkan *term*/kata yang tidak penting atau tidak bermakna.
4. *Syllable Counting*: menghitung jumlah suku kata tiap *term*/kata.
5. *Stemming*: mengubah *term*/kata menjadi bentuk dasar.
6. Menghitung TF-ISF.

b. Ekstraksi Fitur

Ekstraksi fitur ini disesuaikan dengan tujuan pembuatan Tugas Akhir ini yaitu mempermudah memahami isi bacaan, sehingga ekstraksi fitur disesuaikan dengan tujuan tersebut.

Maka dari itu, fitur yang diambil adalah *sentence position*, *centrality*, *title similarity*, dan *keyword occurrences*. Fitur tersebut termasuk dalam *informative features*. Sedangkan *readability features* mencakup *sentence length*, *word length*, dan *noun occurrences*.

c. Representasi Data dan Menghitung Informative Score

Data yang telah dipraproses digambarkan dalam bentuk *Vector Space Model* dengan menyimpan bobot tiap *term*. Data digambarkan dalam bentuk *Vector Space Model* dengan $S_i = (w_{i1}, w_{i2}, \dots, w_{ik})$. S adalah *sentence*. Sedangkan w_{ik} adalah bobot/*weight* dari *term*/kata ke- k di kalimat/*sentence* ke- i .

Nilai tersebut bisa dihitung dengan kombinasi linear antara *feature* dan bobot yang ada. Bisa dirumuskan dengan $\sum_{i=1}^n W_i * F_i$ di mana W adalah bobot suatu kata sedangkan F adalah nilai tiap *feature* seperti yang disebutkan pada Bab 9 bagian B.

Setelah itu dilakukan penghitungan kemiripan setiap kalimat dengan kalimat lain dengan menggunakan *Cosine Similarity*. *Cosine Similarity* menghitung kosinus sudut dari dua vektor.

d. Implementasi Algoritma DE dan Proses Optimasi

Langkah ini bertujuan untuk maksimasi rata-rata *informative score* dari ringkasan yang mempunyai kemiripan yang besar menggunakan metode *Differential Evolution* dengan memodifikasi *crossover* dan *mutation*. Awalnya konstanta *crossover* adalah $CR \in (0.0, 1.0)$ dan konstanta *mutation* adalah $F \in (0.0, 1.0+]$ namun kedua konstanta tersebut telah dimodifikasi sehingga diharapkan menghasilkan performa yang baik.

Bagian yang dimodifikasi yaitu parameter *mutation* dan *crossover* tanpa mengubah batasan dari masing-masing parameter tersebut. Modifikasi algoritma DE dijelaskan pada Gambar 3 dan Gambar 4 [6].

```

Algorithm 1. Modifikasi mutation
1. procedure Modified Mutation
2. Select two random chromosome  $\{X_{ra}, X_{rc}\}$  from P and the current best chromosome  $X_{rb}$ 
3.   for  $i \leftarrow 2, n - 1$  do
4.      $M[i] = (Diff(X_{ra}[i], X_{rc}[i]))$ 
5.   end for
6.   for  $i \leftarrow 2, n - 1$  do
7.     if  $(M[i] < > 0)$  then
8.       if  $((X_{rb}[i+1]) - (X_{rb}[i-1]) > 2)$  then
9.         Generate Rnum
10.        if  $((R_{num} > (X_{rb}[i-1])) \text{ .AND. } (R_{num} < (X_{rb}[i+1])) \text{ .AND. } (R_{num} < > (X_{rb}[i])))$  then
11.           $X_{rb}[i] = R_{num}$ 
12.        end if
13.      end if
14.    end if
15.  end for
16. end procedure.

```

Gambar 3. *Pseudocode* Modifikasi *Mutation* pada DE

```

Algorithm 2. Modifikasi crossover
1. procedure Modified Crossover
2. Find the potential crossing sites
3.   for  $i \leftarrow 2, n - 1$  do
4.     if  $((X_{rb}[i] > V_g[i]) \text{ .AND. } (X_{rb}[i] < V_g[i+1]))$  then ► Potential Crossing Sites
5.        $CS[k] = i$ 
6.     end if
7.   end for
8.   Rand(CR) ► Generate a random number for performing crossover in random position
9.   for  $i \leftarrow 1$  do
10.     $U_{gi} = X_{rb}[1 : i] \oplus V_g[i+1 : n]$ 
11.  end for
12. end procedure.

```

Gambar 4. *Pseudocode* Modifikasi *Crossover* pada DE

e. Evaluasi dan Pengujian

Proses mengukur tingkat kohesi adalah dengan menggunakan *Cosine Similarity*. Sedangkan pengukuran terhadap kemudahan dibaca menggunakan Flesch Kincaid Readability.

Uji Flesch Kincaid adalah uji coba seberapa mudah suatu dokumen berbahasa Inggris dibaca dengan faktor berupa panjang kalimat, panjang kata, dan suku kata. Uji Flesch Kincaid dapat dirumuskan dengan Persamaan 2.

$$206.835 - 1.015 \frac{\text{total kata}}{\text{total kalimat}} - 84.6 \frac{\text{total suku kata}}{\text{total kata}} \quad (2)$$

Uji Flesch Kincaid mempunyai hasil nilai dengan rentang antara 0.0 sampai dengan 100.0. Semakin besar nilainya maka semakin mudah dibaca. Nilai 90-100 mudah dipahami oleh anak usia sekitar 11 tahun. Sedangkan hasil uji Flesch Kincaid dengan hasil di bawah 30.0 dipahami secara maksimal oleh mahasiswa perguruan tinggi [7].

Evaluasi yang dilakukan adalah dengan membandingkan tingkat kohesi dan *readability*. Perbandingan dilakukan antara performa algoritma *Differential Evolution* yang dimodifikasi dengan metode *Differential Evolution*.

10.METODOLOGI

a. Penyusunan Proposal Tugas Akhir

Tahap pertama dari penyusunan Tugas Akhir adalah dengan membuat proposal Tugas Akhir. Proposal ini berisikan gagasan penulis untuk menyelesaikan masalah terhadap orang yang mempunyai kesulitan membaca dengan membuat ringkasan suatu dokumen yang dapat dibaca.

b. Studi Literatur

Tahap selanjutnya adalah melakukan studi dan pemahaman terhadap konsep yang bisa dijadikan referensi dalam pengerjaan Tugas Akhir. Studi literatur ini bersumber dari buku, artikel ilmiah, internet, materi kuliah yang sesuai dengan Tugas Akhir ini.

c. Implementasi Algoritma

Implementasi merupakan tahap utama untuk membangun sistem yang telah dirancang sebelumnya. Algoritma yang akan digunakan adalah *Differential Evolution* yang dimodifikasi.

d. Pengujian dan Evaluasi

Pengujian dilakukan dengan cara membandingkan kinerja dari algoritma *Differential Evolution* dengan algoritma DE yang sudah dimodifikasi.

e. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam Tugas Akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Secara garis besar, sistematika penulisan buku Tugas Akhir adalah sebagai berikut:

1. Pendahuluan
 - a. Latar Belakang
 - b. Rumusan Masalah
 - c. Batasan Tugas Akhir
 - d. Tujuan
 - e. Metodologi

- f. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

11. JADWAL KEGIATAN

Berikut ini ditampilkan jadwal kegiatan pengerjaan Tugas Akhir ini pada Tabel 1.

Tabel 1. Jadwal Kegiatan Pengerjaan Tugas Akhir

Tahapan	Tahun																	
	Februari				Maret				April				Mei				Juni	
Penyusunan Proposal	■	■	■	■														
Studi literatur		■	■	■	■	■	■	■										
Implementasi									■	■	■	■	■	■	■	■	■	
Pengujian dan evaluasi									■	■	■	■	■	■	■	■	■	
Penyusunan buku											■	■	■	■	■	■	■	■

12. DAFTAR PUSTAKA

- [1] J. Kupiee, J. Pedersen dan F. Chen, "A Trainable Document Summarizer," dalam *SIGIR '95 Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1995.
- [2] K. S. Jones, "Automatic Summarising : The State of The Art," *Information Processing and Management*, vol. 43, pp. 1449-1481, 2007.
- [3] R. M. Alguliev, R. M. Aliguliyev dan C. A. Mehdiyev, "Sentence Selection for Generic Document Summarization Using an Adaptive Differential Evolution Algorithm," *Swarm and Evolutionary Computation*, vol. 1, pp. 213-222, 2011.

- [4] R. Storn dan K. Price, "Differential Evolution - A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces," *Journal of Global Optimization*, vol. 11, pp. 341-359, 1997.
- [5] G. C. Onwubolu dan B. V. Babu, *New Optimization Techniques in Engineering*, Springer, 2004.
- [6] K. Nandhini dan S. R. Balasundaram, "Extracting Easy to Understand Summary Using Differential Evolution Algorithm," *Swarm and Evolutionary Computation*, 2013.
- [7] Wikipedia.org, "Wikipedia - Flesch–Kincaid readability tests," [Online]. Available: http://en.wikipedia.org/wiki/Flesch%E2%80%93Kincaid_readability_tests. [Diakses 5 Maret 2014].