BAB II

TINJAUAN PUSTAKA

2.1. *Related research*

Penelitian yang dilakukan oleh Abdel Fatah dan Fuji Ren membahas beberapa bentuk model pembobotan pada fitur teks pada peringkasan teks yaitu *mathematical regression* (MR) dan *Genetic Algorithm* (GA) [5]. GA dan MR dipakai untuk menemukan kombinasi yang tepat untuk bobot-bobot dari fitur teks.

Mathematical regression adalah model yang bagus untuk memperkirakan bobot dari fitur teks [13]. Dalam model MR digunakan sebuah fungsi matematika untuk merelasikan antara input dan output. Dalam penelitiannya notasi matriks digunakan sebagai representasi hubungan regresi antara input dan output seperti berikut ini.

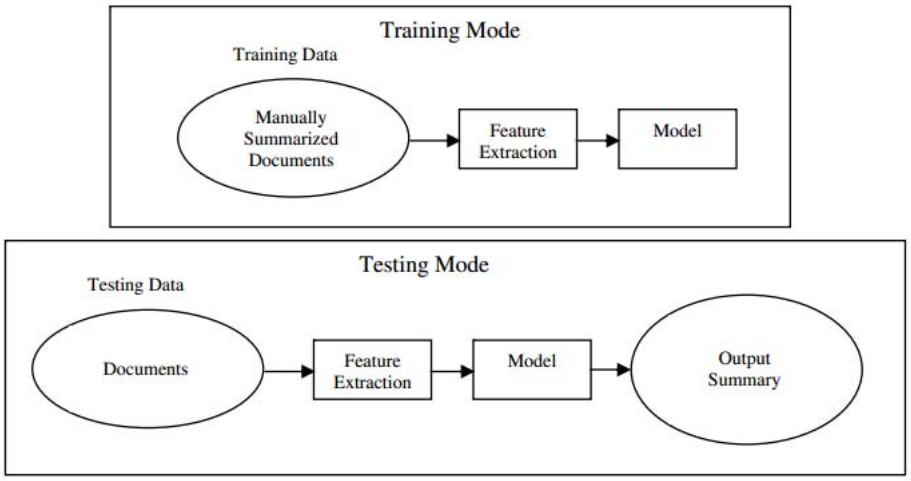
$$\begin{bmatrix} Y0 \\ Y1 \\ \vdots \\ \vdots \\ Ym \end{bmatrix} = \begin{bmatrix} X01 & X02 & X03 & \cdots & X010 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Xm1 & Xm2 & Xm3 & \cdots & Xm10 \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ \vdots \\ w_{10} \end{bmatrix} \tag{2.1}$$

Diasumsikan [Y] adalah vektor output, [X] adalah vektor input dan [W] adalah model statistik linear dari bobot fitur teks.

Sedangkan pada model lain, yaitu GA yang memiliki fungsi tujuan dasar yaitu *optimization*. GA digunakan untuk menentukan secara spesifik bobot dari masing-masing fitur teks. Dari sebuah kalimat (s) dan w_i bobot dari masing-masing fitur teks (f_i), dapat dihitung skornya dengan menggabungkan hasil dari pembobotan dari masing-masing fitur teks.

$$\begin{aligned} Score(s) = {} & w_1 \cdot Score_{f_1}(s) + w_2 \cdot Score_{f_2}(s) + w_3 \cdot Score_{f_3}(s) + w_4 \cdot Score_{f_4}(s) + w_5 \cdot Score_{f_5}(s) \\ & + w_6 \cdot Score_{f_6}(s) + w_7 \cdot Score_{f_7}(s) + w_8 \cdot Score_{f_8}(s) + w_9 \cdot Score_{f_9}(s) + w_{10} \cdot Score_{f_{10}}(s) \end{aligned} \tag{2.2}$$

Pemampatan ringkasan teks dalam penelitian ini dibedakan menjadi tiga kategori, yaitu 10%, 20% dan 30%. Nilai dari *Precision* (P) dapat dilihat pada tabel 2-1.



Gambar 2.1 Model yang diusulkan Abdel Fatah dan Fuji Ren
Sumber: Fattah dan Ren [5]

Tabel 2-1 Nilai *precision* dengan pendekatan GA untuk peringkasan teks bahasa Inggris

Compression rate (CR)	**10%**	**20%**	**30%**
Precission	0.4486	0.4591	0.4563

Penelitian yang dilakukan oleh Turney dan Pantel mengungkapkan bahwa VSM dapat dijadikan alternatif sebagai metode semantik baru [10]. VSM dibangun sebagai metode SMART *Information Retrieval* yang ikut mengawali terlahirnya mesin *search engine* [11]. Keberhasilan implementasi VSM dalam mesin pencari menginspirasi penelitian lain untuk mengembangkan VSM sebagai metode semantik dalam NLP (*Natural Language Processing*).

Berikut ini rangkuman terkait penelitian peringkasan teks di atas.

Tabel 2-2 Rangkuman penelitian terkait peringkasan teks

Nama Peneliti, Judul	Tahun	Masalah	Metode	Hasil
Fattah dan Ren [5], GA, MR, FFNN, PNN and GMM based models for automatic text summarization	2009	Pemilihan konten kalimat yang masih perlu ditingkatkan dengan pemilihan kombinasi bobot	GA, MR, FFNN, PNN dan GMM	Dengan pembobotan pada fitur teks terjadi peningkatan precision untuk pemampatan 10%, 20% 30% sebesar 0.4486, 0.4591, 0.4563
Turney dan Pantel [10], From Frequency to Meaning : Vector Space Models of Semantics	2010	NLP membutuhkan sumber daya yang besar	VSMs	VSMs dapat dikembangkan menjadi metode semantik baru untuk menemukan term dan dokumen yang relatif penting
Suanmali [6], Fuzzy Genetic Semantic Based Text Summarization	2011	GA tidak mampu menangkap hubungan semantik yang terjadi antar kalimat	Integrasi logika fuzzy, GA dan *semantic role labeling*	Integrasi logika fuzzy GA dan *semantic role labeling* dapat meningkatkan precision dari 0.49800 menjadi 0.49958

Pada penelitin ini, peringkasan dioptimalkan untuk dokumen teks bahasa Indonesia. Untuk pemilihan bobot-bobot dari fitur teks digunakan GA. Sedangkan untuk mengetahui hubungan semantik antar kalimat digunakan VSM. Skor kalimat diperoleh dengan menggabungkan skor yang berasal dari fitur teks dan skor dari *Vector Space Model.* Kalimat dengan skor tertinggi akan dipilih sebagai bagian dari ringkasan teks.

2.2. Peringkasan teks

Peringkasan teks adalah sebuah proses untuk mendapatkan intisari dari suatu sumber teks dengan hasil pemampatan kurang dari 50% dari teks asal [3]. Penelitian

peringkasan teks dipelopori oleh Luhn [14]. Beberapa tool yang dapat digunakan untuk peringkasan teks seperti *Microsoft word 2007 summarizer, Copernic Summarizer,* dan *MANYASPECTS summarizer* [6] namun masih terbatas pada bahasa tertentu. Metode dalam peringkasan teks dibagi menjadi dua, yaitu metode abstraksi dan metode ekstraksi [5].

Metode abstraksi adalah mengambil intisari dari sumber dokumen teks kemudian membuat ringkasan dengan menciptakan kalimat-kalimat baru [5] yang merepresentasikan intisari teks asal dalam bentuk berbeda. Metode abstraksi membutuhkan pemrosesan lebih mendalam, khususnya pada *Natural Language Processing* (NLP), *inference* dan *natural language generation* yang sampai saat ini masih belum matang [5]. Sedangkan metode ekstraksi merupakan suatu teknik memilih kalimat atau frase dari teks asli yang memiliki skor tertinggi dan menggabungkannya kembali menjadi ringkasan tanpa mengubah kalimat asal [6].

Berdasarkan jumlah sumber dataset ringkasan, peringkasan dokumen teks dapat dibedakan menjadi dua macam yaitu peringkasan *single-document* dan *multi-document* [15]. Peringkasan *multi-document* mempunyai tugas lebih komplek dibanding peringkasan *single-document*. Hal ini karena informasi-informasi yang masuk akan tumpang tindih antar dokumen yang dapat menyebabkan redundansi dalam hasil ringkasan.

Peringkas Dokumen sangat membantu pembaca untuk mengetahui apakah suatu dokumen memenuhi kebutuhan pengguna dan masih layak untuk dibaca lebih lanjut untuk mengetahui informasi yang dibutuhkan. Kunci utama dari peringkas teks adalah mampu mereduksi besar dokumen teks namun masih mengandung informasi yang relefan dari dokumen asal.

Sebuah dokumen asal yang akan dilakukan peringkasan perlu dilakukan *text pre-processing* yang meliputi *sentence splitting, case folding, tokenizing, stopword removal* dan *stemming* untuk meningkatkan efisiensinya.

2.3. *Text preprocessing*

2.3.1. Sentence Splitting

Sentence splitting adalah proses pemisahan string dokumen teks menjadi kalimat-kalimat. Dalam proses pemecahan dokumen menjadi kalimat dapat menggunakan tanda titik ".", tanda tanya "?" dan tanda seru "!" sebagai delimiter untuk memotong string dokumen.

2.3.2. Case folding

Case folding adalah tahapan proses untuk mengubah semua huruf pada dokumen menjadi huruf-huruf kecil, serta menghilangkan karakter selain huruf a-z dan data numerik.

Contoh:

Input : Sebagai kompensasi, selain mendapat bonus kitab suci Alquran, kapten Timnas Portugal itu juga mendapat bayaran tinggi senilai enam juta dolar AS alias Rp 58 miliar

Output : sebagai kompensasi selain mendapat bonus kitab suci alquran kapten timnas portugal itu juga mendapat bayaran tinggi senilai enam juta dolar as alias rp 58 miliar

2.3.3. Tokenizing

Tokenizing adalah proses pemotongan kalimat yang berasal dari tahapan sebelumnya berdasarkan tiap kata penyusunnya. Perhatikan contoh berikut ini.

"iran uji coba kembali rudal jelajah jangka menengahnya."
Setelah proses *tokenizing* pada kalimat di atas, akan diperoleh delapan kata yaitu: "iran", "uji", "coba", "kembali", "rudal", "jelajah", "jangka", "menengahnya".

2.3.4. Stopword removal

Stopword removal adalah proses penghapusan kata-kata yang termasuk dalam *stoplist*. *Stopword* perlu dilakukan untuk efisiensi pemrosesan pada tahap berikutnya karena keberadaan *stopword* dianggap tidak mempengaruhi makna [16] dari suatu kondokumen teks. Contoh dari kata-kata dalam *stoplist* bahasa Indonesia yaitu: "dan", "seperti", "meskipun", "jika", "andai", "jikalau" dan sebagainya.

Contoh:

Input : Kabar Ronaldo dihadiahi Alquran pun beredar luas

Output : kabar ronaldo dihadiahi alquran beredar luas

2.3.5. Stemming

Stemming merupakan proses mencari akar (*root*) suatu kata dari tiap token kata yang yaitu pengembalian suatu suatu kata berimbuhan ke bentuk dasarnya (*stem*) [17]. Algoritma *stemming* yang dipakai dalam penelitian ini adalah algoritma porter untuk bahasa Indoneisa.

Algoritma stemming ada beberapa macam, diantaranya yaitu:

1. Nazief-Andriani *Stemmer*
2. Porter Stemmer for Bahasa Indonesia
3. CS Stemmer (*Confix Stripping Stemmer*)
4. ECS Stemmer (*Enhanced Confix Stripping Stemmer*)

Langkah-langkah algoritma porter untuk Bahasa Indonesia adalah sebagai berikut.

1. Hapus *Particle* (“-lah”, “-kah”, “-tah” atau “-pun”),
2. Hapus *Possesive Pronoun*.
3. Hapus awalan pertama. Jika tidak ada lanjutkan ke langkah 4a, jika ada cari maka lanjutkan ke langkah 4b.
4. a. Hapus awalan kedua, lanjutkan ke langkah 5a.
   b. Hapus akhiran, jika tidak ditemukan maka kata tersebut diasumsikan sebagai *root word*. Jika ditemukan maka lanjutkan ke langkah 5b.
5. a. Hapus akhiran. Kemudian kata akhir diasumsikan sebagai *root word*
   b. Hapus awalan kedua. Kemudian kata akhir diasumsikan sebagai *root word*

Contoh:

Input : Untuk membiayai hidup sehari-hari, satu persatu lukisan Henk pun dijual

Output : Untuk biaya hidup hari-hari, satu satu lukis Henk pun jual

2.4. Fitur teks

Peringkasan teks dengan metode ekstraksi diperoleh dengan melakukan penskoran terhadap fitur-fitur teks. Dalam penelitian ini, tujuh fitur teks yang dipakai adalah

panjang kalimat (f_1), kalimat yang mengandung data numerik (f_2), kemiripan antar kalimat (f_3), bobot kata (f_4), kata tematik (f_5), posisi kalimat (f_6) dan kalimat yang menyerupai judul (f_7).

2.4.1. Panjang kalimat

Panjang kalimat dapat dihitung skornya berdasarkan jumlah kata unik dalam sebuah kalimat dibagi dengan jumlah kata unik dalam dokumen. Berikut ini ilustrasi dari penerapan penskoran panjang kalimat.

> "*Real Madrid merespon keras pembuatan video kompilasi tentang permainan kasar Pepe terhadap pemain Barcelona. Blaugrana membuat video sepak terjang bek timnas Portugal itu dalam empat laga El Clasico terakhir. Hal itu sebagai respon atas pernyataan Pepe yang menuding pemain El Barca suka sandiwara di lapangan."*

Jumlah kata unik dari paragraf di atas adalah 35 kata. Sedangkan banyak kata unik pada kalimat pertama adalah 12 kata. Sehingga skor fitur teks panjang kalimat untuk kalimat pertama adalah $\frac{12}{35}$.

Berdasarkan ilustrasi di atas, perhitungan skor untuk panjang kalimat dapat dihitung dengan (2.3),

$$Skor\ f_1\ (s)\ =\ \frac{jumlah\ kata\ unik\ dalam\ (s)}{jumlah\ kata\ unik\ dalam\ dokumen} \qquad (2.3)$$

dengan asumsi *s* adalah kalimat dan f_1 adalah fitur teks panjang kalimat.

2.4.2. Kalimat yang mengandung data numerik

Kalimat di dalam dokumen yang mengandung data numerik ikut dipertimbangkan. Karena dalam kalimat tersebut berisi informasi yang ingin disampaikan kepada pembaca. Oleh karena itu, diperlukan penskoran khusus untuk kalimat yang memuat data numerik. Contoh dokumen yang memuat data numerik diantaranya laporan keuangan, laporan investigasi atau laporan inventaris.

Berikut diberikan contoh perhitungan fitur teks kalimat yang mengandung data numerik.

> *Uji coba kali ini dilakukan oleh Which mencakup beberapa jenis smartphone teratas ketika melakukan aktivitas browsing di internet. Hasilnya, tempat pertama diduduki oleh flagship Samsung yaitu Galaxy SIII dengan total waktu mencapai 359 menit. Berikutnya ditempati oleh Sony Xperia S dengan 276 menit, HTC One X dengan 226 menit, Apple iPhone 4S dengan 208 menit, dan diikuti oleh iPhone 5 dengan hanya 200 menit saja. Uji coba ini mencakup 90 jenis ponsel pintar dimana rata-rata ketahanan baterai pada ponsel pintar ketika browsing adalah 193 menit.*

Berdasarkan dokumen di atas, maka skor kalimat pertama adalah nol, karena tidak memiliki data numerik. Kalimat kedua memuat satu data numerik sehingga skornya adalah $\frac{1}{13}$. Kalimat ketiga memuat enam data numerik, maka skor untuk kalimat kedua adalah $\frac{6}{21}$. Kalimat terakhir memuat dua data numerik, maka skor untuk kalimat kedua adalah $\frac{2}{15}$.

Secara matematis, perhitungan skor untuk fitur teks data numerik ditunjukkan pada (2.4) .

$$Skor\ f_2\ (s)\ = \frac{data\ numerik\ dalam\ (s)}{panjang\ kalimat\ (s)} \tag{2.4}$$

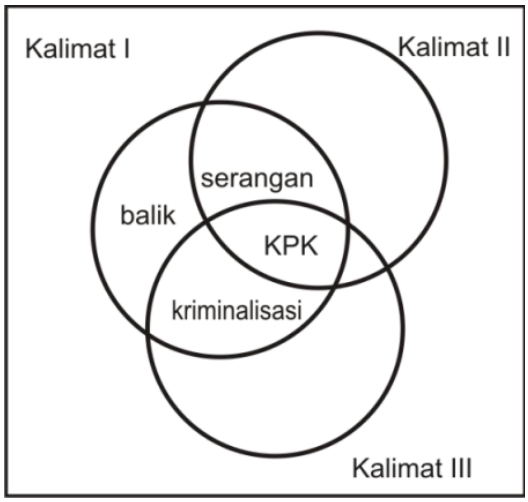
Diasumsikan f_2 sebagai fitur teks data numerik sedangkan *s* sebagai kalimat di dalam dokumen.

2.4.3. Kemiripan antar kalimat

Kemiripan antar kalimat merupakan kata unik yang muncul dalam kalimat sama dengan kata yang muncul dalam kalimat lain. Berikut diberikan contoh penerapan kemiripan antar kalimat.

- KPK mendapat serangan balik dengan kriminalisasi.
- KPK mendapat serangan.
- KPK kembali dikriminalisasi.

Berdasarkan ketiga kalimat di atas, maka perhitungan skor fitur teks kemiripan antar kalimat dari masing-masing kalimat dapat diilustrasikan pada Gambar 2.2. Kalimat pertama memiliki tiga kata yang sama dengan kalimat kedua dan ketiga yaitu kata "**KPK, serangan, kriminalisasi"**. Sehingga skor untuk kalimat pertama adalah $\frac{3}{7}$. Kalimat kedua memiliki dua kata yang sama dengan kalimat pertama dan ketiga, yaitu **"KPK, serangan"**. Sehingga diperoleh skor untuk kalimat kedua adalah $\frac{2}{7}$. Kalimat ketiga memiliki 2 kata yang sama dengan dengan kalimat pertama dan kedua yaitu **"KPK, kriminalisasi"**. Sehingga skor untuk kalimat ketiga adalah $\frac{2}{7}$.



Gambar 2.2 Koneksi antar Kalimat

Secara matematis, perhitungan skor untuk fitur teks kemiripan antar kalimat adalah sebagai berikut.

$$Skor\ f_3\ (s)\ = \frac{keyword\ pada\ \ s \cap keyword\ antar\ kalimat}{keyword\ pada\ \ s \cup keyword\ antar\ kalimat} \quad (2.5)$$

dengan f_3 adalah fitur teks kemiripan antar kalimat sedangkan s adalah kalimat dalam dokumen.

2.4.4. Bobot kata

Bobot kata adalah suatu ukuran yang dipakai untuk menghitung bobot suatu kata (term) yang dinyatakan dengan perkalian skalar antara *Term Frequency* dan *Inverse Document Frequency* [18]. Secara matematis dapat dinyatakan sebagai berikut.

$$w_i = tf \times idf \quad (2.6)$$

Term Frequency didefinisikan sebagai frekuensi munculnya suatu kata (term) dalam kalimat dan dinyatakan dengan tf. Sedangkan frekuensi munculnya kata pada sebuah dokumen dinamakan *Document Frequency* dan dinyatakan dengan df. Untuk menghitung *Inverse Document Frequency* (idf) digunakan persamaan di berikut ini.

$$idf = \log\left(\frac{n}{df}\right) \tag{2.7}$$

Dengan asumsi *n* adalah banyak kalimat.

Skor fitur teks bobot kata dinyatakan dalam rentang [0,1] dan dapat dihitung dengan persamaan (2.8) seperti di bawah ini.

$$Skor\ f_4\ (s_i) = \frac{jumlah\ bobot\ kata\ dalam\ kalimat_i}{Max(jumlah\ bobot)} \tag{2.8}$$

2.4.5. Kata tematik

Dihitung dari jumlah kata tematik pada suatu kalimat, fitur ini penting karena kata yang sering muncul pada dokumen akan lebih sering dikaitkan dengan topik. Yang dimaksud kata tematik di sini adalah kata yg sering muncul. Skor fitur teks f_5 dihitung dengan rumus sebagai berikut.

$$Skor\ f_5\ (S_i) = \frac{jumlah\ kata\ tematik\ dalam\ kalimat}{panjang\ kalimat(jumlah\ kata\ pada\ kalimat)} \tag{2.9}$$

2.4.6. Posisi kalimat

Posisi kalimat adalah letak kalimat pada sebuah paragraf dalam sebuah dokumen teks. Pada penelitian ini, diasumsikan kalimat pertama adalah kalimat yang paling penting. Perhatikan contoh penghitungan fitur teks posisi kalimat berikut ini.

> **"***Masjid merupakan sebuah tempat suci yang tidak asing lagi kedudukannya bagi umat Islam. Masjid selain sebagai pusat ibadah umat Islam, ia pun sebagai lambang kebesaran syiar dakwah Islam. Kaum muslimin pun telah terpanggil untuk bahu-membahu membangun masjid-masjid di setiap daerahnya masing-masing. Hampir tidak dijumpai lagi suatu daerah yang mayoritasnya kaum muslimin kosong dari masjid. Bahkan terlihat renovasi bangunan masjid-masjid semakin diperlebar dan diperindah serta dilengkapi*

dengan berbagai fasilitas, agar dapat menarik dan membuat nyaman jama'ah."

Berdasarkan contoh pada paragraf di atas yang terdiri dari 4 kalimat, maka perhitungan skor fitur teks posisi kalimat untuk kalimat pertama adalah $\frac{4}{4}$, skor untuk kalimat kedua $\frac{3}{4}$, skor untuk kalimat ketiga $\frac{2}{4}$ dan skor untuk kalimat keempat adalah $\frac{1}{4}$. Oleh karena itu, secara matematis untuk menghitung skor fitur teks posisi kalimat dapat dinyatakan sebagai berikut.

$$Skor\ f_6\ (s)\ = \frac{n-(x-1)}{n} \tag{2.10}$$

Andaikan f_6 adalah fitur teks posisi kalimat, s adalah kalimat di dalam paragraf yang akan dihitung skornya, n adalah jumlah kalimat dalam paragraf sedangkan x adalah posisi kalimat dalam paragraf.

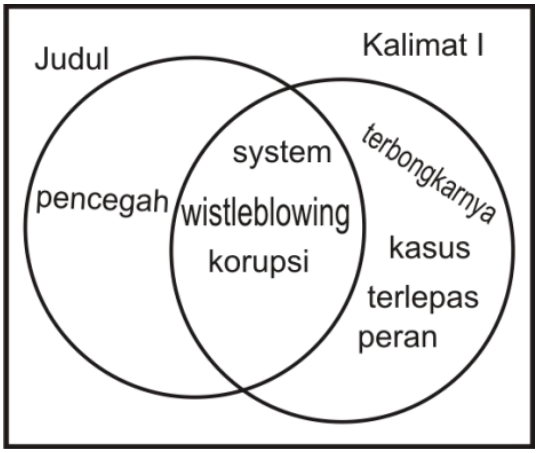
2.4.7. Kalimat yang menyerupai judul

Kalimat yang menyerupai judul adalah kalimat yang memiliki kemiripan kata-kata penyusun dari judul dokumen. Berikut contoh penskoran fitur teks kalimat yang menyerupai judul.

Judul dokumen: ***Whistleblowing System*** **Sebagai Pencegah Korupsi**

- Terbongkarnya kasus korupsi tak terlepas dari peran *whistleblowing system*.
- Ancaman hukuman yang berat dapat mencegah untuk melakukan korupsi.
- Peran masyarakat sangat penting untuk mencegah korupsi.

Berdasarkan contoh di atas, perhitungan skor untuk fitur teks kalimat yang menyerupai judul adalah dapat diilustrasikan seperti gambar 2.2. Kalimat pertama memiliki 3 kata yang mirip dengan judul, yaitu **"*whistleblowing, system,* korupsi"**. Sehingga skor untuk kalimat pertama adalah $\frac{3}{7}$. Kalimat kedua memiliki 2 kata yang mirip dengan judul, yaitu **"mencegah, korupsi"**. Sehingga skor untuk kalimat kedua adalah $\frac{2}{7}$. Kalimat ketiga memiliki 2 kata yang mirip dengan judul, yaitu **"mencegah, korupsi"**. Sehingga skor untuk kalimat ketiga adalah $\frac{2}{7}$.



Gambar 2.3 Ilustrasi Kemiripan Kalimat I dengan Judul

Secara matematis, perhitungan skor fitur teks kalimat yang menyerupai judul ditunjukkan (2.11).

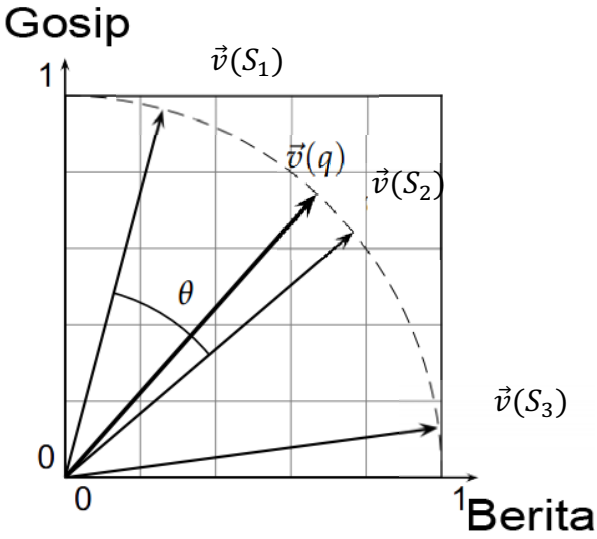
$$Skor\ f_7\ (s)\ =\ \frac{keyword\ pada\ \ s \cap keyword\ dalam\ \ judul}{keyword\ pada\ \ s \cup keyword\ dalam\ \ judul} \qquad (2.11)$$

dimana diasumsikan f_7 sebagai fitur teks kalimat yang menyerupai judul dan s adalah kalimat dalam dokumen.

2.5. Hubungan semantik antar kalimat

Kalimat semantik adalah kalimat yang menyatakan suatu hubungan semantik yang terjadi antar term-term dalam suatu kalimat dengan kalimat lainnya dalam suatu dokumen. Sebuah dokumen yang tersusun atas beberapa kalimat dinyatakan dalam suatu vektor. Kemiripan kalimat semantik dapat dihitung menggunakan metode yang paling umum digunakan yaitu menggunakan *Vector Space Model* (VSM) [18].

Model VSM digunakan untuk mengklasifikasikan kalimat berdasarkan nilai kemiripan semantiknya. Ide dasar dari VSM adalah tiap-tiap term dan kalimat dalam dokumen dinyatakan dalam titik-titik koordinat di dalam ruang vektor. Titik-titik yang berdekatan secara semantik dianggap dekat [10]. Sebaliknya, titik-titik yang berjauhan dianggap memiliki hubungan semantik yang jauh sebagaimana diilustrasikan pada gambar 2.4.



Gambar 2.4 Ilustrasi Cosine Similarity, sim (S_1,S_2) = Cos θ

Terdapat 3 metode pembobotan atau *term weighting* dalam VSM yaitu *Term Frequency* (TF), *Invers Document Frequency* (IDF) dan *Term Frequency Invers Document Frequency* (TFIDF) [11]. Semakin besar bobot TFIDF pada suatu term, semakin penting term tersebut dalam suatu kalimat. Dalam penelitian ini digunakan TFIDF sebagai *term weighting*. Diasumsikan dalam penelitian ini dokumen adalah berupa himpunan kalimat, maka pembobotannya menggunakan persamaan TFISF (*Term Frequency-Inverse Sentence Frequency*) seperti di bawah ini.

$$TFISF = tf \times \log \frac{n}{df} \tag{2.12}$$

Diasumsikan *tf* adalah banyak term dalam satu dokumen, *df* adalah jumlah kalimat yang mengandung term dan *n* adalah banyak kalimat dalam dokumen. Untuk menghitung similaritas antara dua kalimat menurut [11] dapat menggunakan *cosine similarity* antar dua vektor kalimat $\vec{v}(S_1)$ dan $\vec{v}(S_2)$ seperti persamaan 2.13.

$$sim_1(S_1, S_2) = \frac{\vec{v}(S_1) \cdot \vec{v}(S_2)}{|\vec{v}(S_1)| \cdot |\vec{v}(S_2)|} \tag{2.13}$$

Untuk menambah akurasi dalam pengenalan terhadap *abbreviation* maka dipadukan dengan algoritma Monge-Elkan [12], [19], [20]:

$$sim_2(S_1, S_2) = \frac{1}{|S_1|} \sum_{i=i}^{|S_1|} \max_{j=1}^{|S_2|} match(S_1, S_2) \quad (2.14)$$

diasumsikan S_1 dan S_2 adalah kalimat yang dibandingkan.

Skor total untuk hubungan semantik antar kalimat diperoleh dengan persamaan berikut ini [12].

$$sim(S_1, S_2) = a \cdot sim_1(S_1, S_2) + b \cdot sim_2(S_1, S_2) \quad (2.15)$$

dengan $a = 0{,}92$ dan $b = 0{,}08$.

2.6. *Genetic Algorithm* (GA)

Menurut Goldberg dalam Cordon [21], *genetic algorithm* atau GA adalah algoritma pencarian yang didasari pada mekanisme genetik alamiah dan seleksi alamiah. GA dapat diaplikasikan untuk menyelesaikan permasalahan optimasi kombinasi, yaitu dengan mendapatkan suatu nilai solusi optimal terhadap suatu permasalahan yang mempunyai banyak kemungkinan.

Karakteristik dasar GA yaitu:

1. Representasi kromosom untuk memudahkan penemuan solusi dalam masalah pengoptimasian.
2. Inisialisasi populasi.
3. *Fitness function* yang mengevaluasi setiap solusi.
4. Proses genetik yang menghasilkan sebuah populasi baru dari populasi yang ada.
5. Parameter seperti ukuran populasi, peluang proses genetik, dan jumlah generasi.

Sebuah solusi yang dibangkitkan dalam GA disebut sebagai kromosom, sedangkan kumpulan kromosom tersebut disebut sebagai populasi. Sebuah kromosom dibentuk dari komponen-komponen penyusun yang disebut sebagai gen dan nilainya dapat berupa bilangan numerik, biner, simbol ataupun karakter tergantung dari permasalahan yang ingin diselesaikan. Kromosom-kromosom tersebut akan berevolusi secara berkelanjutan yang disebut dengan generasi.

Dalam tiap generasi kromosom-kromosom tersebut dievaluasi tingkat keberhasilan nilai solusinya terhadap masalah yang ingin diselesaikan (fungsi objektif) menggunakan ukuran yang disebut dengan *fitness*. Untuk memilih kromosom yang tetap dipertahankan untuk generasi selanjutnya dilakukan proses yang disebut dengan seleksi. Proses seleksi kromosom menggunakan konsep aturan evolusi Darwin yang telah disebutkan sebelumnya yaitu kromosom yang mempunyai nilai fitness tinggi akan memiliki peluang lebih besar untuk terpilih lagi pada generasi selanjutnya.

Kromosom-kromosom baru yang disebut dengan *offspring*, dibentuk dengan cara melakukan perkawinan antar kromosom-kromosom dalam satu generasi yang disebut sebagai proses crossover. Jumlah kromosom dalam populasi yang mengalami *crossover* ditetukan oleh paramater yang disebut dengan *crossover rate*. Mekanisme perubahan susunan unsur penyusun mahkluk hidup akibat adanya faktor alam yang disebut dengan mutasi direpresentasikan sebagai proses berubahnya satu atau lebih nilai gen dalam chromosome dengan suatu nilai acak. Jumlah gen dalam populasi yang mengalami mutasi ditentukan oleh parameter yang dinamakan *mutation rate*. Setelah beberapa generasi akan dihasilkan kromosom-kromosom yang nilai gen-gennya konvergen ke suatu nilai tertentu yang merupakan solusi terbaik yang dihasilkan oleh GA terhadap permasalahan yang ingin diselesaikan.

Secara umum proses dalam GA adalah sebagai berikut.

a. Membuat populasi awal dari n buah kromosom membentuk suatu gen
b. Menghitung fitness cost dari generasi pertama
c. Melakukan pengulangan proses regenerasi yaitu seleksi (memilih gen terbaik), crossover, mutasi, dan memasukkan gen hasil proses ke dalam generasi baru
d. Memproses generasi baru tersebut untuk proses selanjutnya
e. Evaluasi apakah proses akan diulang
f. Kembali ke langkah b jika nilai fitnest belum tercapai

2.6.1. Variabel-variabel pada GA

Dalam proses GA terdapat variabel-variabel yang akan dibutuhkan dalam pemrosesan GA tersebut. Pemberian inisialisasi pada variabel-variabel tersebut akan

berpengaruh pada hasil GA yang didapat. Variabel-variabel tersebut antara lain sebagai berikut.

1. Jumlah kromosom

Semakin besar jumlah kromosom yang diinputkan dalam variabel jumlah kromosom, semakin besar variasi individu yang dihasilkan. Hal tersebut berpengaruh pada besarnya kesempatan untuk mendapatkan solusi hasil terbaik.

2. Jumlah generasi

Proses looping dalam menentukan berubahnya populasi awal ditentukan oleh jumlah generasi. Variabel ini juga menentukan besarnya kesempatan untuk mendapatkan solusi hasil terbaik.

3. Tingkat Mutasi

Variabel tingkat mutasi menentukan kemungkinan terjadinya mutasi dalam suatu populasi. Misalnya jumlah kromosom 10, variabel tingkat mutasi 0.5, maka setiap kromosom memiliki peluang 0.5 untuk mengalami mutasi.

4. Tingkat Persilangan (*Crossover*)

Sama seperti variabel tingkat mutasi, variabel tingkat persilangan merupakan variabel peluang yang menentukan kemungkinan terjadinya persilangan antara sepasang individu. Hasil akhir dari persilangan akan diperoleh jumlah kromosom dua kali lipat dari jumlah kromosom semula. Dari kromosom-kromosom tersebut akan dipilih sejumlah kromosom dengan *fitness cost* terbaik.

5. Tingkat Reproduksi

Variabel ini mempengaruhi kemungkinan munculnya individu baru di dalam suatu populasi atau yang dinamakan reproduksi.

2.6.2. Langkah-langkah dalam GA

Berikut adalah penjelasan langkah-langkah yang digunakan oleh metode GA secara umum:

1. Pembuatan populasi awal

   Pembuatan populasi awal ini dapat dilakukan melalui pemilihan secara acak dari seluruh kromosom yang ada atau bisa juga dengan menggunakan metode heuristik tertentu atau memberikan input secara manual. Semua populasi dalam

GA berasal dari satu populasi yang disebut populasi awal. Kromosom terbaik dari populasi awal akan dipertahankan dan akan mengalami proses evolusi berikutnya dalam metode GA.

2. Menghitung nilai fitness

   Nilai fitness adalah nilai pembanding untuk menentukan baik tidaknya nilai setiap individu pada populasi. Melalui nilai fitness didapatkan kromosom terbaik dengan cara mengambil nilai fitness yang mempunyai nilai paling tinggi dari kromosom-kromosom yang ada dalam generasi. Perhitungan nilai fitness juga dilihat dari jumlah pelanggaran dalam tiap kromosom. Semakin banyak pelanggaran maka nilai fitness sebuah kromosom menjadi rendah. Kromosom yang baik akan dipertahankan sedangkan kromosom yang kalah bersaing akan diubah lagi untuk mendapatkan kromosom yang baru.

3. Melakukan proses regenerasi

   Proses regenerasi ini bertujuan menciptakan populasi baru dengan melakukan langkah-langkah sebagai berikut.

   a. Seleksi

      Dalam proses seleksi ini akan dipilih dua kromosom parents dari populasi. Parents ini dipilih dari hasil nilai fitness kromosom yang paling baik. Semakin baik nilai fitness-nya berarti semakin tinggi kemungkinan untuk terpilih menjadi parents.

   b. *Crossover* atau pindah silang

      Proses ini dilakukan antara dua kromosom dalam satu populasi dan digunakan suatu probabilitas untuk mengendalikan frekuensi terjadinya *crossover*. Dengan probabilitas *crossover*, penyilangan induk dilakukan untuk membentuk keturunan yang baru. Pemilihan kromosom dan panjang kromosom yang akan disilangkan dipilih secara acak. Setelah *crossover*, akan dilakukan pengecekan untuk menghindari adanya gen yang sama.

   c. Mutasi

      Mutasi merupakan cara lain untuk membuat variasi populasi. Sama seperti *crossover*, mutasi juga mempunyai probabilitas mutasi untuk menentukan

tingkat mutasi yang terjadi. Mutasi adalah proses memilih dua gen secara acak dalam suatu kromosom untuk ditukarkan posisinya.

d. *Accepting*

Menempatkan keturunan (*offspring*) yang baru di dalam populasi baru.

4. Memproses generasi baru untuk proses berikutnya

Menggunakan generasi populasi yang baru untuk digunakan dalam proses pengulangan algoritma berikutnya.

5. Evaluasi perlu tidaknya pengulangan proses

Proses akan mengalami evaluasi untuk melihat apakah kondisi akhir sudah memenuhi syarat. Jika sudah, proses akan dihentikan dan memakai solusi terbaik dalam populasi tersebut. Jika belum, maka dilakukan pengulangan proses.

2.7. Penerapan GA dan VSM untuk peringkasan teks

Menurut Suanmali [6] metode ekstraksi dengan pembobotan GA hanya mampu mendapatkan konten utama dari suatu dokumen teks namun tidak dapat menangkap hubungan semantik yang terjadi antar kalimat. Oleh karena itu, VSM akan ditambahkan sedemikian hingga integrasi dari keduanya mampu mendapatkan konten utama dari suatu dokumen teks dan dapat menangkap hubungan semantik yang terjadi antar kalimat.

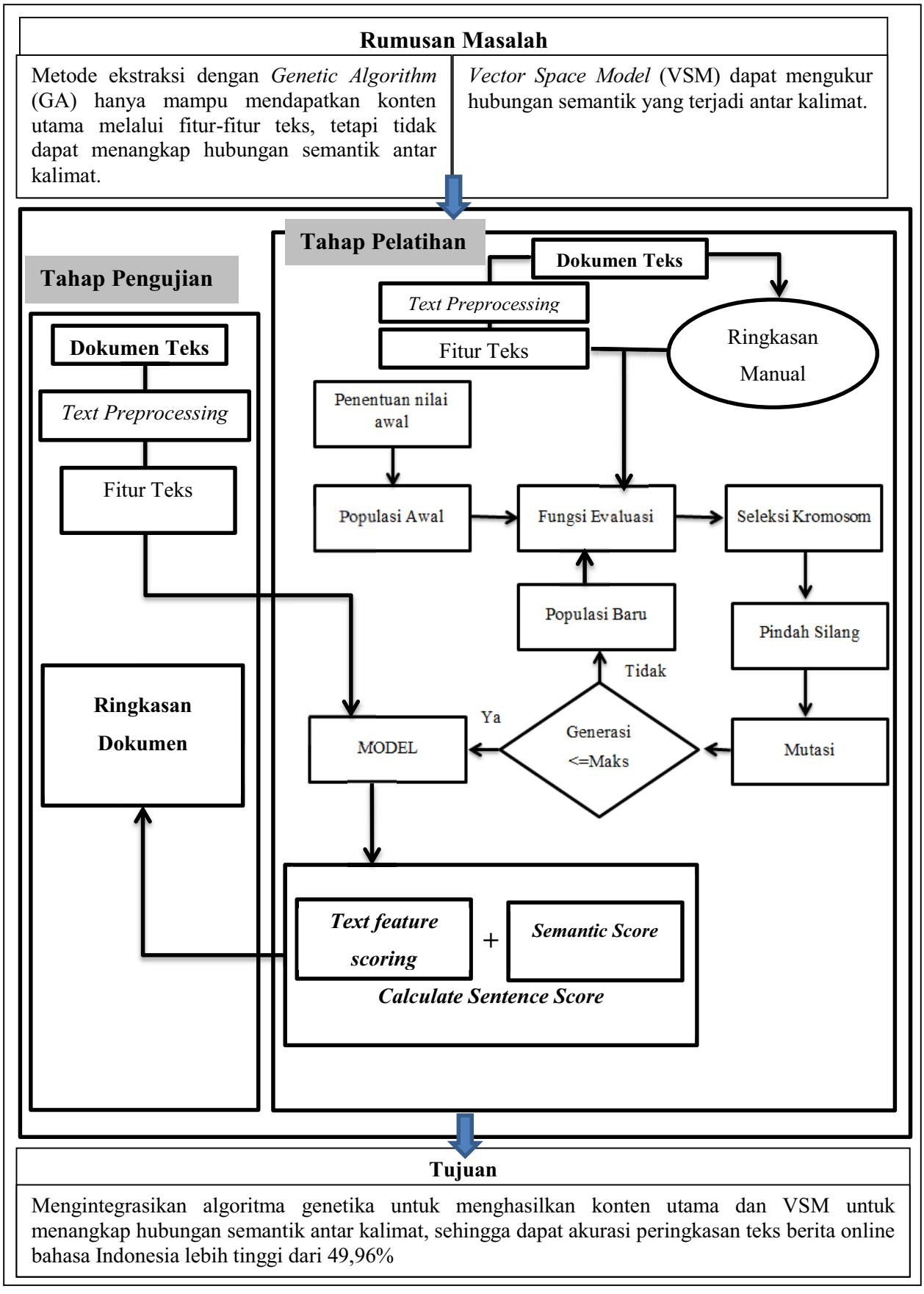
Data awal dalam penelitian ini adalah dokumen teks berupa artikel berita online dalam bahasa Indonesia yang dikumpulkan secara acak dari portal berita online www.kompas.com dan www.republika.co.id mulai dari bulan Nopember 2012 sampai Maret 2013. Sebanyak 40 buah artikel tersebut di atas akan digunakan dalam penelitian ini sekaligus diringkas secara manual. Dimana 20 buah dokumen akan digunakan sebagai data pelatihan dan sisanya akan digunakan sebagai data uji.

Sebelum pembobotan fitur teks dilakukan terlebih dahulu dilakukan *text preprocessing* terhadap data training yang meliputi *sentence splitting, case folding, tokenizing, stopword removal dan stemmer*. Pembobotan fitur teks dilakukan dengan menguji 20 data training dengan GA. Fitur teks yang dipakai adalah panjang kalimat (f_1), kalimat yang mengandung data numerik (f_2), kemiripan antar kalimat (f_3), bobot kata (f_4), kata tematik (f_5), posisi kalimat (f_6) dan kalimat yang menyerupai judul (f_7).

Hasil pembobotan fitur teks adalah kombinasi bobot-botot dari ketujuh fitur teks. Pembobotan ini hanya dilakukan pada tahap pelatihan.

Pada tahap selanjutnya akan dilakukan pengujian terhadap data *testing* dimana penskoran fitur teks dilakukan dengan memakai bobot-bobot fitur teks yang diperoleh pada tahap pelatihan. Hasil ringkasan pada tahap ini akan dibandingkan dengan ringkasan manual dengan menghitung nilai presisi. Sehingga akan diketahui peningkatan akurasinya.

Secara umum metode penelitian yang dilakukan mengacu pada kerangka pemikiran seperti tampak gambar 2.5.



Gambar 2.5 Kerangka Pemikiran