

Automatic Text Summarization Menggunakan Metode Graph dan Ant Colony Optimization

I Wayan Adi Setyadi¹, Duman Care Khrisne², I Made Arsa Suyadnya³

Abstract—Information Retrieval (IR) methods is widely used to seek information from documents. Research in the field of IR has been made since 1950s. But in document retrieval, the IR methods use to retrieve document will encounter problems if they have to seek specific information from many document with a lot of text within the document. One of many approach to assist the IR, is to present an automatic summary of documents that can be indexed for sake of document search assist. Automatic Text Summarization in this paper combines two methods, Graph and the Ant Colony Optimization, and combined with the use of four sentences features. The feature is the similarity between sentences, sentences that resembles the title of the document, TF-ISF and TF-IDF. Tests performed to get Cosine Similarity between system's summary with the manual summary from expert. The test show us that system's summary have 78.43% similarity compared with the expert's summary, with compression rate of 78.2%.

Intisari—Metode Information Retrieval (IR) banyak digunakan untuk mencari informasi dari dalam dokumen. Penelitian di bidang IR telah dilakukan sejak tahun 1950an. Tapi dalam pencarian dokumen, metode IR yang digunakan untuk mencari dokumen akan mengalami masalah jika harus mencari informasi spesifik dari banyak dokumen dengan banyak teks di dalam dokumen. Salah satu dari banyak pendekatan untuk membantu IR dalam masalah ini, adalah dengan menyajikan ringkasan dokumen secara otomatis yang dapat diindeks untuk memberi bantuan dalam pencarian dokumen. Peringkasan Teks Otomatis dalam penelitian ini menggabungkan dua metode yaitu metode Graph dan Ant Colony Optimization, yang dikombinasikan dengan penggunaan empat fitur kalimat. Fitur kalimat tersebut adalah, kemiripan antar kalimat, kalimat yang menyerupai judul dokumen, TF-ISF dan TF-IDF. Pengujian dilakukan untuk mendapatkan Cosine Similarity antara ringkasan sistem dengan ringkasan manual dari ahli. Uji coba menunjukkan bahwa ringkasan sistem memiliki kemiripan 78,43% dibandingkan dengan ringkasan ahli, dengan tingkat kompresi 78,2%.

Kata Kunci—Informasi, Information Retrieval, Ant Colony Optimization, Graph, Fitur dokumen.

I. PENDAHULUAN

Informasi yang dianggap *valid* dan bisa dipertanggung jawabkan biasanya dimasukkan dalam bentuk dokumen, artikel, atau jurnal. Karena begitu pentingnya menyebarluaskan informasi sehingga dokumen yang berisi informasi sangat banyak diunggah ke Internet. Banyaknya-

pengguna Internet di Indonesia dan banyaknya situs web yang memberi informasi setiap hari, berdampak pada bertambahnya jumlah artikel atau jurnal yang diposting. Berdasarkan statistik yang dicatat oleh sintal.ristekdikti.go.id, jumlah artikel peneliti Indonesia yang diindeks oleh *google scholar* adalah 349.465 artikel [1]. Dari begitu banyaknya dokumen, tidak semua dokumen tersebut memuat informasi yang diperlukan oleh pembaca atau *user*, didalam dokumen tidak semua teks adalah informasi yang penting. Idealnya dari sekian banyak kalimat pada dokumen hanya beberapa kalimat yang merupakan pokok pikiran penulis. Pokok pikiran ini merupakan informasi penting dalam paragraf yang ingin disampaikan oleh penulis.

Untuk mencari informasi dari dokumen yang banyak, para peneliti di bidang *Information Retrieval* (IR) sudah melakukan penelitian sejak tahun 1950-an. Sistem temu kembali dokumen adalah sistem yang dapat mencari dokumen berdasarkan kata kunci, namun sistem temu kembali dokumen akan menemui kendala jika harus mencari informasi spesifik dari dokumen yang jumlahnya banyak dengan jumlah teks penyusun dokumen yang jumlahnya tidak sedikit [2]. Salah satu pendekatan yang diusulkan untuk membantu *Information Retrieval System* (IRS), adalah dengan menyajikan ringkasan dokumen yang dapat dijadikan indeks saat pencarian dokumen tersebut dilakukan, namun jumlah dokumen yang meningkat dalam Internet tidak diiringi dengan meningkatnya jumlah ringkasan. Maka muncul ide untuk membantu memenuhi jumlah indeks dokumen berupa ringkasan dengan membangun peringkasan teks otomatis atau *Automatic Text Summarization* (ATS) [3]. Sebelumnya penelitian tentang ATS dan IR sudah pernah dilakukan. Beberapa diantaranya adalah penelitian yang dilakukan Aristoteles pada tahun 2013 mengenai penerapan Algoritma Genetika pada peringkasan teks dokumen Bahasa Indonesia yaitu meringkas dokumen bahasa Indonesia yang berjenis file teks dengan menggunakan algoritma genetika. Hasil pengujian menunjukkan bahwa akurasi dengan pemampatan 30%, 20%, 10% sebesar 47.46%, 41.29% dan 35.01% [4], penelitian yang dilakukan oleh Riandayani pada tahun 2013, membahas mengenai peringkasan teks otomatis menggunakan metode *Fuzzy Logic* dan *Fuzzy C-Mean* (FCM) pada dokumen Berbahasa Indonesia yaitu untuk membandingkan hasil ringkasan dari dua metode tersebut. Metode *Fuzzy logic* dan metode *Fuzzy C-Means* akan menggolongkan kalimat kedalam tiga fungsi keanggotaan yaitu: kalimat penting, kalimat rata-rata dan kalimat tidak penting, sehingga didapat kalimat yang benar-benar penting dan mencerminkan isi teks sebagai kalimat penyusun hasil ringkasan [5] dan juga penelitian tentang *Automatic Text Summarization* (ATS) yang dilakukan oleh Pradnyana dan Mogi pada tahun 2014 yang membahas mengenai implementasi *automated text summarization* untuk dokumen tunggal Berbahasa Indonesia dengan menggunakan *Graph-Based Summarization Algorithm* dan Algoritma Genetika

¹Mahasiswa, Program Studi Teknik Elektro Fakultas Teknik, Universitas Udayana, Jln. Jalan Kampus Bukit Jimbaran 80361 INDONESIA (telp: 0361-703315; fax: 0361-4321; e-mail: adiesetyadi@gmail.com)

^{2, 3}Dosen, Jurusan Teknik Elektro dan Komputer Fakultas Teknik Universitas Udayana, Jln. Jalan Kampus Bukit Jimbaran 80361 INDONESIA (telp: 0361-703315; fax: 0361-4321; e-mail: duman@unud.ac.id, arsa.suyadnya@unud.ac.id)

yaitu mengembangkan suatu metode *ATS* berbasis graph (*Graph-Based Summarization Algorithm*) untuk dokumen berbahasa Indonesia yang menggunakan Algoritma Genetika sebagai penyeleksi kalimat yang tidak sesuai dengan ide pokok dari teks. Dari hasil pengujian diperoleh kesimpulan bahwa untuk proses dokumen yang sama, waktu proses Algoritma Genetika dipengaruhi oleh penentuan nilai generasi, populasi dan elitism. Nilai generasi dan populasi yang ditetapkan berbanding lurus dengan waktu proses dan penggunaan sumber daya [6].

Dari penelitian sebelumnya mengenai *ATS*, pada penelitian ini akan diuji hasil *Automatic Text Summarization (ATS)*, dengan menggabungkan metode Graph dan metode *Ant Colony Optimization*. Metode *Ant Colony Optimization (ACO)* dipilih karena dapat akan melakukan analisa pada setiap titik-titik yang digambarkan oleh Graph. Semut akan melakukan rute perjalanan untuk mendapatkan hasil nilai terbaik. Semut terbaik nantinya akan membangun rute perjalanan hingga mendapatkan rute terbaik. Graph dibangun dengan menggunakan bobot *edge*. Untuk mencari bobot dari setiap simpul pada Graph digunakan empat fitur dokumen yaitu : kemiripan antar-kalimat (f1), kalimat yang meyerupai judul dokumen (f2), *TF-ISF* (f3) dan *TF-IDF* (f4). Fitur-fitur ini telah banyak digunakan dalam penelitian sebelumnya [4][5][6][7], dan sudah dibuktikan mampu mengambil fitur dari kalimat. Dengan penggabungan metode ini diharapkan mampu menghasilkan hasil ringkasan yang baik tanpa mengurangi informasi-informasi yang terdapat pada teks asli.

II. LANDASAN TEORI

A. Peringkasan Teks Otomatis

Peringkasan teks otomatis (*automatic text summarization* atau *ATS*) adalah teknik pembuatan ringkasan dari sebuah teks secara otomatis dengan memanfaatkan aplikasi yang dijalankan pada komputer untuk menghasilkan informasi yang paling penting dari dokumen aslinya [8].

B. Metode Graph

Metode graph didefinisikan oleh himpunan verteks dan himpunan sisi (*edge*). Verteks menyatakan entitas-entitas data dan sisi menyatakan keterhubungan antara verteks. Biasanya untuk suatu metode graph G digunakan notasi matematis (1).

$$G = (V, E) \quad (1)$$

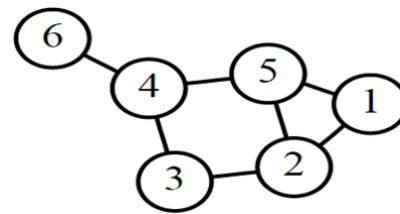
Dimana :

G = Graph

V = Simpul atau Vertex, atau Node, atau Titik

E = Busur atau *Edge*, atau arc

V adalah himpunan verteks dan E himpunan sisi yang terdefinisi antara pasangan - pasangan verteks. Sebuah sisi antara verteks x dan y ditulis $\{x, y\}$. Suatu graph $H = (V1, E1)$ disebut subgraph dari graph G jika $V1$ adalah himpunan bagian dari V dan $E1$ himpunan bagian dari E [9].



Gambar 1: Graf dengan 6 simpul dan 7 sisi

Contoh definisi dari graph pada Gambar 1 adalah : $V = \{1,2,3,4,5,6\}$ dan $E = \{(1,2), (1,5), (2,3), (3,4), (4,5), (5,2), (4,6)\}$. Dari definisi pada Gambar 1, kalimat-kalimat pada setiap paragraf dalam sebuah teks akan dibentuk graph dimana setiap titik simpul pada graph memiliki suatu nilai pada setiap titik simpulnya. Nilai antar titik simpul didapatkan dengan menggunakan perhitungan bobot *edge* (2).

$$Bobot_{i,j} = overlap_{i,j} \times weight_j \quad (2)$$

Dimana :

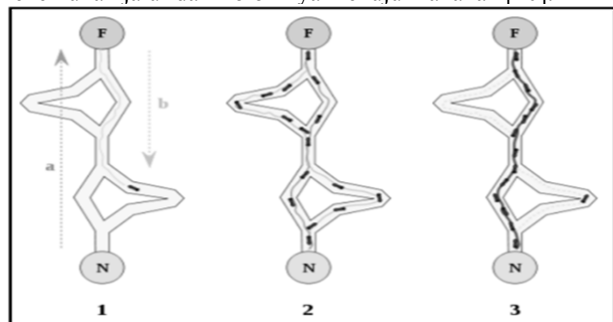
$overlap_{i,j}$ = jumlah kata yang sama antara kalimat ke- i dan kalimat ke- j

$weight_j$ = rata-rata hasil perhitungan dari 4 fitur teks

Nilai $overlap_{i,j}$ diperoleh dengan menghitung jumlah kata yang sama antara kalimat ke- i dan kalimat ke- j dengan mengabaikan *stopword* yang ada di dalam kalimat-kalimat tersebut.

C. Ant Colony Optimization

Algoritma semut atau *ACO* diperkenalkan oleh Moysson dan Manderick dan secara meluas dikembangkan oleh Marco Dorigo, merupakan teknik probabilistik untuk menyelesaikan masalah komputasi dengan menemukan jalur terbaik melalui grafik. Algoritma ini terinspirasi oleh perilaku semut dalam menemukan jalur dari koloninya menuju makanan [10].



Gambar 2: Jalur makanan semut

Pada Gambar 2 merupakan jalur makanan semut dengan F adalah *food source* yang berarti sumber makanan dan N adalah *nest* yang berarti sarang semut. Berikut adalah kronologis bagaimana semut mencari makanannya:

1. Semut pertama mencari sumber makanan melalui jalur manapun dalam hal ini adalah (a). Kemudian kembali ke sarang (N) dengan meninggalkan jejak *pheromone* (b).
2. Semut tanpa pandang bulu akan mengikuti empat kemungkinan, tapi dengan *pheromone* yang lebih kuat,



semut lebih tertarik memilih jalur tersebut sebagai jalur terpendek.

- Akhirnya semut akan mengambil jalur terpendek, dimana jalur lainnya akan semakin kehilangan *pheromone* yang telah menguap.

D. Cosine Similarity

Metode *Cosine Similarity* merupakan metode yang digunakan untuk menghitung *similarity* (tingkat kesamaan) antar dua buah objek. Secara umum penghitungan metode ini didasarkan pada *vector space similarity measure*. Metode *cosine similarity* ini menghitung *similarity* antara dua buah objek (misalkan D1 dan D2) yang dinyatakan dalam dua buah vektor dengan menggunakan *keywords* (kata kunci) dari sebuah dokumen sebagai ukuran. Perhitungan *cosine similarity* yang memperhitungkan perhitungan pembobotan kata pada suatu dokumen dapat dinyatakan dengan persamaan (3) [11] :

$$\text{CosSim}(d_i, q_i) = \frac{q_i \cdot d_i}{|q_i| |d_i|} = \frac{\sum_{j=1}^t (q_{ij} \cdot d_{ij})}{\sqrt{\sum_{j=1}^t (q_{ij})^2} \cdot \sqrt{\sum_{j=1}^t (d_{ij})^2}} \quad (3)$$

Keterangan :

$$q_{ij} = \text{bobot istilah } j \text{ pada dokumen } i = t f_{ij} * i d f_j$$

$$d_{ij} = \text{bobot istilah } j \text{ pada dokumen } i = t f_{ij} * i d f_j$$

Dalam penelitian ini nilai *Cosine Similarity* digunakan sebagai pembandingan keberhasilan peringkasan dokumen, dengan membandingkan kemiripan hasil peringkasan otomatis dengan hasil ringkasan secara manual (*ground truth*).

E. Kemiripan Antar-Kalimat (f1)

Fitur kemiripan antar-kalimat digunakan untuk mencari kata kunci yang sama, yang terdapat antar kalimat dalam paragraf. Nilai kemiripan antar-kalimat (f1) dapat dilihat pada persamaan (4).

$$\text{Score}_{f_1}(s) = \frac{|\text{Keyword Pada } S \cap \text{Keyword antarkalimat}|}{|\text{Keyword Pada } S \cup \text{Keyword antarkalimat}|} \quad (4)$$

Dengan f_1 adalah fitur kemiripan antar kalimat dan s adalah kalimat di dalam dokumen [3].

F. Kalimat yang Menyerupai Judul Dokumen (f2)

Fitur kalimat yang menyerupai judul dokumen adalah nilai kalimat yang diukur dengan menghitung kepemilikan kalimat atas kata yang muncul pada judul dokumen. Kalimat yang menyerupai judul dokumen (f2). Dengan f_2 adalah fitur kalimat yang menyerupai judul dokumen dan s adalah kalimat dalam dokumen, maka skor untuk f_2 dapat dihitung dengan persamaan (5) [3].

$$\text{Score}_{f_2}(s) = \frac{|\text{Keyword dalam } s \cap \text{Keyword dalam judul}|}{|\text{Keyword dalam } s \cup \text{Keyword dalam judul}|} \quad (5)$$

G. TF-IDF (f3)

TF-IDF adalah metode untuk menghitung bobot setiap kata yang paling umum digunakan pada *information retrieval* [12]. Persamaan (6) adalah cara menghitung TF – IDF :

$$W_{dt} = t f_{dt} \times I D F_t \quad (6)$$

Dimana:

d = dokumen ke- d

t = kata ke- t dari kata kunci

W = bobot dokumen ke- d terhadap kata ke- t

$t f$ = banyaknya kata yang dicari pada sebuah dokumen

IDF = *Inversed Document Frequency*

IDF dapat dihitung dengan persamaan (7) :

$$I D F(t) = \log \frac{N}{d f(t)} \quad (7)$$

Dimana $d f(t)$ adalah banyak dokumen yang mengandung *term* t . TF-IDF merupakan kombinasi metode TF dengan metode IDF. Sehingga persamaan TF-IDF adalah(8):

$$T F - I D F(d, t) = T F(d, t) \times I D F(t) \quad (8)$$

H. TF-ISF (f4)

TF-ISF adalah suatu indikator penting atau tidaknya suatu kata dalam merepresentasikan kalimat. Metode ini menggabungkan jumlah kemunculan kata pada tiap kalimat atau *Term Frequency* (TF) dengan banyaknya kalimat dimana suatu kata muncul atau *Sentence Frequency* (SF).

$$w_{ij} = t f_i \times \log \left(\frac{N}{S F_i} \right) \quad (9)$$

Dengan w_{ij} didefinisikan pada persamaan (9) dan $t f_i$ adalah banyaknya kemunculan *term* ke- i pada kalimat. $S F_i$ (*sentences frequency*) merupakan banyak kalimat yang mengandung *term* ke- i dan N adalah banyaknya kalimat dalam satu dokumen [13].

I. Compression Rate

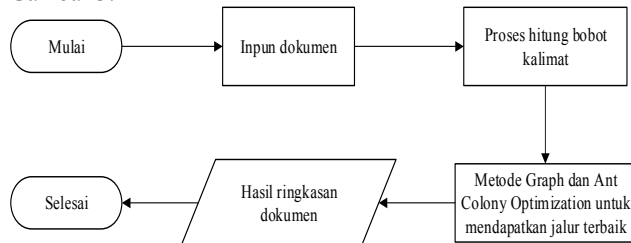
Rasio kompresi (*compression rate*) pada suatu ringkasan berfungsi untuk menentukan persentase batas panjang ringkasan yang akan ditampilkan [14]. Perhitungan *compression rate* dapat dilihat pada persamaan (10).

$$\text{Compression rate} = \frac{P_j. \text{Rngks Sistem}}{\text{Jml. Kt Teks Asli}} \times 100\% \quad (10)$$

III. METODOLOGI PENELITIAN

Pada penelitian ini fitur yang dipilih untuk mewakili kalimat-kalimat dalam dokumen adalah 4 fitur kalimat (f1-f4). Fitur kalimat yang digunakan adalah kemiripan antar-kalimat (f1) fitur ini digunakan untuk mencari kemiripan kata antara kalimat yang satu dengan kalimat lainnya agar hubungan antar kalimat tidak terputus saat diringkas. Idenya adalah sebuah kalimat masih memiliki hubungan dengan kalimat berikutnya dalam paragraf jika terdapat kata kunci yang sama pada kedua kalimat. Fitur berikutnya adalah fitur kalimat yang menyerupai judul dokumen (f2), idenya setiap kalimat yang memiliki kata menyerupai judul kalimat akan memiliki kemungkinan lebih besar untuk menjadi hasil ringkasan. Fitur (f3) digunakan untuk menghasilkan kalimat dengan kata kunci, kalimat dengan banyak kata kunci yang penting (nilai *TF-IDF*

tinggi) dalam penelitian ini berpeluang lebih besar dipilih sebagai hasil ringkasan. Fitur kalimat terakhir adalah *TF-ISF* (f4) yang digunakan untuk memastikan agar kalimat yang penting adalah kalimat dengan jumlah kata penting yang banyak tetapi kalimat yang sama tidak boleh terlalu banyak muncul dalam dokumen. Keempat fitur ini adalah input yang digunakan untuk meringkas dokumen dengan memberi input tersebut pada gabungan dua metode yaitu metode graph dan *ACO*. Skenario peringkasan teks otomatis dapat dilihat pada Gambar 3.



Gambar 3: Alur skenario untuk mendapatkan hasil ringkasan

Secara garis besar skenario peringkasan otomatis akan melalui tahapan sebagai berikut :

1. *Input Dokumen dan Stemming*

Untuk mendapatkan hasil ringkasan *user* akan menginputkan dokumen yang akan di ringkas. Dilanjutkan dengan tahap *stemming* untuk mencari kata dasar dari dokumen. Kata dasar ini akan menjadi bahan untuk proses-proses berikutnya

2. *Perhitungan Bobot Kalimat dan Relasi Antar Kalimat*

Pada penelitian ini metode graph digunakan untuk membuat jalur virtual yang akan dilewati semut pada metode *ACO*. Untuk membangun graph. Sebelum membangun graph 4 fitur teks digunakan untuk mendapatkan perhitungan bobot pada masing-masing kalimat. Fitur teks yang digunakan antara lain : kemiripan antar-kalimat (f1), kalimat yang menyerupai judul dokumen (f2) *TF-ISF* (f3), dan *TF-IDF* (f4). Tabel 1 berisi contoh dokumen yang akan dihitung dengan cara manual sesuai tahapan perhitungan pada penelitian ini.

TABEL 1 CONTOH DOKUMEN

Judul	Tiang Bendera Sekolah
Kalimat	Tiang bendera yang berdiri tegak di pinggir lapangan upacara itu terbuat dari besi. Tinggi tiang itu sekitar 7 meter dan dicat berwarna putih. Di ujung tiang terlihat sang merah putih berkibar. Bendera tersebut baru saja dikibarkan pada upacara Senin pagi.

Kalimat yang di ekstraksi dari dokumen pada Tabel 1 dapat dilihat pada Tabel 2.

TABEL II KALIMAT-KALIMAT DALAM DOKUMEN

Kalimat 1	Tiang bendera yang berdiri tegak di pinggir lapangan upacara itu terbuat dari besi.
Kalimat 2	Tinggi tiang itu sekitar 7 meter dan dicat berwarna putih.
Kalimat 3	Di ujung tiang terlihat sang merah putih berkibar.
Kalimat 4	Bendera tersebut baru saja dikibarkan pada upacara Senin pagi.

a. *Kemiripan antar-kalimat (f1)*

Berdasarkan persamaan (4) maka hasil perhitungan skor f_1 untuk kalimat 1 adalah sebagai berikut :

$$(f1) = \frac{\text{Jmlh kemiripan antar kalimat}}{\text{Jmlh kata unik}} = \frac{3}{20}$$

Pada kalimat 1 terdapat 3 kata yang sama pada kalimat 2, 3, 4 dan 20 kata unik. Sedangkan dengan cara yang sama untuk kalimat 2 memiliki bobot ($\frac{2}{20}$), bobot kalimat 3 ($\frac{2}{20}$) dan bobot kalimat 4 adalah ($\frac{2}{20}$).

b. *Kalimat yang menyerupai judul dokumen (f2)*

Pada contoh dokumen pada Tabel 1 didapatkan perhitungan skor f_2 untuk kalimat 1 adalah 2 kata yang sama dengan judul yaitu kata tiang dan bendera, maka dengan cara yang sama didapatkan skor f_2 untuk kalimat 2, 3 adalah 1 kata yang sama dengan judul yaitu kata tiang dan skor f_2 untuk kalimat 4 adalah 1 kata yang sama dengan judul yaitu kata bendera.

c. *TF-IDF (f3)*

Dengan persamaan (8), perhitungan *TF-IDF* pada Kalimat 1 pada Tabel 2 adalah :

$$\begin{aligned} w_{d,t} &= T_{f_{dt}} \times IDF_t \\ &= 1 \times \log\left(\frac{6}{1}\right) = 0.77815125038364 \end{aligned}$$

Maka hasil perhitungan *TF-IDF* dengan menggunakan empat sampel kalimat yang dipakai adalah Kalimat 1 (9.3378), Kalimat 2 (7.00335), Kalimat 3 (7.00335) dan Kalimat 4 (5.14602).

d. *TF-ISF (f4)*

Dengan menggunakan persamaan (9), perhitungan *TF-ISF* pada Kalimat 1 pada Tabel 2 adalah :

$$\begin{aligned} w_{ij} &= t_{f_i} \times ISF_i \\ &= 1 \times \log\left(\frac{4}{3}\right) = 0.12493 \end{aligned}$$

Hasil perhitungan *TF-ISF* dari empat sampel kalimat yang dipakai adalah : Kalimat 1 (4.58912), Kalimat 2 (3.38503), Kalimat 3 (2.78298) dan Kalimat 4 (3.01023).

Setelah melakukan perhitungan pada masing-masing kalimat dengan fitur teks yang digunakan, maka didapatkan hasil bobot kalimat dari masing-masing kalimat dengan cara menjumlahkan keempat nilai fitur teks lalu dibagi dengan 4.



Maka rata-rata bobot kalimat adalah : Kalimat 1 (4.01923), Kalimat 2 (2.872095), Kalimat 3 (2.7215825) dan Kalimat 4 (2.3140625).

3. Metode Graph Jalur Terbaik Menggunakan Metode Ant Colony Optimization dan Perhitungan Bobot Edge

Pada penelitian ini ringkasan didapat dengan menentukan jalur terbaik menggunakan metode *Ant Colony Optimization*, dari graph yang dihasilkan dengan menghubungkan semua titik (kalimat) dalam dokumen dan memisahkan hubungan tersebut dengan bobot *edge*. Perhitungan untuk mendapatkan nilai bobot *edge* menggunakan persamaan (2). Untuk mendapatkan nilai bobot *edge*, sebagai contoh antara kalimat 1 dan 2 bobot *edge* adalah $Bobot_{1,2} = 2 \times 2.872095 = 5.74419$.

Gambar 4 merupakan suatu graph ABCD yang mewakili rute dari semua kalimat dalam dokumen. Untuk mendapatkan ringkasan harus dicari jalur terbaik, yang bergantung dari nilai bobot antara titik satu ke titik yang lainnya.

Pada penelitian ini untuk mendapatkan nilai terbaik dari perjalanan yang dilakukan oleh algoritma semut yang ditunjukkan pada persamaan (11).

$$\tau \times \text{bobot} = \text{maks} \quad (11)$$

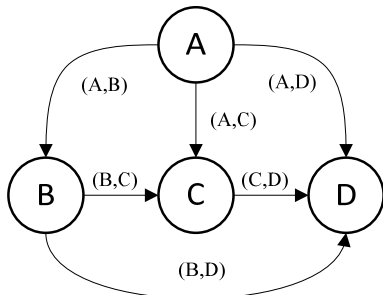
Dimana :

τ = nilai *pheromone*

bobot = nilai bobot antar titik pada perhitungan bobot *edge*

TABEL III HASIL PERHITUNGAN BOBOT EDGE

Perhitungan bobot edge		
Kalimat	Perhitungan	Hasil
Kalimat 1,2	$Bobot_{1,2} = 2 \times 2.872095$	5.74419
Kalimat 1,3	$Bobot_{1,3} = 2 \times 2.7215825$	4.543165
Kalimat 1,4	$Bobot_{1,4} = 3 \times 2.3140625$	6.9421875
Kalimat 2,3	$Bobot_{2,3} = 3 \times 2.7215825$	8.1647475
Kalimat 2,4	$Bobot_{2,4} = 1 \times 2.3140625$	2.3140625
Kalimat 3,4	$Bobot_{3,4} = 1 \times 2.3140625$	2.3140625



Gambar 4: Graph untuk dokumen contoh yang diberikan

Nilai *pheromone* didapatkan dengan iterasi banyaknya semut yang melewati jalur tersebut, setelah dikurangi penguapan *pheromone*. Nilai *pheromone* didapat dengan membuat *roulette wheel* menggunakan jalur yang sudah dibuat. Cara membuat *roulette wheel* dengan membuat juring-juring sebesar nilai bobot dari masing-masing titik. Persamaan (12) digunakan untuk mendapatkan nilai *roulette wheel*.

$$\text{Roulette wheel} = \frac{\text{Nilai bobot edge}}{\text{Semua nilai bobot edge dari titik mulai}} \quad (12)$$

TABEL IV NILAI ROULETTE WHEEL DIMULAI DARI TITIK A

Titik	Roulette wheel	Hasil juring
A ke B	$\frac{5.74419}{17.2295425} = 0.33339$	0 – 0.33
A ke C	$\frac{4.543165}{17.2295425} = 0.26368$	0.33 – 0.59
A ke D	$\frac{6.9421875}{17.2295425} = 0.40292$	0.59 – 0.99

Setelah mendapat *roulette wheel*, dibuat sebuah fungsi untuk memilih secara acak bilangan dari 0 – 1. Bilangan acak yang dihasilkan menentukan kemana semut akan berpindah. Sebagai contoh, pada Tabel 4, misalkan bilangan acak yang dihasilkan 0,4 maka semut akan berpindah dari titik A ke C. Setelah semut berpindah maka akan dibuatkan lagi jalur dari titik C ke titik – titik lainnya dengan langkah membuat *roulette wheel* dan menentukan bilangan acak akan berulang sampai titik yang dipilih semut sampai titik akhir dalam contoh ini adalah titik D. Proses ini akan diulang sampai semua semut telah sampai ke titik akhir.

Ketika semut melalui sebuah jalur maka *pheromone* dari jalur tersebut akan bertambah, ketika seekor semut telah sampai pada titik akhir secara global *pheromone* di seluruh jalur akan diuapkan (dikurangi). Harapnya adalah jalur yang paling bagus adalah jalur yang paling sering dilalui semut sehingga memiliki *pheromone* paling besar.

TABEL V NILAI PHEROMONE

Titik	Nilai pheromone (τ)
1. Titik A ke B	10
2. Titik A ke C	15
3. Titik A ke D	30
4. Titik B ke C	25
5. Titik B ke D	15
6. Titik C ke D	10

Berdasarkan nilai *pheromone* pada Tabel 5, titik A ke D memiliki nilai *pheromone* yang paling besar maka jalur yang dipilih oleh semut adalah jalur dari titik A ke D. Maka hasil ringkasan yang diperoleh adalah sebagai berikut :

TABEL VI CONTOH HASIL RINGKASAN

Kalimat 1	Tiang bendera yang berdiri tegak di pinggir lapangan upacara itu terbuat dari dari besi.
Kalimat 4	Bendera tersebut baru saja dikibarkan pada upacara Senin pagi.

IV. HASIL DAN PEMBAHASAN

Pada Gambar 5 ditampilkan proses pemilihan dokumen, Gambar 6 adalah tampilan dokumen asli yang akan diringkas sedangkan pada Gambar 7 adalah tampilan hasil ringkasan dokumen.

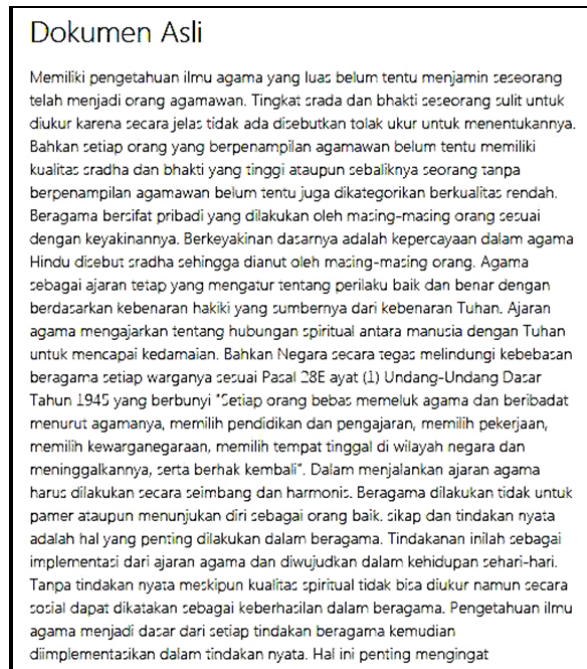


Gambar 5: Tampilan proses pemilihan dokumen

Penelitian yang dilakukan menggunakan 10 dokumen yang dijadikan sampel untuk diringkas oleh aplikasi peringkasan teks otomatis. Pengujian hasil ringkasan dilakukan dengan beberapa cara sebagai berikut ini :

1. *Pengujian Hasil Ringkasan Manual dan Ringkasan Sistem dengan Menggunakan Cosine Similarity*

Hasil ringkasan manual ini dilakukan oleh 10 responden dimana masing-masing akan meringkas setiap dokumen. Kemudian hasil ringkasan manual akan dibandingkan dengan hasil ringkasan dari sistem menggunakan *cosine similarity*. Hasil yang diperoleh setelah membandingkan hasil ringkasan adalah 76.3% yang dapat dilihat pada Tabel 7. Dimana dari hasil perbandingan tersebut informasi yang didapat dari hasil ringkasan sistem 76.3% diantaranya terdapat juga pada hasil ringkasan yang dilakukan secara manual.



Gambar 6: Tampilan dokumen asli yang akan diringkas

Hasil Ringkasan

Memiliki pengetahuan ilmu agama yang luas belum tentu menjamin seseorang telah menjadi orang agamawan. Bahkan Negara secara tegas melindungi kebebasan beragama setiap warganya sesuai Pasal 28E ayat (1) Undang-Undang Dasar Tahun 1945 yang berbunyi "Setiap orang bebas memeluk agama dan beribadat menurut agamanya, memilih pendidikan dan pengajaran, memilih pekerjaan, memilih kewarganegaraan, memilih tempat tinggal di wilayah negara dan meninggalkannya, serta berhak kembali". Tattwa adalah dasar inticari dari ajaran agama yang dalam menjabararkannya disebutkan sebagai panca sradha (lima keyakinan pokok) Widhi sradha menurunkan ajaran Widhi tattwa, attma menurunkan ajaran attma tattwa, karmaphala sradha menurunkan ajaran kharmaphala tattwa, punarbhawa sradha menurunkan ajaran punarbhawa sradha dan moksa sradha menurunkan ajaran moksa tattwa. Dalam pelaksanaan agama ketiga kerangka wajib menjadi acuan dan pedoman dalam menjalankan ajaran agama. Dengan demikian, pengetahuan agama menjadi sangat penting untuk dikuasai dan didalami serta akan menjadi lebih sempurna jika kemudian diimplementasikan dalam kehidupan sehari-hari.

Gambar 7: Tampilan hasil ringkasan dokumen

TABEL VII HASIL PERBANDINGAN RINGKASAN MANUAL DAN SISTEM MENGGUNAKAN *COSINE SIMILARITY*

No.	Judul	Hasil
1.	Beragama Perlu Tindakan, Tak Cukup Hanya Berpengetahuan	64.7%
2.	Guru, Sosok Mulia dalam Mencerdaskan Bangsa	82.2%
3.	Ideologi Pancasila Merajut, Merawat Keberagaman Suku, Budaya, dan Agama di Indonesia	71%
4.	Kesaktian Pancasila Payung Persatuan dalam Keberagaman Nusantara	68.6%
5.	Komodifikasi Kerajinan di Bali	83.8%
6.	Lupakan Sejarah Kelam Bangsa Indonesia	68.8%
7.	Mengapa Perlu Beretika di Media Sosial ?	88%
8.	Pentingnya TIK di Era Globalisasi	76%
9.	Sosial Media Dan Anak Muda Masa Ini	84.1%
10.	Wayang Lemah Masih Kokoh di Bumi Bali	75.8%
Rata – rata <i>cosine similarity</i>		76.3%

2. *Pengujian Hasil Ringkasan Sistem dengan Hasil Ringkasan Auto Summary Tools pada Ms. Word Menggunakan Cosine Similarity*

Pengujian lain yang dilakukan adalah dengan membandingkan hasil ringkasan sistem dengan hasil ringkasan yang didapatkan dari *Microsoft Word* dengan menggunakan *autosummary tools* dan memperoleh hasil ringkasan 68.15%. Setelah melakukan perbandingan dengan *autosummary tools* pada *microsoft word* 68.15% informasi yang yang didapatkan dari hasil ringkasan sistem terdapat juga pada hasil ringkasan yang dilakukan pada *microsoft word*.

3. *Pengujian Hasil Ringkasan Sistem dengan Hasil Ringkasan yang dilakukan oleh Ahli Menggunakan Cosine Similarity*

Pengujian yang dilakukan selanjutnya yaitu dengan membandingkan hasil ringkasan sistem dengan hasil ringkasan yang dilakukan oleh ahli. Ahli yang melakukan ringkasan ini adalah Guru SMA/SMK yang mengajar Bahasa Indonesia. Persentase hasil ringkasan seperti pada Tabel 8 menunjukkan perbandingan hasil ringkasan sistem dengan hasil ringkasan yang dilakukan oleh ahli yang mencapai nilai



rata-rata 79.35%. Hal ini berarti 80% informasi yang dianggap penting oleh pakar sudah dapat di temukan oleh sistem.

4. Compression Rate Hasil Ringkasan Sistem

Untuk mendapatkan *compression rate* digunakan persamaan (10). Tabel 9 merupakan hasil perhitungan *compression rate* panjang ringkasan sistem jika dibandingkan dengan jumlah kata pada teks asli. *Compression ratesistem* adalah sebagai berikut.

$$Cr. Ringkasan = 100\% - 21.80\% = 78.2\%$$

Dari nilai *compression rate*, 78.2% informasi dalam dokumen berhasil diringkas. menyisakan 21.80% informasi yang telah dapat mewakili isi seluruh dokumen.

TABEL VIII HASIL RINGKASAN SISTEM DENGAN HASIL RINGKASAN AHLI MENGGUNAKAN COSINE SIMILARITY

No.	Judul	Hasil
1.	Beragama Perlu Tindakan, Tak Cukup Hanya Berpengetahuan	85.2%
2.	Guru, Sosok Mulia dalam Mencerdaskan Bangsa	86.9%
3.	Ideologi Pancasila Merajut, Merawat Keberagaman Suku, Budaya, dan Agama di Indonesia	82.8%
4.	Kesaktian Pancasila Payung Persatuan dalam Keberagaman Nusantara	65.6%
5.	Komodifikasi Kerajinan di Bali	75.6%
6.	Lupakan Sejarah Kelam Bangsa Indonesia	71.3%
7.	Mengapa Perlu Beretika di Media Sosial ?	80.9%
8.	Pentingnya TIK di Era Globalisasi	85%
9.	Sosial Media Dan Anak Muda Masa Ini	85.3%
10.	Wayang Lemah Masih Kokoh di Bumi Bali	74.9%
Rata – rata cosine similarity		79.35%

TABEL IX HASIL COMPRESSION RATE RINGKASAN SISTEM DAN JUMLAH KATA TEKS ASLI

No. Dkm	Jml. Kt Teks Asli	Pj. Rngks Sistem	Hasil
1	505	145	28.71%
2	454	141	31.05%
3	829	129	15.56%
4	407	57	14%
5	914	125	13.67%
6	590	207	35.08%
7	836	200	23.92%
8	974	107	10.98%
9	650	132	20.30%
10	653	162	24.80%
Rata – rata compression rate			21.80%

V. KESIMPULAN

Berdasarkan hasil dan pembahasan yang dilakukan tentang *Automatic Text Summarization* Menggunakan Metode Graph dan Metode *Ant Colony Optimization* maka dapat disimpulkan beberapa hal sebagai berikut :

1. Rancangan sistem *automatic text summarization* bisa dibangun dengan pendekatan optimasi menggunakan

- metode *ant colony optimization*. Agar metode *ant colony optimization* bisa bekerja kalimat yang akan diringkas ditransformasi menjadi graph yang memiliki bobot pada sisi setiap simpulnya. Nilai bobot ini didapat dengan mengekstrak fitur kalimat dalam dokumen. Dalam penelitian ini fitur kalimat didapat menggunakan fitur f1-f4.
2. Pengujian hasil ringkasan dengan mencari kesamaan hasil ringkasan sistem dengan hasil ringkasan secara manual menggunakan *cosine similarity* memperoleh persentase kesamaan 76.3%. Pengujian dengan *autosummary tools* pada *Microsoft Word* memperoleh hasil ringkasan rata-rata memiliki kesamaan 68.15%. Hasil ringkasan sistem dengan hasil ringkasan ahli memiliki kesamaan 78.43%. Hal ini berarti lebih dari 75% informasi yang dianggap penting oleh manusia sudah dapat di temukan oleh sistem.
3. *Compression rate* ringkasan sistem mencapai 78.2%. Artinya informasi dalam dokumen berhasil diringkas. menyisakan 21.80% informasi yang telah dapat mewakili isi seluruh dokumen.

REFERENSI

- [1] Science and Technology Index. 2017. Kementrian Riset Teknologi dan Pendidikan Tinggi. sintal.ristekdikti.go.id.
- [2] Christopher D. Manning, dkk. 2009. *An Introduction to Information Retrieval*. Cambridge University Press. Cambridge. England
- [3] Fatkhul. A dan Purwatiningtyas. 2015. Rancang Bangun *Information retrieval System (IRS)* Bahasa Jawa Ngoko pada Palintangan Penjebar Semangad dengan Metode *Vector Space Model (VSM)*. Jurnal Teknologi Informasi DINAMIK. Vol. 20, No. 1. pp 25-35.
- [4] Aristoteles. 2013. Penerapan Algoritma Genetika pada Peringkasan Teks Dokumen Bahasa Indonesia. Prosiding Samirata FMIPA Unila, pp. 29-33.
- [5] Riandayani. 2014. Peringkat Teks Otomatis Menggunakan Metode *Fuzzy Logic* dan *Fuzzy C-Mean (FCM)* pada Dokumen Berbahasa Indonesia. JATIT, Vol. 59, No. 3, pp 718-724.
- [6] Pradnyana dan Mogi. 2014. Implementasi *Automated Text Summarization* Untuk Dokumen Tunggal Berbahasa Indonesia Dengan Menggunakan *Graph-Based Summarization Algorithm* Dan Algoritma Genetika. Jurnal Ilmiah NERO , Vol. 1 No. 2, pp. 33-46
- [7] Khrisne, D. C., Rosalin, D. 2014. Peningkatan Penilaian Sentimen pada Komentar Angket Dosen dengan Menggunakan Metode K-Means dan K-Nearest Neighbor Studi Kasus STMIK STIKOM Indonesia. STIKI Applied Science (S@cies), Vol. 4 (2), pp. 57-61.
- [8] Zaman dan Winarko. 2011. Analisis Fitur Kalimat untuk Peringkasan Teks Otomatis pada Bahasa Indonesia. IJCCS, Vol. 5 No. 2, pp. 60-68.
- [9] Samuel Wibisono. 2008. "Matematika Diskrit Edisi 2". Graha Ilmu. Yogyakarta
- [10] Marco Dorigo and Thomas Stutzle. 2004. *Ant Colony Optimization*, A Bradford book. The MIT Press Cambridge, Massachusetts London, England
- [11] G. A. Pradnyana dan N. A. Sanjaya, 2012. "Cosine Similarity", Perancangan Dan Implementasi Automated Document Integration Dengan Menggunakan Algoritma Complete Linkage Agglomerative Hierarchical Clustering, vol. 5, (2), pp. 1-10.
- [12] Robertson, Stephen, Understanding Inverse Document Frequency: On theoretical arguments for IDF, Journal of Documentation, Vol. 60, pp. 502-520
- [13] Xia T, Chai Y. 2011. An improvement to TF-IDF: term distribution based term weight algorithm. Journal of Software. 6 (3): 413-420
- [14] Mustaqhfi, Muchammad. 2011. Peringkasan Teks Otomatis Berbahasa Indonesia Menggunakan Metode Maximum Marginal Relevan. *Skripsi*. Jurusan Teknik Informatika. Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim. Malang.