IMPLEMENTASI METODE DIFFERENTIAL EVOLUTION PADA PERINGKASAN TEKS BERBAHASA INDONESIA

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

ABSTRAK

Dalam beberapa penelitian mengenai peringkasan teks, beberapa permasalahan utama yang muncul dalam bidang peringkasan teks otomatis adalah relevansi dan pokok ringkasan. Hasil ringkasan yang baik harus mengandung informasi yang penting tanpa mengabaikan relevansi dan pokok ringkasannya. Untuk itu diajukan sebuah metode untuk mengoptimalkan fitur-fitur yang diekstraksi dari sebuah dokumen berdasarkan skor informative dan cohesive agar didapatkan hasil ringkasan yang baik yaitu metode Differential Evolution. Metode Differential Evolution merupakan metode optimasi yang baru-baru ini dikembangkan dan terbukti memiliki performansi yang baik. Berdasarkan hasil tersebut, Differential Evolution digunakan dalam penelitian ini untuk menghasilkan ringkasan teks berbahasa indonesia yang memiliki struktur relevansi yang paling baik. Pada penelitian ini diekstrak 5 buah fitur pada dokumen yaitu posisi kalimat, frekuensi kata, kata kunci, kemiripan dengan judul dan sentralitas. Pengujian yang dilakukan adalah pengujian akurasi yang dilakukan dengan membandingkan hasil ringkasan sistem dengan hasil ringkasan manual yang didapat dari ahli bahasa Indonesia.

Pengujian sistem menghasilkan rata-rata akurasi 47,02% dengan hasil terbaik 83,33%. Sedangkan berdasarkan pengujian ROUGE-1 didapat rata-rata nilai *f-measure* 57,22% dengan hasil terbaik 91,46%, pengujian ROUGE-2 didapat rata-rata nilai *f-measure* 49,66% dengan hasil terbaik 92,31%

Kata kunci : Differential Evolution, Peringkasan Teks Otomatis, NLP, Algoritma, Optimasi.

1. PENDAHULUAN

Peringkasan teks otomatis (automatic text summarization) adalah pembuatan bentuk yang lebih singkat dari suatu teks dengan memanfaatkan aplikasi yang dijalankan dan dioperasikan pada komputer [1]. Sedangkan menurut N. Bhatia dan A. Jaiswal, ringkasan adalah teks yang dihasilkan dari sebuah teks atau banyak teks, yang mengandung isi informasi dari teks asli dan panjangnya tidak lebih dari setengah panjang teks aslinya[2]. Penelitian peringkasan teks otomatis umumnya dilakukan dengan menggunakan 2 cara dalam teknik pengklasifikasiannya yaitu ekstraktif dan abstraktif [2]. Peringkasan ekstraktif tidak berfokus dalam memahami teks. Tetapi fokusnya adalah dalam mengekstrak bagian paling penting berdasarkan fitur statistikal dan linguistik seperti kata isyarat, lokasi dan frekuensi dari sebuah kata [3]. Sedangkan peringkasan abstraktif memasukan bagian parafrase dari sumber dokumen. Pada umumnya, peringkasan abstraktif dapat meringkas sebuah teks dengan lebih baik dibandingkan dengan peringkasan ekstraktif. Tapi sistem yang dapat melakukannya akan lebih sulit untuk dibangun karena membutuhkan penggunaan teknologi natural language generation, yang merupakan bidang yang masih berkembang.

Beberapa permasalahan utama yang muncul dalam bidang peringkasan teks otomatis adalah relevansi dan pokok dari ringkasan yang masih dapat dikembangkan lagi [4][5]. Hasil ringkasan yang baik harus mengandung informasi yang penting tanpa mengabaikan relevansi dan pokok ringkasannya[4][5].

Berdasarkan penelitian dari K. Nandhini [6] telah dilakukan perbandingan antara Differential Evolution (DE) dan Algoritma Genetik (AG) dalam mengoptimalkan fitur-fitur yang diekstraksi dari sebuah dokumen berdasarkan skor informative dan cohesive sebagai fungsi fitnessnya. Hasil dari perbandingan tersebut menghasilkan kesimpulan bahwa DE menghasilkan hasil ringkasan yang lebih baik dibandingkan dengan AG. Berdasarkan hasil tersebut, Differential Evolution digunakan dalam penelitian ini untuk menentukan performansinya dalam menghasilkan ringkasan teks berbahasa indonesia yang memiliki struktur relevansi yang paling baik[6]. Hasil lain yang didapatkan dari penggunaan AG dalam peringkasan teks adalah penelitian yang dilakukan oleh Zulkifli dkk yang menghasilkan akurasi ringkasan 53%[7], penelitian yang dilakukan oleh Aristoteles dkk menghasilkan akurasi 40%[8], penelitian yang dilakukan oleh Jaradat menghasilkan nilai *f-measure* 54,8%[9], Sedangkan hasil yang didapat oleh K. Nandhini dalam penggunaan AG menghasilkan rata-rata akurasi pengujiannya 60%[6][10]. Hasil pengujian lain dalam penggunaan DE adalah penelitian yang dilakukan oleh R. Alguliev dkk menghasilkan nilai pengujian ROUGE-1 43%[4], penelitian yang

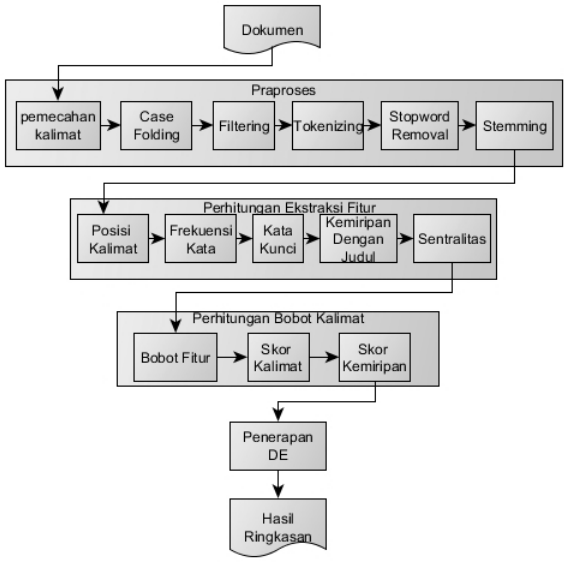
dilakukan oleh Shweta dengan *Discrete* DEnya menghasilkan rata-rata f-measure 36,33%[11], sedangkan Fajarwati dengan permasalahn VRP-DPnya berhasil menurunkan total biaya perjalanan sebesar 18% dengan DE[12], Nurdiansyah dengan penjadwalan *flowshop* multi obyektif dengan banyak mesin menggunakan DEnya berhasil mendapatkan rata-rata presentasi relatif yang lebih baik dibandingkan dengan pengujian lain yaitu selalu dibawah 1%[13], Mohanty dkk berhasil membangun sebuah sistem LFC terhadap sistem energi banyak sumber dengan DE sebagai kontrol *tuning* parameternya yang menghasilkan sistem yang konsisten dengan tidak terganggu terhadap perubahan kondisi *loading* dan performansi kontrolnya terlihat lebih baik[14], sedangkan penelitian yang dilakukan oleh K. Nandhini dalam penggunaan DE terhadap peringkasan teks menghasilkan rata-rata akurasi 71%[6].

Penelitian ini menggunakan metode Differential Evolution (DE) yaitu sebuah metode yang berdasarkan populasi yang mengoptimasikan sebuah permasalahan dengan cara iteratif mengembangkan solusi kandidat dengan bobot kualitas pengukuran. Pada dasarnya DE merupakan algoritma yang mirip dengan Algoritma Genetik (AG), tapi perbedaannya AG bergantung pada crossover dalam membangun solusinya sedangkan DE bergantung pada operasi mutasinya[4]. Disamping itu, DE merupakan algoritma yang efisien dan mudah digunakan dibandingkan dengan algoritma evolustionary lainnya[15].

2. METODE

Pada penelitian ini digunakan sebuah metode Differential Evolution untuk peringkasan teks berbahasa Indonesia, dimana metode Differential Evolution merupakan metode optimasi[16]. Tahapan awal dari penelitian ini adalah analisis data masukan dimana data yang dimasukan harus berbentuk teks. Selanjutnya data masukan dilakukan proses Preprocessing, lalu fitur-fitur dari data diekstrak, kemudian pada tahap akhir fitur-fitur tersebut diolah menggunakan Differential Evolution untuk menghasilkan sebuah ringkasan teks.

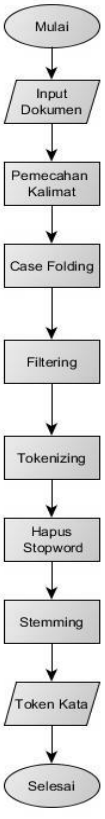
Arsitektur sistem peringkasan teks otomatis yang dibangun memliki 4 tahapan utama yaitu praproses, ekstraksi fitur, menghitung bobot kata, dan menerapkan Differential Evolution. Untuk penjelasan setiap tahap pada proses peringkasan dapat dilihat pada gambar 1 berikut:



Gambar 1. Gambaran Sistem

2.1 Praproses

Praproses atau *Preprocessing* merupakan tahap dimana dokumen yang akan digunakan dibersihkan dari noise atau dari hal yang tidak mempunyai pengaruh dalam ringkasan. Praproses yang dilakukan dalam tahap ini antara lain: pemecahan kalimat, case folding, filtering, tokenizing kata, hapus stopword dan stemming. Berikut merupakan tahapan dari praproses pada gambar 2:



Gambar 2. Praproses

2.2 Ekstraksi Fitur

Bagian ini menjelaskan proses ekstraksi fitur-fitur dari dokumen yang sudah melalui tahap praproses. Fitur-fitur yang diekstrak diantaranya posisi kalimat dalam dokumen, frekuensi kata, kata kunci, kemiripan dengan judul, dan sentralitas.

2.2.1 Posisi Kalimat

Fitur ini didasarkan pada asumsi bahwa kalimat pertama dan kalimat terakhir pada paragraf merupakan bagian paling penting[6]. Hasil ini didapatkan melalui rumus (1) berikut:

$$Posisi\ Kalimat = Max(\frac{1}{i}, \frac{1}{n-i}) \quad (1)$$

Dimana:
n = jumlah kalimat
i = urutan kalimat

2.2.2 Frekuensi Kata

Kebermaknaan suatu kata dapat ditentukan melalui frekuensi kemunculannya di suatu paragraf. Untuk itu digunakan sebuah metode untuk mengekstrak frekuensi kata dari paragraf yaitu tf-isf. Tf-isf adalah sebuah metode statistikal yang digunakan untuk mengevaluasi kebermaknaan suatu kata di sebuah paragraf dokumen. Metode ini dapat mengekstrak bobot dari setiap kata setelah kata stopword dihilangkan dari teks. Tf-isf dapat dihitung dengan rumus (2) dan (3) berikut[6]:

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

Dimana :
N = jumlah kalimat di dokumen
n = jumlah kalimat yang mengandung kata ke-i

$$Frekuensi\ Kata = Tf * isf \quad (3)$$

2.2.3 Kata Kunci

Awalnya, Kata Kunci didefinisikan sebagai kata non-isyarat dengan bobot terukur yang tertinggi, dan bobot kunci yang digunakan menjadi frekuensi kemunculan di sebuah dokumen[17]. Dalam penelitian ini, fitur kata kunci yang diekstrak didapatkan dari ≤ 10 kata yang sering muncul dalam sebuah teks. Setelah kata kunci didapatkan, setiap token dalam kalimat yang sama dengan kata kunci akan diberi nilai 0,1 dan yang tidak sama akan diberi nilai 0[18].

2.2.4 Kemiripan Dengan Judul

Judul mengandung kumpulan kata yang memberikan arah penting tentang konsep dari paragraf [19]. Kecocokan dari setiap kata pada judul tersebut dihitung dengan rumus (4) berikut:

$$Kemiripan = \frac{|Kata\ di\ judul \cap kata\ di\ Kalimat(i)|}{|Kata\ di\ judul \cup kata\ di\ Kalimat(i)|} \quad (4)$$

Berikut merupakan contoh perhitungan nilai kemiripan pada sebuah dokumen dengan kalimat pertamanya pada tabel 1 dan 2 berikut:

Tabel 1. Token Judul

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. Contoh Pencocokan Dengan Kalimat

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 Sentralitas

Sentralitas dari sebuah kalimat menyatakan kemiripannya ke kalimat lain [20]. Disebutkan juga pada jurnal [21] bahwa sentralitas merupakan tumpang tindih kosa kata antara dua buah kalimat. Bila sebuah kalimat memiliki sentralitas yang lebih tinggi, maka kalimat tersebut merupakan kandidat terbaik untuk di masukkan ke hasil ringkasan. Maka dari itu, dalam penelitian ini untuk mencari nilai sentralitas dari sebuah kalimat maka dilakukan pencocokan terhadap kalimat-

kalimat lainnya sampai ditemukan nilai sentralitas terbesar yang dimiliki dari suatu kalimat. Nilainya dihitung berdasarkan rumus (5) berikut:

$$Sentralitas = \frac{|Kata\ di\ Kalimat(i) \cap kata\ di\ Kalimat(j)|}{|Kata\ di\ Kalimat(i) \cup kata\ di\ Kalimat(j)|} \quad (5)$$

Dimana:
i = indeks kalimat ke-i
j = indeks kalimat lain selain i

Berikut merupakan contoh perhitunganya pada tabel 3:

Tabel 3. Contoh Perhitungan Sentralitas

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 Perhitungan Bobot Kalimat

Perhitungan bobot kalimat dilakukan sebagai data masukan untuk fungsi objektif di Differential Evolution. Untuk perhitungannya dilakukan melalui 3 tahapan yaitu bobot fitur, skor kalimat dan skor kemiripan.

2.3.1 Bobot Fitur

Untuk menghitung bobot fitur, digunakan sebuah matriks pembanding yang akan membandingkan nilai setiap fitur dengan fitur lainnya. Matriks tersebut dapat digambarkan seperti pada tabel 4 berikut[22]:

Tabel 4. Matriks Bobot Fitur

Fitur	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

Dalam penelitian ini, fitur-fitur yang diekstrak dari dokumen adalah:
F1 = Posisi Kalimat
F2 = Frekuensi Kata
F3 = Katakunci
F4 = Kemiripan dengan judul
F5 = Setnralitas

Nilai fitur-fitur tersebut kemudian dirata-ratakan untuk dimasukkan kedalam matriks pembanding dengan menggunakan rumus (6) berikut:

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

Dimana:
Fi = Fitur ke-i
i = urutan fitur
j = urutan kalimat
n = jumlah kalimat

Kemudian nilai dari bobot fitur tersebut dinormalisasikan dengan cara membagi setiap nilai Fi/Fj dengan W dibawahnya. Kemudian nilai tersebut dijumlahkan kembali secara vertikal kebawah untuk mendapatkan nilai bobot fitur.

2.3.2 Skor Kalimat

Skor dari setiap kalimat di dokumen digunakan sebagai salah satu fungsi fitness yang akan digunakan di Differential Evolution. Skor dari setiap kalimat dalam dokumen dapat dihitung menggunakan rumus (7) berikut[6][10]:

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

Dimana :
Wi = Bobot fitur ke-i
Fi = Fitur ke-i
n = Jumlah Fitur

2.3.2 Skor Kemiripan

Pengukuran kemiripan merupakan hal yang penting dalam riset yang berhubungan dengan teks seperti

peringkasan teks, pengklusteran dokumen dan pencarian informasi[6]. *Cosine similarity* merupakan salah satu pengukuran kemiripan dokumen karena kesensitifannya terhadap pola vektor dokumen. *Cosine measure* menghitung kosinus dari sudut antara dua vektor fitur. *Cosine similarity* dapat dihitung dengan rumus (8) berikut:

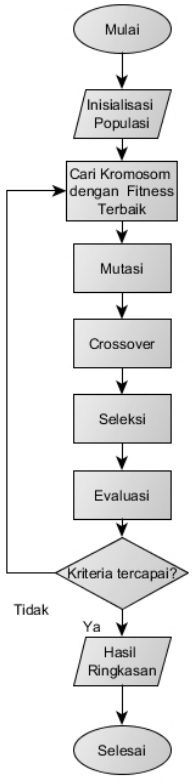
$$Sim(Si, Sl) = \frac{\sum_{j=1}^{n} w_{ij} * w_{lj}}{\sqrt{\sum_{j=1}^{n} (w_{ij})^2} * \sqrt{\sum_{j=1}^{n} (w_{lj})^2}} \quad (8)$$

Dimana :
S = kalimat
W = Fitur ke-j pada kalimat i dan l
i = indeks kalimat
l = indeks kalimat
j = indeks fitur
n = Jumlah fitur

Melalui Cosine similarity maka didapatkan matriks kemiripan antar kalimat yang nantinya akan digunakan untuk menghitung fungsi fitness di Differential Evolution.

2.4 Differential Evolution

Differential Evolution (DE) merupakan algoritma yang baru dikembangkan, simple dan efisien, berbasis populasi heuristik dan memiliki performansi yang baik dalam konvergensi dan operasinya[6]. Berikut merupakan alur utama dari DE pada gambar 3:



Gambar 3. Alur DE

2.4.1 Representasi Individu

DE menggunakan istilah kromosom untuk mewakili calon solusinya. Di DE, kromosom-kromosom tersebut terdiri dari beberapa larik bilangan real atau bilangan bulat. Pada penelitian ini, digunakan bilangan bulat yang merupakan posisi kalimat sebagai alel dari kromosom. Untuk mewakilkan 1 calon solusi, sebuah kromosom membutuhkan beberapa nilai yang kemudian akan didecode menjadi sebuah solusi yang valid dan dapat diukur nilai kelayakannya (fitness). Rancangan kromosom yang digunakan adalah sebagai berikut:

a. Jumlah gen/alel pada kromosom sesuai dengan tingkat kompresi dan jumlah kalimat pada dokumen sumber. Misalkan bila tingkat kompresi 50% dan jumlah kalimat 13 maka gen pada kromosom adalah 7. Setiap gen berupa bilangan bulat yang bernilai >= 0
b. Nilai dalam gen-gen ini kemudian akan di-decode dengan bilangan acak antara 1 sampai jumlah kalimat yang akan disusun berdasarkan urutan kalimatnya. Nilai gen setelah di-decode akan menentukan kalimat mana saja yang akan menjadi bagian dari hasil ringkasan pada sebuah kromosom. Karena teks yang digunakan merupakan teks ilmiah, maka posisi awal dan akhir dari kromosom diposisikan untuk kalimat pertama dan kalimat terakhir pada teks dokumen.

Berikut merupakan contoh kromosom yang akan dibangkitkan berdasarkan dokumen yang memiliki panjang kalimat 10 dengan tingkat kompresi 50% pada gambar 4:

Nilai Alel	1	4	8	9	10
Urutan Alel	1	2	3	4	5

Gambar 4. Contoh Kromosom DE

2.4.2 Inisialisasi Populasi

Pada saat awal sistem berjalan, akan dibangkitkan kromosom-kromosom awal lainnya secara acak berdasarkan kromosom awal hasil decode individu. Misalkan didapatkan hasil decode kromosom seperti pada gambar 4 Maka individu lainnya akan dibangkitkan pula dengan cara yang sama. Banyaknya jumlah calon solusi dalam satu populasi adalah sebuah nilai yang akan diobservasi. Karena untuk DE, semakin kecil ukuran populasi akan semakin kecil keberagaman individunya dan semakin cepat ia berevolusi dan begitu sebaliknya, semakin besar ukuran populasi semakin besar keberagamannya dan semakin lama ia berevolusi[4].

2.4.3 Fungsi Fitness

Fungsi fitness harus dapat mengukur kualitas dari kromosom di populasi dengan akurat. Fungsi fitness digunakan untuk mendapatkan kombinasi dari kalimat yang memiliki skor informatif dan kohesi yang tertinggi. Berikut rumus yang digunakan untuk

mendapatkan skor informatif dan kohesif pada dokumen pada rumus (9) dan (10):

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

Dimana:
N = jumlah kalimat
n = jumlah fitur
W = Bobot Fitur
F = Fitur
i = indeks kalimat
j = indeks fitur

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

Dimana:
n = indeks kalimat
i = indeks kalimat
j = indeks kalimat
S = nilai similarity

Kedua nilai tersebut kemudian dikalikan untuk mendapatkan nilai fitness dari sebuah kromosom seperti pada rumus (11) berikut:

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

2.4.4 Mutasi

DE merupakan algoritma yang berbasis mutasi yang dilakukan setelah populasi diinisialisasikan untuk menghasilkan populasi baru. Proses mutasi pada DE dilakukan dengan menjumlahkan perbedaan yang diperoleh dari beberapa individu yang dipilih secara acak, Berbeda dengan sebagian besar algoritma evolusioner yang melakukan operasi mutasinya dengan didefinisikan oleh fungsi probabilitas[4]. Proses mutasi DE pada penelitian ini dilakukan dengan menggunakan proses mutasi yang dimodifikasi yang digambarkan pada gambar 4 berikut[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_i[i])) then
10:         trial[i] = R;
11:       endif
12:     endif
13:   endif
14: endfor
end procedure
```

Gambar 4. Prosedur Mutasi

Pada awal prosedur, dicari 2 buah indeks yang saling unik satu sama lain dan juga berbeda dengan indeks yang melakukan mutasi. Lalu diihitung nilai M[i] yang merupakan nilai yang digunakan sebagai penentu terjadinya mutasi. R merupakan bilangan bulat yang dibangkitkan secara acak antara alel Xj[i-1] dan Xj[i+1]. Larik trial merupakan larik yang memiliki panjang dan nilai yang sama dengan kromosom j namun digunakan untuk menampung nilai R. Setelah sebuah kromosom melewati proses mutasi, didapatkan sebuah larik trial yang merupakan kombinasi antara nilai alel kromosom tersebut dengan nilai alel baru.

2.4.5 Crossover

Untuk meningkatkan keberagaman dari vektor hasil mutasi maka dilakukanlah crossover[23]. Pada penelitian ini, crossover yang digunakan adalah crossover yang dimodifikasi berdasarkan strategi crossover yang dilakukan oleh Nandhini[6]. Berikut merupakan prosedur yang dilalui larik trial untuk melakukan 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 pada DE dilakukan antara kromosom asal dengan kromosom hasil mutasinya[4][6][16]. Crossover dapat dilakukan dengan mencari titik alel yang berpotensi untuk ber-crossover. Titik tersebut merupakan posisi alel yang memiliki nomor kalimat yang sama atau lebih besar dari alel larik trial dan lebih kecil dari larik trial selanjutnya. Hal tersebut dilakukan untuk menjaga urutan kalimat dan redudansi nomor kalimat pada kromosom trial. randCR merupakan bilangan real acak yang dibangkitkan untuk menentukan probabilitas terjadinya crossover. cR merupakan variabel masukan yang ditentukan pengguna untuk menentukan probabilitas terjadinya crossover.

2.4.6 Seleksi

DE menggunakan prinsip "Survival of the fittest" dalam proses seleksinya. Setelah dilakukan proses mutasi dan crossover, setiap kromosom baru yang dihasilkan dalam suatu generasi akan dilakukan

perbandingan nilai fitnessnya dengan kromosom lamanya [23]. Diantara kedua kromosom tersebut, kromosom yang memiliki nilai fitness yang lebih baik akan menjadi kromosom baru untuk generasi selanjutnya. Apabila seluruh kromosom pada generasi i hasil mutasi dan crossovernya memiliki nilai fitness yang lebih rendah dibandingkan dengan kromosom lamanya maka seluruh kromosom pada generasi i akan bertahan dan maju ke generasi i+1. Seperti misalkan pada kromosom ke-j dihasilkan kromosom mutan/trial yang memiliki nilai fitness 0.025. Sedangkan nilai fitness dari kromosom asalnya adalah 0.026. Maka pada proses seleksi, kromosom mutan akan gugur dan kromosom asal (original) sebelum terjadi mutasi akan tetap bertahan ke generasi selanjutnya.

2.4.7 Kriteria Berhenti

Kriteria berhenti merupakan solusi yang digunakan untuk membatasi DE agar tidak terjadi infinite loop. Kriteria berhenti yang umumnya digunakan untuk menghentikan DE adalah ketika tidak terjadi perubahan dalam beberapa generasi terakhir (local converge) atau ketika jumlah iterasi dari sebuah generasi sudah mencapai angka tertentu[6]. Dalam penelitian ini digunakan jumlah iterasi sebagai kriteria berhenti DE. Namun dalam pengujiannya ditemukan 3 buah dokumen yang titik konvergennya selalu mendekati batas maksimum generasi. Maka dari itu untuk ketiga dokumen tersebut ditambahkan kondisi berhenti tambahan yaitu selama nilai *fitness* terbaik tidak berubah secara signifikan selama (0,2 * generasi) maka *loop* generasi akan dihentikan.

2.4.7 Keluaran DE

Setelah DE berhenti, akan didapatkan sebuah generasi kromosom-kromosm yang berisikan kombinasi dari alel-alel yang dibangkitkan pada tahap awal DE. Kromosom-kromosom tersebut kemudian dicari yang memiliki nilai fitness yang paling tinggi untuk digunakan urutan kalimatnya sebagai hasil ringkasan. Berikut merupakan contoh hasil akhir dari DE terhadap sebuah dokumen yang memiliki panjang kalimat 10 dan tingkat kompresi 50% pada tabel 5:

Tabel 5. Hasil DE

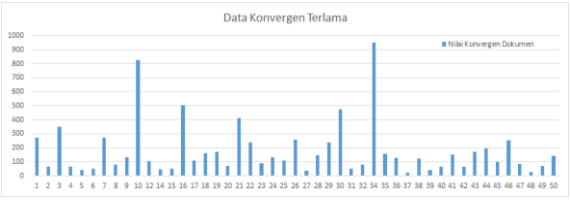
Gen	Kalimat
1	Deteksi manusia pada suatu citra digital dapat dilakukan dengan menggunakan metode dari Dalal dan Triggs pada tahun 2005 yaitu Histograms of Oriented Gradients (HOG)
6	Pada citra yang banyak terdapat noise, hasil perhitungan gradien masih kurang maksimal sehingga dapat mempengaruhi akurasi 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. PEMBAHASAN

Pengujian metode dilakukan dengan tujuan untuk mengetahui hasil performansi dan penerapan metode *Differential Evolution* dalam menghasilkan kalimat hasil ringkasan. Pada pengujian ini akan menggunakan 50 data teks latar belakang dokumen penelitian tugas akhir. Hasil dari ringkasan yang dihasilkan sistem kemudian akan dibandingkan dengan hasil ringkasan manual yang dihasilkan oleh ahli bahasa Indonesia yang merupakan lulusan Sarjana Sastra Indonesia, mengingat studi kasus yang diambil di dalam penelitian ini yaitu dokumen teks berbahasa Indonesia. Lalu dilakukan pula pengujian ROUGE-1 dan ROUGE-2 terhadap dokumen hasil sistem dan hasil manual. Sebelum dilakukan pengujian akurasi dari ringkasan yang dihasilkan oleh sistem, akan dilakukan pencarian parameter DE terbaik. Pada penggunaan metode DE ini terdapat beberapa parameter yang dapat digunakan untuk melihat perbedaan hasil performansi apabila diubah nilainya. Diantaranya crossover rate, populasi dan generasi. Parameter yang memiliki nilai paling baik (memiliki hasil fitness yang paling baik dalam jumlah iterasi yang minimum) akan digunakan untuk mengolah data-data teks dokumen lainnya. Setelah didapatkan parameter DE terbaik pada setiap dokumen, dilakukan pengujian akurasi dengan membandingkan hasil ringkasan manual dengan hasil ringkasan sistem. Data yang digunakan dalam penelitian tugas akhir ini berjumlah 50 dokumen.

Setelah dilakukan pengujian terhadap 1 dokumen yang memiliki panjang rata-rata, didapatkan

parameter *crossover rate* dan populasi yang masing-masing bernilai 0.6 dan 60. Dengan didapatkannya kedua nilai tersebut maka digunakanlah nilai-nilai tersebut sebagai parameter untuk mencari nilai *fitness* terbaik dari setiap dokumen dengan menggunakan parameter generasi yang sama yaitu 1500 generasi. Setelah dilakukan pengujian, didapatkan titik konvergen terlama pada setiap dokumen dalam mencari nilai *fitness* optimal untuk setiap dokumennya beserta nilai *fitness* terbaiknya. Berikut merupakan titik konvergen terlama yang didapatkan dari pengujian terhadap 47 dokumen menggunakan parameter *crossover rate* 0.6 dan populasi 60 pada gambar 6:



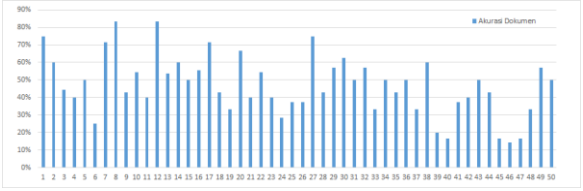
Gambar 6. Grafik Titik Konvergen Dokumen

Berdasarkan hasil yang didapatkan, memang tidak disertakan hasil pengujian untuk dokumen 11, 13 dan 33. Hal tersebut dikarenakan ke-3 dokumen tersebut memiliki titik konvergen yang mendekati 1500 generasi. Sehingga untuk ketiga dokumen tersebut hanya dilakukan pencarian nilai *fitness* terbaiknya saja.

Seperti yang sudah disebutkan sebelumnya, nilai *fitness* terbaik yang didapat akan dibandingkan nilai-nilai kromosomnya dengan hasil ringkasan manual yang dihasilkan oleh ahli bahasa Indonesia. Pembandingan tersebut dilakukan untuk mengukur hasil kedekatan analisis dengan hasil sebenarnya. Pada pengujian akurasi ini, kualitas ringkasan ditentukan oleh *accuracy*. Nilai *accuracy* didapatkan melalui rumus (12) berikut [24]:

$$Accuracy = \frac{\sum Kalimat\ Sistem\ yang\ relevan\ dengan\ Hasil\ Manual}{\sum Kalimat\ Sistem\ yang\ muncul} \quad (12)$$

Pengujian terhadap sistem dilakukan dengan kalimat ringkasan yang memiliki tingkat kompresi 50%. Kalimat-kalimat tersebut kemudian dihitung kemiripannya dengan data ringkasan manual yang didapat dari ahli bahasa Indonesia yang juga diminta untuk memilih setengah dari total kalimat yang dimiliki sebuah dokumen. Berikut merupakan hasil pengujian akurasi pada gambar 7:



Gambar 7. Akurasi setiap dokumen

Seperti yang sudah disebutkan sebelumnya, pengujian juga dilakukan menggunakan ROUGE-1 dan ROUGE-2. Berikut merupakan rata-rata *recall*, *precision* dan *f-measure* yang didapat dari pengujian ROUGE-1 dan ROUGE-2 terhadap 50 dokumen yang diujikan pada tabel 6:

Tabel 6. Hasil Pengujian ROUGE

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

4. KESIMPULAN

4.1 Kesimpulan

Berdasarkan hasil dari penelitian yang telah dilakukan, maka dapat disimpulkan bahwa metode Differential Evolution untuk sistem peringkasan teks ini menghasilkan rata-rata nilai akurasi 47,02% dengan hasil terbaik pengujian 83,33%. Hasil pengujian ROUGE-1 memiliki rata-rata nilai *recall* 51,54%, *precision* 66,61% dan *f-measure* 57,22% dengan hasil terbaik *recall* 92,59%, *precision* 90,36% dan *f-measure* 91,46%. Sedangkan hasil pengujian ROUGE-2 memiliki rata-rata hasil *recall* 44,74%, *precision* 57,99% dan *f-measure* 49,66% dengan hasil terbaik *recall* 93,51%, *precision* 91,14% dan *f-measure* 92,31%. Hasil rata-rata pengujian tersebut merupakan hasil yang rendah karena masih jauh dari 80%. Parameter crossover rate, generasi dan populasi menjadi penentu seberapa besar kombinasi kalimat yang memiliki fitness paling besar didapatkan. Crossover rate menentukan seberapa seringnya suatu kromosom melakukan crossover dengan kromosom trial-nya. Populasi menentukan cepat atau lambatnya kromosom dengan fitness tertinggi ditemukan dan menentukan cepat atau lambatnya populasi menemukan titik konvergennya. Sedangkan generasi menentukan cepat atau lambatnya proses DE berhenti.

4.2 Saran

Berdasarkan beberapa masalah yang didapat dalam pengerjaan penelitian ini maka saran untuk pengembangannya adalah:

1. Menggunakan proses mutasi dan crossover yang berbeda. Karena DE merupakan algoritma yang proses utamanya bergantung pada mutasi dan crossover-nya. Dalam perkembangannya, sudah mulai juga ditemukan proses mutasi dan crossover yang berbeda dari yang diajuhkan oleh Storn dan Price pada saat menemukan DE.
2. Menambahkan kondisi berhenti lainnya untuk metode DE. Karena seperti hasil pengujian dokumen yang didapatkan dengan menambahkan kondisi berhenti DE, didapatkan nilai fitness maksimum dalam titik rata-rata 21,03% dari total generasinya.
3. Menggunakan dataset yang memiliki penggunaan kata dan kalimat yang lebih baik. Karena berdasarkan

pengujian yang dilakukan dengan menggunakan 50 dataset dokumen pada penelitian ini, didapatkan berbagai macam hasil titik konvergen yang dimana selisih konvergen antara setiap dokumennya bervariasi, Mulai dari yang signifikan sampai yang tidak dengan jumlah kalimat yang berbeda ataupun sama.

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.