

Peringkasan Multi Dokumen Berbahasa Indonesia Menggunakan Whale Optimization Algorithm

I Made Widiartha¹, Rukmi Sari Hartati², Dewa Made Wiharta³, Nyoman Putra Sastra⁴

¹²³⁴Fakultas Teknik, Universitas Udayana
Jalan Raya Kampus Unud, Jimbaran, Badung, Indonesia

e-mail: madedewidiartha@unud.ac.id¹, rukmisari@unud.ac.id², wiharta@unud.ac.id³,
putra.sastra@unud.ac.id⁴

Received : January, 2025

Accepted : March, 2025

Published : April, 2025

Abstract

Document summarization is a natural language processing method that extracts information from one or more input text documents into an informative and accurate output. Swarm intelligence is one approach that can be used in making summaries that have superior performance. In this study, a multi-document summarization model for Indonesian language was created using one of the swarm intelligence methods, namely Whale Optimization Algorithm (WOA). In making the WOA model for this multi-document summary, 500 Indonesian news documents were used from various online news media. This dataset is divided into two categories, namely 80% for training data and 20% for testing data. In the training stage, WOA is used to optimize the weight of sentence features that are used as a reference in the process of selecting summary sentences. From the results of model testing and validation using k-fold cross validation, it was found that WOA get the best performance value compared to several other methods, namely Rouge-1 = 0.3946, Rouge-2 = 0.1859, Rouge-3 = 0.1309, and Rouge-L = 0.2226. The superiority of the k-fold cross validation results shows the consistency of the model's performance reliability in various document combinations

Keywords: swarm intelligence, whale optimization algorithm, multi documents summarization

Abstrak

Peringkasan dokumen adalah metode pemrosesan bahasa alami yang memadatkan informasi dari satu atau lebih dokumen teks input ke dalam teks keluaran asli dengan hasil yang informatif dan akurat. Swarm intelligence merupakan salah satu pendekatan yang dapat digunakan dalam pembuatan ringkasan yang memiliki kinerja unggul. Penelitian ini bertujuan untuk melakukan pembuatan model peringkasan multi dokumen berbahasa Indonesia menggunakan salah satu metode swarm intelligence, yaitu Whale Optimization Algorithm (WOA). Dalam penelitian ini dimanfaatkan 500 dokumen berita berbahasa Indonesia yang bersumber dari berbagai media berita online. Dataset ini dibagi ke dalam dua kategori yaitu 80% untuk data pelatihan dan 20% untuk data pengujian. Pada tahap pelatihan, WOA digunakan untuk melakukan optimasi bobot fitur kalimat yang dijadikan acuan dalam proses pemilihan kalimat ringkasan. Dari hasil pengujian model dan validasi menggunakan k-fold cross validation didapat bahwa WOA berhasil mendapatkan nilai kinerja terbaik dibandingkan dengan beberapa metode lainnya, yaitu Rouge-1 = 0.3946, Rouge-2 = 0.1859, Rouge-3 = 0.1309, dan Rouge-L = 0.2226. Keunggulan pada hasil k-fold cross validation ini menunjukkan konsistensi kehandalan kinerja model dalam berbagai kombinasi dokumen.

Kata Kunci: swarm intelligence, whale optimization algorithm, peringkasan multi dokumen

1. PENDAHULUAN

Kebutuhan terhadap teknologi peringkasan dokumen secara otomatis saat ini semakin meningkat, hal ini disebabkan oleh lonjakan data tekstual yang sangat pesat dengan hadirnya teknologi digital sebagai media penelusuran dan penyebaran informasi di dunia maya. Penelitian terkait pembuatan teknologi peringkasan dokumen secara otomatis telah banyak dilakukan. Dilihat dari jumlah dokumen yang diringkas, terdapat dua kategori peringkasan, yaitu peringkasan dokumen tunggal dan peringkasan multi dokumen [1]. Pada peringkasan dokumen tunggal, kalimat-kalimat ringkasan bersumber dari sebuah dokumen sedangkan pada peringkasan multi dokumen, kalimat ringkasan dibuat dari beberapa sumber dokumen yang memiliki topik yang sama [2]. Dari sisi hasil ringkasan, peringkasan secara otomatis ini dapat dibagi kedalam dua pendekatan metode, yaitu metode ekstraktif dan abstraktif [3]. Metode ekstraktif melakukan peringkasan dokumen dengan mengambil kalimat-kalimat utama dalam dokumen sumber untuk dirangkai dalam sebuah hasil ringkasan [4], sedangkan peringkasan dengan metode abstraktif dilakukan dengan membentuk kalimat-kalimat baru yang tidak ada dalam dokumen sumber untuk dapat menggambarkan ringkasan dari isi dokumen tersebut [5]. Pada penelitian ini model peringkasan multi dokumen yang dibangun berbasis pada pendekatan ekstraktif. Pendekatan ekstraktif memiliki beberapa kelebihan diantaranya membutuhkan waktu komputasi yang rendah dan ringkasan yang dihasilkan merupakan kalimat yang berasal dari dokumen sumber sehingga tidak mengubah makna dan tata bahasa kalimat sumber [6]. Tantangan utama dalam metode ekstraktif adalah pemilihan kalimat yang memiliki tingkat signifikansi yang tinggi terhadap hasil peringkasan. Terdapat berbagai metode yang dapat digunakan dalam menyeleksi kalimat utama peringkasan salah satunya adalah *Optimization-Based* [7]. Dalam beberapa tahun terakhir, pendekatan berbasis optimasi ini telah menarik banyak perhatian karena kemampuannya memberikan solusi peringkasan dokumen dengan mengoptimalkan berbagai kriteria, khususnya pengurangan redundansi, dan cakupan konten. Metode berbasis optimasi yang memiliki kinerja komputasi yang rendah dan menghasilkan solusi lebih optimal pada

model kombinatorial kompleks adalah metode metaheuristik [8]. Metode metaheuristik merupakan sebuah metode optimasi yang memanfaatkan komponen stokastik untuk mendapatkan titik optimal di setiap iterasinya [9]. Metode ini dikelompokkan ke dalam metode berbasis pada solusi individual yang memiliki dependensi terhadap populasinya [10]. Sebuah individu memanfaatkan hasil pengalaman pribadi dan pengalaman koloni dalam eksploitasi area lokal dan eksplorasi area global pada ruang pencarian. Cara kerja ini membuat metode metaheuristik mampu untuk mendapatkan solusi optimum global [11]. Dalam dua dekade terakhir, metode ini telah menjadi salah satu area penelitian utama sehingga metaheuristik telah dikenal luas sebagai metode yang paling menjanjikan untuk dapat menyelesaikan permasalahan kombinatorial kompleks [12].

Swarm intelligence merupakan metode yang berbasis metaheuristik sebagai suatu solusi untuk mendapatkan pemecahan masalah terdistribusi yang terinspirasi oleh perilaku kehidupan kolektif koloni serangga atau koloni hewan lainnya [13]. Berbagai pendekatan telah dilakukan untuk menentukan kalimat ringkasan tersebut secara tepat, salah satunya adalah pendekatan berbasis optimasi berbasis *swarm intelligence*. Beberapa penelitian tentang peringkasan dokumen menggunakan *swarm intelligence* telah dilakukan, namun permasalahan pemilihan kalimat yang tepat untuk menghasilkan nilai Rouge yang lebih baik masih menjadi isu utama dalam peringkasan dokumen ekstraktif. Metode *Whale Optimization Algorithm* (WOA) yang merupakan salah satu metode *swarm intelligence* yang memiliki kinerja yang unggul. WOA merupakan algoritma optimasi yang mengadopsi perilaku koloni paus bungkuk dalam mencari makanan. Dalam beberapa penelitian, WOA berhasil mengungguli beberapa metode lain dalam berbagai uji fungsi objektif [14][15]. Keunggulan WOA terletak pada mekanisme eksplorasi dan eksploitasinya dalam menemukan titik optimal. Penggunaan WOA dalam pemodelan peringkasan multi dokumen bahasa Indonesia belum pernah dilakukan, sehingga dalam penelitian ini keunggulan WOA digunakan pada model peringkasan multi dokumen bahasa Indonesia. Dalam melakukan optimasi bobot fitur kalimat, WOA memanfaatkan fitur kalimat yang terdiri dari fitur statistik dan fitur linguistik.

2. Penelitian Terkait

Berbagai penelitian peringkasan dokumen menggunakan pendekatan *swarm intelligence* telah dilakukan dalam upaya menghasilkan model peringkasan yang optimal. *Cat Swarm Optimization* (CSO) merupakan salah satu metode *swarm* yang telah dimanfaatkan dalam model peringkasan multi dokumen [16]. CSO merupakan algoritma metaheuristik yang memanfaatkan perilaku kucing dalam mencari makanan. Dalam penelitian ini, fitur yang digunakan dalam pembobotan kalimatnya adalah fitur skor informatif yang didapat dari penjumlahan frekuensi kata yang terdapat dalam kalimat tersebut dan fitur nilai kemiripan kalimat antara kalimat satu dan lainnya dalam kumpulan dokumen yang tersedia. Sebuah penelitian juga telah dilakukan untuk mengembangkan model peringkasan dokumen tunggal dengan memanfaatkan karakteristik kawanan burung Cuckoo dalam mencari tempat potensial untuk dijadikan sarang bertelur [17]. Dua fitur kalimat yang digunakan dalam proses optimasi ringkasan adalah TF-IDF untuk pembobotan kalimat dan juga fitur similaritas antara kalimat dan kata kunci yang dicari. Jumlah kalimat hasil ringkasan, jumlah total kalimat, dan matriks similarity merupakan inputan pengguna yang dibutuhkan dalam model peringkasan ini. Algoritma Cuckoo akan mengoptimasi fungsi tujuan sehingga didapatkan peringkat kalimat yang nantinya akan diambil sejumlah kalimat teratas sesuai dengan jumlah kalimat hasil ringkasan yang telah ditentukan oleh pengguna. Pengembangan domain peringkasan algoritma Cuckoo ini telah dilakukan oleh Rautray dan Balabantaray pada kasus peringkasan multi dokumen [18]. Pada penelitian permasalahan multi dokumen ini digunakan dua fitur, yaitu *term frequency* pada setiap kalimat dan kemiripan antar kalimat dalam sebuah dokumen. Sebuah penelitian berbeda juga telah dilakukan untuk pengembangan model peringkasan dokumen dengan memanfaatkan algoritma *Ant Colony System* (ACS) [19]. Algoritma ACS memanfaatkan pola interaksi koloni semut dalam mencari sumber makanan. Model peringkasan dengan ACS ini diawali dengan pembentukan tiga buah graf, yaitu graf hubungan setiap kata dengan kalimat, graf hubungan antar pasangan kalimat, dan graf hubungan antara setiap pasang kata yang menggunakan pengukuran jumlah urutan

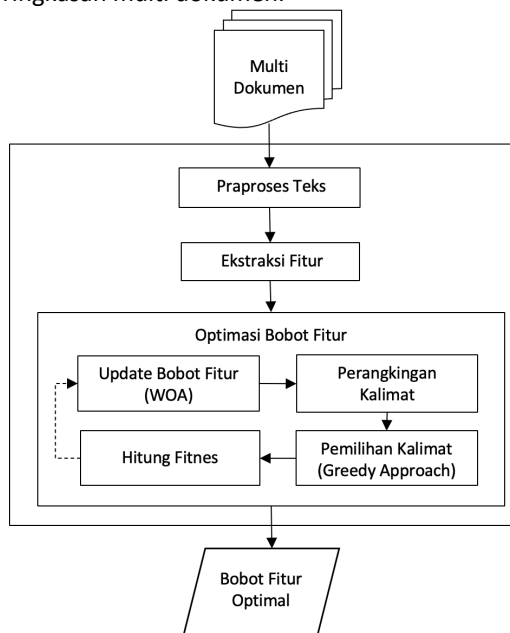
karakter sama yang terpanjang dari setiap pasangannya (*longest common substring*). Hasil dari pembentukan graf ini kemudian diolah untuk menghasilkan bobot untuk setiap kata dan bobot untuk setiap kalimat. ACS melakukan optimasi terhadap kata dan kalimat untuk mendapatkan kalimat-kalimat utama yang menjadi hasil peringkasan dokumen ini.

Pemanfaatan metode *Multi-Objective Artificial Bee Colony* (MOABC) telah dilakukan untuk membangun model peringkasan multi dokumen [20]. Tujuan ganda dari penelitian ini adalah memaksimalkan ruang lingkup sekaligus menekan redundansi kalimat hasil ringkasan dari kumpulan dokumen yang digunakan. Konsep yang diterapkan dalam fungsi tujuannya adalah himpunan solusi *non-dominated* atau lebih dikenal dengan pareto set. Dalam penelitian lanjutan, MOABC ini telah dikembangkan dengan melakukan paralelisasi pada algoritma ABC [21]. Penelitian ini dilakukan dengan melihat kelemahan utama dari peringkasan dokumen otomatis adalah pada waktu proses yang terlampau lama untuk dokumen yang memiliki kata atau kalimat dalam jumlah besar. Paralelisasi ini dilakukan dengan membuat dua skema paralel, yaitu skema *synchronous* dan *asynchronous* pada tiap tahapan inisialisasi, *employed bee*, *onlooker bee*, dan *scout bee*. Dalam skema *synchronous*, proses iterasi akan dieksekusi secara paralel sehingga pada setiap skema *bee* akan dihasilkan solusi yang independen antara *bee* satu dengan lainnya. Pada skema *asynchronous*, lebah yang telah menyelesaikan tugasnya tidak perlu menunggu lebah lain untuk memulai tugas berikutnya. Model ini diharapkan dapat mengoptimasi proses eksplorasi solusi optimal. Penelitian peringkasan dokumen berbahasa Arab yang memanfaatkan metode *Particle Swarm Optimization* (PSO) telah berhasil dilakukan [22]. Untuk menghasilkan ringkasan yang optimal, pada penelitian ini digunakan dua buah fitur, yaitu fitur semantik dan fitur informatif (kemiripan terhadap judul, panjang kalimat, dan posisi kalimat). Kedua fitur ini dibentuk dalam skema graf yang kemudian dilakukan optimasi dengan PSO untuk mendapatkan nilai keseimbangan antara nilai fitur informatif dan keterbacaan sehingga didapatkan peringkat bobot kalimat untuk dapat dipilih sebagai kalimat ringkasan. Penelitian menggunakan PSO juga telah dilakukan oleh Villa-Monte [23]. Dalam penelitian ini digunakan enam kategori fitur,

antara lain posisi kalimat, panjang kalimat, kata kunci, frekuensi (tf-isf), judul, dan ruang lingkup. Setiap kalimat dalam setiap dokumen dikonversi ke dalam vektor numerik dengan dimensi sesuai dengan jumlah fitur yang digunakan. PSO digunakan untuk mengoptimasi bobot tiap fitur yang nantinya akan digunakan dalam perhitungan bobot total kalimat. Hasil pemeringkatan bobot kalimat akan digunakan untuk tahapan pemilihan kalimat sesuai dengan *threshold* yang diinginkan.

3. METODE PENELITIAN

Pengembangan model peringkasan multi dokumen menggunakan WOA berbasis *supervised learning*. Tahapan pelatihan WOA ditujukan untuk mendapatkan bobot fitur kalimat yang optimal. Terdapat beberapa tahapan pelatihan antara lain input data berupa dataset latih, tahap praproses teks, tahap ekstraksi fitur, tahap optimasi bobot hingga menghasilkan bobot optimal untuk setiap fitur kalimat. Bobot hasil pelatihan akan digunakan sebagai koefisien fitur dalam fungsi pembobotan kalimat. Tahap pelatihan model peringkasan multi dokumen menggunakan WOA sesuai dengan Gambar 1. Pada tahapan pengujian, proses pembobotan kalimat dilakukan dengan memanfaatkan bobot fitur optimal hasil pelatihan. Kalimat diperingkatkan berdasarkan bobot kalimat secara *descending* dan dipilih *n* kalimat teratas sebagai hasil ringkasan multi dokumen.



Gambar 1. Tahapan Optimasi Bobot Fitur

3.1 Praproses

Tahapan awal dari peringkasan dokumen ini adalah pengolahan dokumen sumber yang menjadi input dari model peringkasan multi dokumen ini. Tahapan *preprocessing* terdiri dari beberapa tahapan antara lain :

- 1) *Segmentasi Kalimat*: Dokumen-dokumen sumber yang menjadi inputan peringkasan dipisah ke dalam himpunan $D_i = \{S_1, S_2, \dots, S_n\}$. S_j merupakan kalimat ke j dari dokumen ke i . n merupakan jumlah kalimat pada dokumen ke- i .
- 2) *Tokenization*: Tahapan untuk memisahkan setiap kata pada semua kalimat yang dilakukan melalui proses parsing menjadi potongan kata-kata tunggal. Tahapan ini menghasilkan himpunan $T_i = \{t_1, t_2, \dots, t_m\}$. t_j merupakan kata berbeda yang ada dalam dokumen D_i . m merupakan jumlah kata dalam dokumen.
- 3) *Penghapusan Stopword*: Tahapan untuk menghilangkan kata-kata yang tidak memiliki makna penting dalam dokumen. Penghapusan kata-kata ini berdasarkan dari kata-kata yang ada pada kamus *stopword*.
- 4) *Stemming*: Tahapan untuk mereduksi kata-kata pada sebuah dokumen ke dalam bentuk kata dasar dengan cara menghilangkan imbuhan yang mengikuti pola aturan tertentu.

3.2 Ekstraksi Fitur

Setelah dilakukan praproses terhadap seluruh kata dalam dokumen, maka selanjutnya dilakukan tahapan ekstraksi fitur untuk setiap kalimat dalam dokumen sumber. Nilai dari fitur fitur kalimat ini akan diolah dalam sebuah fungsi obyektif untuk mendapatkan bobot dari sebuah kalimat. Pada penelitian ini, terdapat delapan fitur yang digunakan, antara lain:

- 1) *Kemiripan dengan Judul*: Fitur ini merupakan jumlah kata dalam judul yang ada pada kalimat. Semakin banyak kata judul yang ada pada sebuah kalimat maka semakin tinggi bobot kalimat tersebut [24]. Fitur kemiripan dengan judul ini dihitung dengan rumus (1).

$$T_F = \frac{\text{jumlah kata judul dalam kalimat}}{\text{jumlah kata dalam judul}} \quad (1)$$

- 2) *Panjang Kalimat*: Fitur panjang kalimat ini merupakan rasio dari jumlah kata yang ada pada sebuah kalimat dengan jumlah kata yang ada pada kalimat terpanjang pada dokumen. Kalimat pendek tidak signifikan untuk dijadikan hasil peringkasan [25]. Perhitungan untuk mendapatkan nilai fitur ini adalah seperti pada rumus (2).

$$S_L = \frac{\text{jumlah kata dalam kalimat}}{\text{jumlah kata dalam kalimat terpanjang}} \quad (2)$$

3) Posisi Kalimat: Letak dari sebuah kalimat menjadi salah satu pertimbangan dalam peringkasan dokumen. Kalimat pertama pada paragraf memiliki nilai tertinggi [26]. Semakin diawal kalimat tersebut maka semakin besar nilai kecocokannya untuk dijadikan kalimat hasil peringkasan [27]. Misal terdapat 4 kalimat dalam paragraf (berurut dari kalimat ke-1 sampai kalimat ke-4) maka perhitungan nilai setiap kalimatnya adalah seperti pada (3).

$$S_{p1} = \frac{4}{4}, S_{p2} = \frac{3}{4}, S_{p3} = \frac{2}{4}, S_{p4} = \frac{1}{4} \quad (3)$$

4) Kemiripan antar Kalimat: Kemiripan sebuah kalimat s dengan kalimat lainnya dihitung menggunakan konsep *cosine similarity* seperti pada rumus (4). Nilai fitur ini untuk kalimat s didapat dengan cara menghitung rasio dari total nilai kemiripan kalimat s dengan kalimat lainnya dan maksimum total nilai kemiripan pada dokumen.

$$SS_{sim} = \frac{\sum sim(s_i, s_j)}{\max(\sum sim(s_i, s_j))} \quad (4)$$

s_i = kalimat i

s_j = kalimat j

$sim(s_i, s_j)$ = nilai kemiripan dari kalimat ke- i dan ke- j

5) Kata Benda: Perhitungan fitur kata benda ini seperti pada rumus (5)

$$N_p = \frac{\text{jumlah kata benda dalam kalimat}}{\text{panjang kalimat}} \quad (5)$$

6) Kata Tema: Fitur ini mengindikasikan kata yang paling sering muncul pada dokumen. Dalam hal ini ditentukan 10 peringkat teratas kata tema. Semakin banyak kalimat yang mengandung kata tema maka semakin tinggi kemungkinan untuk dijadikan kalimat ringkasan [28]. Perhitungannya dilakukan menggunakan rumus (6).

$$W_T = \frac{\text{jumlah kata tema dalam kalimat}}{\max(\text{jumlah kata tema dalam kalimat keseluruhan})} \quad (6)$$

7) Data Numerik: Kalimat yang mengandung data numerik biasanya memberikan informasi penting dalam sebuah dokumen [28]. Nilai data numerik didapat dari rasio jumlah data numerik pada kalimat dengan panjang kalimat tersebut. Perhitungan fitur ini seperti pada rumus (7).

$$D_N = \frac{\text{jumlah kata numerik dalam kalimat}}{\text{jumlah kata dalam kalimat}} \quad (7)$$

8) TF-IDF: Metode ini terkenal efisien, mudah dan memiliki hasil yang akurat [29]. Metode ini menghitung nilai *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) pada setiap token (kata) di setiap dokumen dalam korpus. Penentuan bobot dari sebuah kalimat ini merupakan total dari bobot setiap kata yang ada

pada kalimat tersebut. Penentuan bobot ini dihitung dengan rumus (8).

$$W_{at} = t f_{at} \times IDF_t \quad (8)$$

3.3 Whale Optimization Algorithm (WOA)

WOA merupakan algoritma optimasi berbasis kecerdasan berkelompok yang terinspirasi dari strategi berburu paus bungkuk dengan menggunakan jaring gelembung [14]. Dalam mencari solusi optimal, WOA memanfaatkan tiga karakteristik perburuan antara lain mengelilingi mangsa (*encircling prey*), metode penyerangan dengan jaring gelembung (*bubble-net attacking method*), and pencarian mangsa (*search for prey*) [30]. Gambaran dari ketiga mekanisme WOA ini adalah sebagai berikut.

1) Mengelilingi/Mengepung Mangsa (*Encircling Prey*):

Pada tahapan ini, paus menggunakan area solusi global optimal sebelumnya untuk dijadikan referensi dalam penentuan area pergerakan memutar untuk mendapatkan mangsa. Seluruh populasi paus akan bergerak secara perlahan menuju target mangsa menggunakan perhitungan seperti pada rumus (9) dan (10).

$$\vec{D} = |\vec{C} \cdot \vec{X}^*(t) - \vec{X}(t)| \quad (9)$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D} \quad (10)$$

Variabel t adalah waktu atau iterasi saat ini, \vec{X} merupakan vektor posisi yang merupakan sebuah solusi, dan \vec{X}^* merupakan solusi terbaik pada saat iterasi ke- t . Variabel A dan C merupakan dua buah variabel kontrol pergerakan paus yang ditentukan melalui rumus (11) dan (12). Variabel a adalah faktor linear dari 2 hingga 0 dan r adalah bilangan acak pada rentang $[0, 1]$.

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \quad (11)$$

$$\vec{C} = 2 \cdot \vec{r} \quad (12)$$

2) Metode Penyerangan dengan Jaring Gelembung (*Bubble-Net Attacking Method*):

Pada tahapan ini, paus akan melakukan penyerangan menggunakan jaring gelembung berbentuk spiral. Perhitungan dari pergerakan ini dapat dilihat pada rumus (13) dan (14). Variabel \vec{D}^l merupakan jarak antara paus saat ini dengan paus yang memiliki nilai solusi terbaik. Variabel l adalah sebuah bilangan random pada rentang $[-1, 1]$ dan b merupakan bilangan konstan yang umumnya ditentukan dengan nilai 1 untuk menentukan bentuk spiral [31].

$$\vec{D}^l = |\vec{X}^*(t) - \vec{X}(t)| \quad (13)$$

$$\vec{X}(t+1) = \vec{D}^l \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \quad (14)$$

Tahap eksploitasi terdiri dari dua strategi yaitu mengepung mangsa dan menyerang dengan jaring gelembung. Penggunaan masing-masing strategi ini memiliki probabilitas 50%. Dengan menggabungkan kedua strategi ini diperoleh acuan pergerakan seperti pada rumus (15).

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D}, & \text{if } p < 0,5 \\ \vec{D}^l \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t), & \text{if } p \geq 0,5 \end{cases} \quad (15)$$

3) Pencarian Mangsa (Search For Prey):

Dalam algoritma WOA, tahapan pencarian mangsa ini merupakan tahapan eksplorasi pada seluruh area ruang pencarian. Model perhitungan tahap ini mirip dengan rumus (9) dan (10). Perbedaan terletak pada penggunaan paus acak sebagai pengganti paus optimal. Perhitungan pergerakan ini dapat dilihat seperti pada rumus (16) dan (17).

$$\vec{D} = |\vec{C} \cdot \vec{X}_{rand} - \vec{X}| \quad (16)$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A} \cdot \vec{D} \quad (17)$$

\vec{X}_{rand} menunjukkan pergerakan paus dalam populasi yang dilakukan secara acak. Penentuan pemilihan metode antara pengepungan (eksploitasi) dan pencarian (eksplorasi) dilakukan dengan memanfaatkan nilai $|A|$. Apabila nilai $|A| < 1$ maka metode yang dipilih adalah eksploitasi sebaliknya eksplorasi akan dipilih saat nilai $|A| \geq 1$. Mekanisme algoritma WOA ini untuk mendapatkan bobot fitur optimal pada permasalahan peringkasan multi dokumen dapat dilihat pada Algoritma 1.

Algoritma 1. Optimasi Bobot Fitur Kalimat

Input : Kalimat (s), Jumlah Kalimat (ns),
Fitur (k), Jumlah Fitur (m), Jumlah
Whale (nw), Maks Iterasi ($tmaks$)

Output : Bobot Fitur Optimal (X)

```

1  X = rand(0,1), i = 1, t = 0, Fbest = 0
2  while (i ≤ nw) do
3      j = 1
4      while (j ≤ ns) do
5          Score sij = ∑a=1m Xa · kija
6          j = j + 1
7      end while
8      resulti = sort_descending(si)
9      summaryi = greedy(resulti)
10     Fi = fitness(summaryi)
11     if (Fi > Fbest) then
12         Xbest = Xi
13     end if
14     i = i + 1
15 end while
16 while (t ≤ tmaks) do
17     // iterasi whale
18     i = 1

```

```

19 while (i ≤ nw) do
20     Perbaharui nilai a, A, C, l, dan p
21     If (p < 0.5)
22         if (|A| < 1)
23             Perbaharui X rumus (10)
24         else if (|A| > 1)
25             Pilih random (Xrand)
26             Perbaharui X rumus (17)
27         end if
28     else
29         Perbaharui X rumus (14)
30     end if
31     Fi = fitness(summaryi)
32     if (Fi > Fbest) then
33         Xbest = Xi
34     end if
35     i = i + 1
36 end while
37 t = t + 1
38 end while

```

4. HASIL DAN PEMBAHASAN

4.1 Deskripsi Data

Dataset yang digunakan dalam penelitian ini berupa dokumen tekstual berita yang bersumber dari media berita *online*. Dataset berita ini didapat melalui metode *scraping* pada berbagai website media berita berbahasa Indonesia. Jumlah dokumen berita yang diambil sebanyak 500 dokumen. Dalam model yang dikembangkan terdapat tahapan optimasi bobot fitur dengan pendekatan *supervised learning* yang membutuhkan data latih berupa kumpulan berita Indonesia beserta referensi ringkasannya. Setiap berita pada dataset telah memiliki topik masing-masing. Sebuah topik memiliki lima buah berita dari lima media berita online yang berbeda sehingga total ringkasan referensi yang dibuat oleh para pakar bahasa sejumlah 100. Hasil ringkasan pakar ini akan dijadikan acuan dalam tahap pelatihan (*training*) dan juga dalam pengukuran kinerja model pada tahap pengujian (*testing*). Dataset dibagi ke dalam dua kategori, yaitu dataset latih sebanyak 80% dan dataset uji sebanyak 20%. Dataset pelatihan digunakan untuk menentukan model formula fungsi bobot kalimat optimal untuk peringkasan multi dokumen.

4.2 Pembahasan

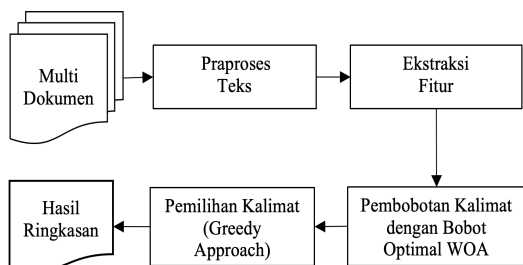
Pada penelitian ini WOA mengoptimasi bobot fitur kalimat yang digunakan untuk menentukan bobot sebuah kalimat. Pemilihan kalimat ringkasan didasarkan pada bobot masing-

masing kalimat. Semakin tinggi bobot maka semakin tinggi peringkat prioritas kalimat untuk dipilih sebagai kalimat ringkasan. Bobot sebuah kalimat didapat dari akumulasi nilai fitur-fitur yang dimiliki kalimat tersebut. Setiap fitur memiliki koefisien (bobot fitur) yang menunjukkan pengaruh fitur tersebut terhadap tingkat kepentingan kalimat. Optimasi bobot ini dilakukan dengan menggunakan pendekatan *supervised learning* yang memanfaatkan 80% data latih, bobot setiap fitur diatur pada rentang $[-1, 1]$. Fungsi fitness yang digunakan pada tahap pelatihan ini adalah matriks Rouge-2. Dari hasil pelatihan model diperoleh hasil bobot fitur optimal seperti pada Tabel 1.

Tabel 1. Bobot Fitur Optimal

No	Fitur	Bobot
1	Kemiripan dengan Judul	0.624065
2	Panjang Kalimat	-0.001561
3	Posisi Kalimat	0.618793
4	Kemiripan antar Kalimat	-0.013958
5	Kata Benda	0.004289
6	Kata Tema	-0.236431
7	Data Numerik	0.221513
8	TF-IDF	-0.613362

Bobot fitur optimal yang telah didapat, digunakan untuk proses perhitungan bobot kalimat. Hasil akhir dari pembobotan kalimat ini berupa daftar peringkat kalimat dari bobot yang terbesar sampai bobot terkecil. Dengan memanfaatkan *greedy selection*, n buah kalimat teratas akan dipilih menjadi kalimat ringkasan. Kinerja model WOA yang dibangun diuji dengan menggunakan empat matriks Rouge, yaitu Rouge-1, Rouge-2, Rouge-3, dan Rouge-L. Proses pengujian dilakukan melalui beberapa tahapan seperti terlihat pada Gambar 2.



Gambar 2. Tahapan Pengujian

Dalam penelitian ini, kualitas ringkasan diukur menggunakan matriks evaluasi ROUGE-N. ROUGE-N mengukur nilai *recall n-gram* antara ringkasan referensi dan ringkasan yang

dihasilkan sistem. ROUGE-N dihitung seperti pada rumus (18)[32].

$$ROUGE - N = \frac{\sum_{S \in \{Referencesummary\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{Referencesummary\}} \sum_{gram_n \in S} Count(gram_n)} \quad (18)$$

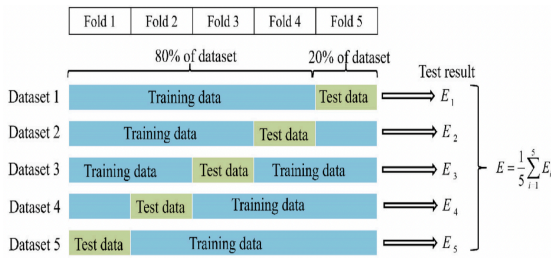
Pada penelitian ini, kinerja model WOA juga dibandingkan dengan dua metode *swarm intelligence* lain diantaranya *Bat Algorithm* (BAT) dan *Invasive Weed Optimization* (IWO). Hasil perbandingan kinerja antara WOA, BAT, dan IWO menggunakan data uji dapat dilihat pada Tabel 2. Dari hasil pengujian terlihat bahwa model WOA unggul pada Rouge-1 dan Rouge-L.

Tabel 2. Hasil Kinerja dengan Matriks Rouge

Metode	Rouge-1	Rouge-2	Rouge-3	Rouge-L
WOA	0.4577	0.2479	0.1913	0.2880
BAT	0,4557	0,2586	0,2013	0,2655
IWO	0.4572	0.2597	0.2022	0.2705

Hasil kinerja model yang didapat dari data uji perlu dilakukan validasi. Tujuan validasi ini adalah untuk mengetahui kinerja model secara umum, memastikan bahwa model tidak hanya bekerja baik pada data tertentu namun juga baik pada berbagai kombinasi dokumen lainnya. Pada penelitian ini, skenario validasi yang digunakan adalah *K-Fold Cross Validation*. Dengan menggunakan *K-Fold Cross Validation*, model dievaluasi pada berbagai subset data, bukan hanya pada satu pembagian data latih dan data uji. Dengan validasi ini dapat diperoleh gambaran yang lebih baik tentang kinerja model dengan karakteristik dokumen yang berbeda-beda. Rata-rata dari hasil-hasil ini memberikan estimasi yang akurat mengenai performa model secara umum.

Dalam proses *K-Fold Cross Validation* dataset dibagi menjadi k bagian data yang sama besar. Model dilatih sebanyak k kali, setiap pelatihan menggunakan satu bagian sebagai data uji dan sisanya sebagai data latih. Hasil dari tiap iterasi kemudian dirata-rata untuk mendapatkan metrik kinerja akhir. Adapun gambaran skema validasi dengan *K-Fold Cross Validation* seperti pada Gambar 3. Pada skema validasi untuk model peringkasan multi dokumen ini, jumlah *fold* yang digunakan sebanyak lima sehingga dataset akan dibagi ke dalam 5 bagian berbeda yang masing-masing diatur sebagai data latih dan data uji.



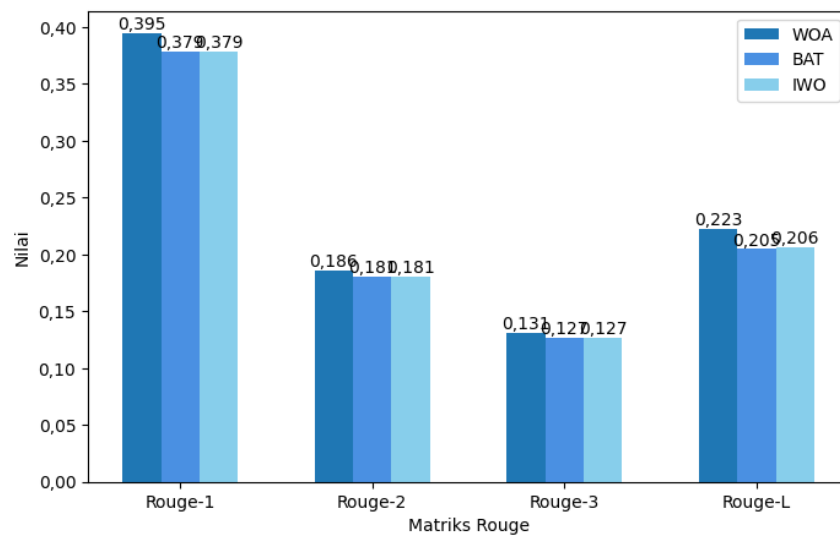
Gambar 3. K-Fold Cross Validation [33]

dapat dilihat pada Tabel 3. Dari validasi yang telah dilakukan, didapatkan hasil bahwa secara umum model WOA untuk peringkasan multi dokumen memiliki kinerja yang lebih baik dari model lainnya. Hal ini terlihat dari hasil pengujian bahwa WOA unggul pada keempat pengukuran kinerja menggunakan matriks Rouge. Secara lebih jelas, perbedaan kinerja dari ketiga model yang dibandingkan dapat dilihat seperti pada Gambar 4.

Hasil pengujian *K-Fold Cross Validation* pada model WOA untuk peringkasan multi dokumen

Tabel 3. Validasi Model dengan *K-Fold Cross Validation*

K-Fold		I (1-20)	II (21-40)	III (41-60)	IV (60-80)	V (81-100)	Rata-rata
Rouge-1	WOA	0,3886	0,3614	0,4011	0,3644	0,4577	0,3946
	BAT	0,3690	0,3373	0,3726	0,3596	0,4557	0,3788
	IWO	0,3690	0,3363	0,3702	0,3623	0,4572	0,3790
Rouge-2	WOA	0,1839	0,1729	0,1930	0,1317	0,2479	0,1859
	BAT	0,1667	0,1549	0,1773	0,1473	0,2586	0,1810
	IWO	0,1667	0,1535	0,1760	0,1470	0,2597	0,1806
Rouge-3	WOA	0,1267	0,1192	0,1363	0,0810	0,1913	0,1309
	BAT	0,1072	0,1064	0,1259	0,0943	0,2013	0,1270
	IWO	0,1072	0,1053	0,1247	0,0942	0,2022	0,1267
Rouge-L	WOA	0,2265	0,1947	0,2126	0,1910	0,2880	0,2226
	BAT	0,2071	0,1916	0,1781	0,1826	0,2655	0,2050
	IWO	0,2073	0,1915	0,1791	0,1822	0,2705	0,2061



Gambar 4. Perbandingan Hasil Validasi Kinerja Model Peringkasan Multi Dokumen

4. KESIMPULAN

Pada penelitian ini telah dibangun sebuah model peringkasan multi dokumen berbahasa indonesia dengan menggunakan metode *Whale Optimization Algorithm* (WOA). Dalam model peringkasan multi dokumen ini WOA berperan

dalam mengoptimasi bobot fitur kalimat yang akan dijadikan acuan dalam menghitung bobot total dari sebuah kalimat. Dari hasil pengujian dan validasi model yang telah dilakukan menggunakan *K-Fold Cross Validation*, didapatkan hasil bahwa WOA telah berhasil

dalam melakukan peringkasan multi dokumen dengan kinerja lebih unggul dibandingkan dengan beberapa metode pembandingan dengan beberapa metode pembandingan lainnya. Acuan kinerja ini dengan menggunakan empat matriks pengukuran nilai Rouge yaitu Rouge-1, Rouge-2, Rouge-3, dan Rouge-L. Dari sisi implikasi praktis, model yang dikembangkan ini dapat dijadikan sebagai acuan dalam pembuatan mesin peringkasan otomatis baik yang dibangun dalam platform web, mobile, ataupun platform aplikasi lainnya. Walaupun model ini telah menunjukkan hasil yang baik dari matriks Rouge, namun model ini belum secara optimal menangani permasalahan redundansi sehingga pada penelitian selanjutnya diharapkan model ini dapat dikembangkan lagi dari sisi penanganan redundansi kalimat ringkasan.

DAFTAR PUSTAKA

- [1] Y. J. Kumar, O. S. Goh, H. Basiron, N. H. Choon, dan P. C. Suppiah, "A review on automatic text summarization approaches," *J. Comput. Sci.*, vol. 12, no. 4, hal. 178–190, 2016, doi: 10.3844/jcssp.2016.178.190.
- [2] N. Zerari, S. Aitouche, M. D. Mouss, dan A. Yaha, "Automatic Text Summarization: A review," *Ninth Int. Conf. Information, Process. Knowl. Manag. (Eknow 2017)*, no. c, hal. 20–25, 2017.
- [3] M. Allahyari *dkk.*, "Text Summarization Techniques: A Brief Survey," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 10, 2017, doi: 10.14569/ijacsa.2017.081052.
- [4] A. Kogilavani dan P. Balasubramani, "Clustering and Feature Specific Sentence Extraction Based Summarization of Multiple Documents," *Int. J. Comput. Sci. Inf. Technol.*, vol. 2, no. 4, hal. 99–111, 2010, doi: 10.5121/ijcsit.2010.2409.
- [5] V. Gupta dan G. S. Lehal, "A Survey of Text Summarization Extractive techniques," *J. Emerg. Technol. Web Intell.*, vol. 2, no. 3, hal. 258–268, 2010, doi: 10.4304/jetwi.2.3.258-268.
- [6] Q. Zhou, F. Wei, dan M. Zhou, "At Which Level Should We Extract? An Empirical Analysis on Extractive Document Summarization," *COLING 2020 - 28th Int. Conf. Comput. Linguist. Proc. Conf.*, hal. 5617–5628, 2020, doi: 10.18653/v1/2020.coling-main.492.
- [7] W. S. El-Kassas, C. R. Salama, A. A. Rafea, dan H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert Syst. Appl.*, vol. 165, hal. 113679, 2021, doi: 10.1016/j.eswa.2020.113679.
- [8] C. L. B. Silveira, A. Tabares, L. T. Faria, dan J. F. Franco, "Mathematical optimization versus Metaheuristic techniques: A performance comparison for reconfiguration of distribution systems," *Electr. Power Syst. Res.*, vol. 196, no. April, 2021, doi: 10.1016/j.epr.2021.107272.
- [9] X.-S. Yang, "Metaheuristic optimization," *Proc. 11th Int. Symp. Oper. Res. Slov. SOR 2011*, no. December, hal. 17–22, 2011, doi: 10.4249/scholarpedia.11472.
- [10] A. Sharma, A. Sharma, S. Choudhary, R. K. Pachauri, A. Shrivastava, dan D. Kumar, "a Review on Artificial Bee Colony and Its Engineering Applications," *J. Crit. Rev.*, vol. 7, no. 11, hal. 4097–4107, 2020, doi: <http://www.jcreview.com/fulltext/197-1596854993.pdf?1597550239>.
- [11] P. Verma dan H. Om, "A novel approach for text summarization using optimal combination of sentence scoring methods," *Sadhana - Acad. Proc. Eng. Sci.*, vol. 44, no. 5, hal. 1–15, 2019, doi: 10.1007/s12046-019-1082-4.
- [12] O. Cordon, F. Herrera, dan T. Stutzle, "A Review on the Ant Colony Optimization Metaheuristic: Basis, Models and New Trends," *Mathw. Soft Comput.*, vol. 9, no. 3, hal. 141–175, 2002, [Daring]. Tersedia pada: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.9.80>.
- [13] E. Bonabeau, M. Dorigo, dan G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, vol. 76, no. 2. 2001.
- [14] S. Mirjalili dan A. Lewis, "The Whale Optimization Algorithm," *Adv. Eng. Softw.*, vol. 95, no. March, hal. 51–67, 2016, doi: 10.1016/j.advengsoft.2016.01.008.
- [15] H. M. Mohammed, S. U. Umar, dan T. A. Rashid, "A Systematic and Meta-Analysis Survey of Whale Optimization Algorithm," *Comput. Intell. Neurosci.*, vol. 2019, 2019, doi: 10.1155/2019/8718571.
- [16] R. Rautray dan R. C. Balabantaray, "Cat swarm optimization based evolutionary framework for multi document summarization," *Phys. A Stat. Mech. its Appl.*, vol. 477, hal. 174–186, 2017, doi: 10.1016/j.physa.2017.02.056.
- [17] S. H. Mirshojaei dan B. Masoomi, "Text

- Summarization Using Cuckoo Search Optimization Algorithm,” *J. Comput. Robot.*, vol. 8, no. 2, hal. 19–24, 2015.
- [18] R. Rautray dan R. C. Balabantaray, “An evolutionary framework for multi document summarization using Cuckoo search approach: MDSCSA,” *Appl. Comput. Informatics*, vol. 14, no. 2, hal. 134–144, 2018, doi: 10.1016/j.aci.2017.05.003.
- [19] A. B. Al-Saleh dan M. El Bachir Menai, “Ant colony system for multi-document summarization,” *COLING 2018 - 27th Int. Conf. Comput. Linguist. Proc.*, hal. 734–744, 2018.
- [20] J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, dan C. J. Pérez, “Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach,” *Knowledge-Based Syst.*, vol. 159, no. June 2020, hal. 1–8, 2018, doi: 10.1016/j.knosys.2017.11.029.
- [21] J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, dan C. J. Pérez, “Parallelizing a multi-objective optimization approach for extractive multi-document text summarization,” *J. Parallel Distrib. Comput.*, vol. 134, no. September, hal. 166–179, 2019, doi: 10.1016/j.jpdc.2019.09.001.
- [22] R. Z. Al-Abdallah dan A. T. Al-Taani, “Arabic Single-Document Text Summarization Using Particle Swarm Optimization Algorithm,” *Procedia Comput. Sci.*, vol. 117, hal. 30–37, 2017, doi: 10.1016/j.procs.2017.10.091.
- [23] A. Villa-Monte, L. Lanzarini, A. F. Bariviera, dan J. A. Olivas, “User-oriented summaries using a PSO based scoring optimization method,” *Entropy*, vol. 21, no. 6, hal. 1–15, 2019, doi: 10.3390/e21060617.
- [24] L. Suanmali, N. Salim, dan M. S. Binwahlan, “Automatic Text Summarization Using Feature-Based Fuzzy Extraction,” *J. Teknol. Mklm.*, vol. 20, no. 2, hal. 105–115, 2008.
- [25] S. Afantenos, V. Karkaletsis, dan P. Stamatopoulos, “Summarization from medical documents: A survey,” *Artif. Intell. Med.*, vol. 33, no. 2, hal. 157–177, 2005, doi: 10.1016/j.artmed.2004.07.017.
- [26] S. A. Babar dan P. D. Patil, “Improving performance of text summarization,” *Procedia Comput. Sci.*, vol. 46, no. Ictict 2014, hal. 354–363, 2015, doi: 10.1016/j.procs.2015.02.031.
- [27] R. P. Reddy, K. Nara, dan S. S. Reddy, “A Comparative Study of Text Summarization Based on Synchronous and Asynchronous PSO,” *Int. J. Adv. Eng. Res. Sci.*, vol. 3, no. 11, hal. 125–130, 2016, doi: 10.22161/ijaers/3.11.22.
- [28] N. Rahman dan B. Borah, “Improvement of query-based text summarization using word sense disambiguation,” *Complex Intell. Syst.*, vol. 6, no. 1, hal. 75–85, 2020, doi: 10.1007/s40747-019-0115-2.
- [29] S. Robertson, “Understanding inverse document frequency: On theoretical arguments for IDF,” *J. Doc.*, vol. 60, no. 5, hal. 503–520, 2004, doi: 10.1108/00220410410560582.
- [30] X. Liang, S. Xu, Y. Liu, dan L. Sun, “A Modified Whale Optimization Algorithm and Its Application in Seismic Inversion Problem,” *Mob. Inf. Syst.*, vol. 2022, 2022, doi: 10.1155/2022/9159130.
- [31] P. Du, W. Cheng, N. Liu, H. Zhang, dan J. Lu, “A modified whale optimization algorithm with single-dimensional swimming for global optimization problems,” *Symmetry (Basel)*, vol. 12, no. 11, hal. 1–23, 2020, doi: 10.3390/sym12111892.
- [32] Chin-Yew Lin, “ROUGE: A Package for Automatic Evaluation of Summaries,” in *Proceedings of the ACL 2004 Workshop on Text Summarization Branches Out, Association for Computational Linguistics*, 2004, hal. 74–81, doi: 10.1253/jcj.34.1213.
- [33] G. Wang dkk., “A fully-automatic semi-supervised deep learning model for difficult airway assessment,” *Heliyon*, vol. 9, no. 5, 2023, doi: 10.1016/j.heliyon.2023.e15629.