BAB II

LANDASAN TEORI

2.1 Peringkasan Teks Otomatis *(Automatic Text Summarization)*

Peringkasan Teks Otomatis (*Automatic Text Summarization)* merupakan pembuatan rangkuman dari sebuah sumber teks secara otomatis dengan menggunakan serta memanfaatkan sistem peringkasan teks yang dijalankan pada komputer. Sebuah sistem peringkasan teks diberi *input* (masukan) berupa teks, kemudian sistem melakukan *process* (proses) peringkasan, dan sistem menghasilkan *output* (keluaran) berupa teks yang lebih singkat dari sumber teks aslinya. Hasil peringkasan mengandung poin-poin penting atau informasi utama dari teks sumber asli [1].

2.1.1 Karateristik Peringkasan Teks Otomatis

Pada peringkasan teks otomatis terdapat dua pendekatan, tipe ringkasan berdasar teknik pembuatan, suatu ringkasan atau rangkuman diambil dari bagian terpenting dari teks sumber asli [2]. Terdapat 2 tipe ringkasan yaitu :

1. Abstraktif

   Tipe peringkasan abstraktif menghasilkan sebuah interpretasi terhadap teks aslinya. Dimana sebuah kalimat akan ditransformasikan menjadi kalimat yang lebih singkat dan menciptakan kalimat baru yang tidak terdapat dalam dokumen teks aslinya.

2. Ekstraktif

   Tipe peringkasan ekstraktif menghasilkan suatu ringkasan dengan memilih sebagian dari kalimat yang ada dalam dokumen teks aslinya.

Berdasarkan jumlah sumbernya, sebuah ringkasan dapat dihasilkan dari satu sumber *(single document)* atau dari banyak sumber *(multi document).* Pada Peringkasan *single document* masukannya (*input*) yaitu berupa sebuah teks dan keluarannya (*output*) berupa sebuah teks yang lebih singkat dari teks sumber aslinya. Sedangkan Pada peringkasan *multi document*, masukannya yaitu berupa beberapa dokumen teks yang memiliki satu jenis tema yang sama, biasanya sudah ada dalam satu kelompok, lalu akan menghasilkan keluaran berupa teks yang lebih singkat yang merangkum informasi-informasi utama pada klaster masukan [3].

Compression rate pada proses peringkasan teks otomatis akan menentukan panjang ringkasan yang dihasilkan. Biasanya diukur berdasarkan persentase dari teks sumber, misalnya hasil rangkuman ingin memiliki panjang ringkasan 25%, 50% atau 70 % dari teks sumber dan dapat pula diukur berdasarkan jumlah kata dari teks sumber asli, misalnya ditentukan ringkasan sepanjang 200 kata [4].

2.2 Artikel

Artikel yaitu sebuah karya tulis secara lengkap, contoh seperti esai di majalah atau laporan berita, surat kabar dan lain sebagainya. Artikel adalah sebuah tulisan lepas yang berisikan pendapat atau opini seseorang yang mengupas tuntas tentang sebuah masalah yang bersifat aktual dan biasanya kontroversial dengan

tujuan untuk mempengaruhi, memberitahu, meyakinkan & menghibur para pembaca [5].

2.2.1 Ciri-ciri Artikel

Artikel memiliki ciri-ciri sebagai berikut [15] :

1. bersifat Faktual. Artikel adalah tulisan nonfiksi atau berdasarkan fakta dan data. Tema atau masalah yang dibahas benar- benar ada atau terjadi bukan khayalan.
2. berisi gagasan dan fakta. Artikel berisikan pendapat yang dilengkapi fakta, peristiwa, atau masalah.
3. meyakinkan. Sebuah artikel dapat menjadi sarana penulisannya guna meyakinkan orang lain (pembaca) akan pentingnya suatu masalah dipikirkan atau disikapi.
4. mendidik. Artikel umumnya mendidik dan mengajarkan sesuatu agar pembaca melakukan atau tidak melakukan sesuatu.
5. memecahkan masalah. Artikel membahas suatu masalah yang disertai alternatif pemecahannya atau jalan keluar.
6. menghibur. Sebuah artikel bisa pula menghibur pembacanya dengan mengangkat tema yang ringan dan lucu.

2.2.2 Jenis-jenis Artikel

Jenis-jenis artikel yaitu sebagai berikut [16] :

1. Artikel deskriptif adalah tulisan yang isinya menggambarkan secara detail ataupun garis besar tentang suatu masalah, sehingga pembaca mengetahui secara umum utuh suatu masalah yang dikemukakan.

2. Artikel eksplanatif adalah artikel yang isinya menerangkan sejelas-jelasnya tentang suatu masalah, sehingga si pembaca memahami betul masalah yang dikemukakan.

3. Artikel prediktif berisi ramalan atau dugaan apa yang kemungkinan terjadi pada masa yang akan datang.

4. Artikel preskriptif isinya mengandung ajakan, imbauan atau perintah bagi pembaca agar melakukan sesuatu.

2.2.3 Kata

Kata yaitu unsur bahasa yang diucapkan atau dituliskan yang merupakan perwujudan kesatuan perasaan dan pikiran yang dapat digunakan dalam berbahasa.

Kategori bentuk kata terdiri dari lima kata yaitu [6] :

1. Kata Benda (nomina)

   Kata benda adalah kata yang mengacu pada manusia, binatang, benda dan konsep atau pengertian.

2. Kata Kerja (verba)

   Kata kerja adalah kata yang menyatakan tindakan.

3. Kata Sifat (adjektiva)

   Kata sifat adalah kata yang memberi keterangan yang lebih khusus tentang sesuatu yang dinyatakan oleh nomina dalam suatu kalimat.

4. Kata Keterangan (adverbia)

   Kata keterangan adalah kategori yang dapat mendampingi adjektiva, preposisi dalam bentuk kata.

5. Kata Tugas

   Kata tugas adalah kata yang hanya memiliki arti gramatikal dan tidak memiliki arti leksikal.

2.2.4 Kalimat

Kalimat yaitu sebagai satuan bahasa yang secara relatif berdiri sendiri yang memiliki pola intonasi final, dan secara aktual maupun potensial terdiri dari klausa, klausa bebas yang menjadi bagian kognitif percakapan, satuan proposisi yang merupakan gabungan klausa atau merupakan satu klausa, yang membentuk satuan bebas, jawaban minimal, seruan, salam, dan sebagainya [7].

Unsur-unsur kalimat terdiri dari kata, kelompok kata dan lagu kalimat. Di dalam kalimat terdapat pengaturan hubungan kedudukan antara bagian-bagiannya. Ada bagian didalam kalimat yang menunjukkan sebagai “pelaku”, ada bagian yang menunjukkan sebagai “perbuatan”. Berdasarkan jabatannya kalimat terdiri dari :

a. Subyek, yaitu bagian yang menjadi pokok pembicaraan.
b. Predikat, yaitu bagian yang menerangkan subyek

c. Obyek, yaitu bagian yang menjadi tujuan.

d. Keterangan, yaitu bagian yang menunjukkan waktu (keterangan waktu), tempat (keterangan tempat), alat (keterangan alat) dan sebagainya.

Sedangkan kalimat berdasarkan fungsinya, dapat dikategorikan sebagai berikut:

1. Kalimat pernyataan
2. Kalimat pertanyaan
3. Kalimat perintah
4. Kalimat seruan

2.2.5 Paragraf

Paragraf adalah seperangkat kalimat yang membicarakan suatu gagasan atau topik. Kalimat dalam paragraf memperlihatkan kesatuan pikiran atau mempunyai keterhubungan dalam membentuk suatu gagasan atau topik tersebut [8].

Terdapat dua syarat dalam membentuk paragraf :

1. Menulis pernyataan (kalimat) tentang pokok bahasan dengan baik.
2. Mengangkat pola susunan rincian dengan baik.

2.3 *Text Mining*

Text Mining (penambangan teks) adalah proses ekstraksi pola (informasi dan pengetahuan) dari sejumlah sumber data besar yang tidak atau kurang terstruktur. *Text mining* memiliki tujuan dan menggunakan proses yang sama dengan *Data Mining* (penambangan data), namun memiliki masukan (*input*) yang

berbeda. Masukan untuk *text mining* yaitu berupa data yang tidak atau kurang terstruktur, seperti dokumen PDF, kutipan teks, Word, dll. Sedangkan untuk masukan *data mining* yaitu berupa data yang terstruktur [9].

Area penerapan *text mining yang* paling populer adalah:

1. Ekstraksi informasi (*information extraction*) merupakan Identifikasi frasa kunci dan keterhubungan atau keterkaitan satu sama lainnya di dalam teks dengan melihat urutan tertentu melalui pencocokan pola.
2. Pelacakan topik (*topic tracking*) yaitu Penentuan dokumen lain yang menarik seorang *user* berdasarkan profil dan dokumen yang dilihat dari *user*.
3. Perangkuman (*summarization*) adalah Pembuatan rangkuman dokumen untuk mendapat intisari dari dokumen aslinya.
4. Kategorisasi (*categorization*) merupakan Penentuan tema utama suatu teks dan pengelompokan teks berdasarkan tema tersebut dan dimasukkan ke dalam kategori yang telah ditentukan.
5. Penggugusan (*clustering*) yaitu Pengelompokan dokumen yang serupa tanpa penentuan kategori sebelumnya (tidak sama dengan *categorization* di atas).
6. Penautan konsep (*concept linking*) adalah Penautan dokumen berhubungan dengan identifikasi konsep yang dimiliki bersama, sehingga membantu *user* untuk menemukan dan mendapatkan informasi yang mungkin tidak akan ditemukan dengan hanya menggunakan metode pencarian tradisional.

7. Penjawaban pertanyaan (*question answering*) yaitu Pemberian jawaban terbaik terhadap suatu pertanyaan dengan mengggunakan pencocokan pola berdasarkan pengetahuan.

Dengan menggunakan *text mining* tugas-tugas yang berhubungan dengan penganalisaan teks dengan jumlah yang besar, penemuan pola dan penggalian informasi yang berguna dari suatu teks dapat dilakukan. Proses *text mining* dibagi menjadi 3 tahap utama, yaitu proses awal terhadap teks (*text preprocessing*), transformasi teks (*text transformation*), dan *pattern discovery/analysis* [10].

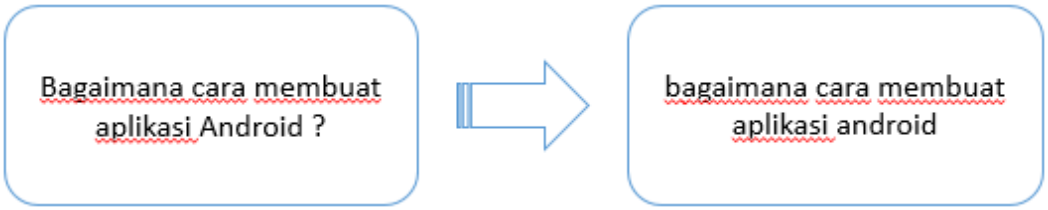
2.3.1 *Text Preprocessing*

Tahapan awal dari *text mining* yaitu *text preprocessing* yang bertujuan untuk mempersiapkan teks menjadi data yang akan diproses pada tahapan berikutnya. Pada *text mining*, informasi yang akan digali berupa data yang tidak terstruktur. Oleh karena itu, diperlukan proses pengubahan bentuk menjadi data yang terstruktur sesuai kebutuhan untuk proses dalam *data mining*, yang biasanya akan menjadi nilai-nilai numerik. Proses ini sering disebut *text preprocessing* [9].

Setelah data menjadi data *terstruktur* dan berupa nilai numerik, maka data dapat dijadikan sebagai sumber data yang dapat diolah lebih lanjut. Kemudian melakukan berberapa proses yang dilakukan sebagai berikut :

1. *Case Folding*

Tahap yang pertama yaitu *case folding*, merupakan tahapan yang mengubah seluruh huruf dalam dokumen sumber teks menjadi huruf kecil, karakter selain huruf harus dihilangkan dan dianggap pembatas [9].



Gambar 2. 1 Proses *Case Folding* [9]

2. *Tokenizing (parsing)*

Tahap selanjutnya yaitu *tokenizing*, merupakan tahap pemotongan *string input* berdasarkan tiap kata yang menyusunnya [9].

Gambar 2. 2 Proses *Tokenizing* [9]

2.3.2 *Text Transformation*

Pada tahap ini yaitu melakukan penyaringan (*filtration*) dan *stemming*. Penyaringan dilakukan dengan menetukan *term* mana yang akan digunakan untuk merepresentasikan dokumen sehingga dapat mendeskripsikan isi dokumen dan membedakan dokumen tersebut dengan dokumen lain. *Text transformation* memliki beberapa tahapan sebagai berikut :

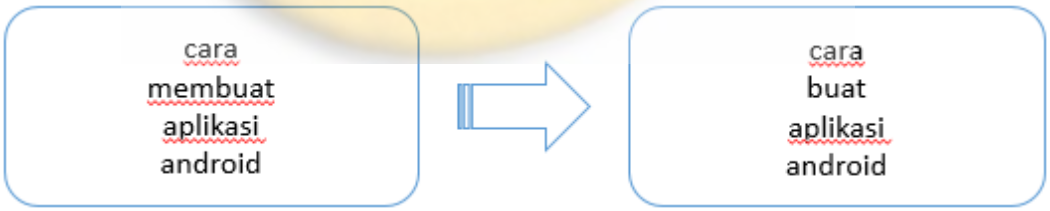
1. *Stopword Removal / Filtering*

Filtering yaitu tahap mengambil kata - kata penting dari hasil token. Dapat menggunakan algoritma *stoplist* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting*). Stoplist / stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words* [9].

Gambar 2. 3 Proses *Filtering* [9]

2. *Stemming*

Stemming merupakan tahapan merubah kata-kata dari tiap kata hasil *filtering* kedalam kata-kata akarnya/kata dasar dengan menggunakan aturan tertentu [9].



Gambar 2. 4 Proses *Stemming* [9]

2.3.3 *Pattern Discovery / Analysis*

Tahapan ini merupakan tahapan terpenting dalam proses *text mining*. Tahapan ini merupakan tahap untuk pembobotan kata dalam suatu dokumen berupa artikel dan kalimat-kalimat yang beranking tinggi untuk dimasukkan ke dalam rangkuman.

Fungsi yang disediakan oleh proses pembobotan yaitu dapat memilih *term* (kata apa saja) yang dapat dijadikan sebagai perwakilan kata yang penting dalam kumpulan dokumen yang akan dianalisis dengan kata lain dalam dokumen, dengan melakukan pembobotan terhadap setiap *term*, dengan menggunakan metode tf-idf yang akan dijelaskan selanjutnya [9].

2.4 *Stemming* Bahasa Indonesia

Stemming merupakan proses membentuk suatu kata menjadi kata akarnya atau kata dasarnya. Contoh :

berkata ⟶ kata

mengatakan ⟶ kata

perkataan ⟶ kata

Untuk *Stemming* bahasa Indonesia beberapa algoritma yang biasanya digunakan yaitu :

a. *Porter Stemmer*. Algoritma ini terkenal dan sering digunakan sebagai *stemmer* untuk bahasa Inggris. *Porter Stemmer* dalam bahasa Indonesia akan menghasilkan kambiguan karena aturan morfologi bahasa Indonesia [11].

b. *Nazief & Adriani Stemmer*. Algoritma ini paling sering dibicarakan dan digunakan dalam *stemming* bahasa Indonesia. Algoritma ini merupakan hasil penelitian internal UI (Universitas Indonesia) dan tidak dipublish secara umum [12]. Tetapi algoritma ini mempunyai dua masalah, yaitu pertama kemampuannya tergantung dari besarnya database kata dasar, dan yang kedua, hasil *stemming* tidak selalu optimal untuk aplikasi *information retrieval* [11].

Bila dibandingkan, untuk teks berbahasa Indoensia, *Porter stemmer* lebih cepat prosesnya daripada Nazief & Adriani *stemmer*, tetapi algoritma Nazief & Adriani memiliki tingkat akurasi lebih tinggi dari *Porter stemmer* [10].

2.4.1 *Stemming* Bahasa Indonesia Algoritma Nazief & Adriani

Adapun langkah-langkah yang digunakan oleh algoritma Nazief dan Adriani yaitu sebagai berikut [13] :

1. Kata dicari di dalam daftar kamus. Bila kata tersebut ditemukan di dalam kamus maka kata tersebut merupakan kata dasar sehingga algoritma dihentikan.
2. Bila kata di dalam langkah pertama tidak ditemukan di dalam kamus, maka diperiksa apakah surfiks tersebut yaitu sebuah partikel ("-lah" atau "-kah"). Bila ditemukan maka partikel tersebut dihilangkan.
3. Kemudain pemeriksaan dilanjutkan pada kata ganti milik ("-ku","-mu","-nya"). Bila ditemukan maka kata ganti tersebut dihilangkan.

4. Pemeriksaan akhiran ("-i", "-an"). Bila ditemukan maka akhiran tersebut dihilangkan. Hingga langkah ini dibutuhkan ketelitian untuk memeriksa apakah akhiran "-an" merupakan hanya bagian dari akhiran "-kan" dan memeriksa lagi apakah partikel ("-lah", "-kah") dan kata ganti milik ("-ku","-mu","-nya") yang telah dihilangkan pada langkah 2 dan 3 bukan merupakan bagian dari kata dasar.
5. Pemeriksaan awalan ("se-","ke-","di-","te-","be-","pe-","me-"). Bila ditemukan, maka awalan tersebut dihilangkan. Pemerikasaan dilakukan dengan berulang *(looping)* mengingat adanya kemungkinan *multi-prefix*. Langka ini juga membutuhkan ketelitian untuk memeriksa kemungkinan peluluhan awalan, perubahan *prefix* yang disesuaikan dengan huruf awal kata dan aturan kombinasi prefix-suffix yang diperbolehkan.

Setelah menyelesaikan semua langkah dengan sukses, maka algoritma akan mengembalikan kata akar atau kata dasar yang ditemukan.

2.5 Metode *Term Frequency-Inversed Document Frequency*

Riset peringkasan teks otomatis dikembangkan sejak tahun 1958 hingga sekarang. Berbagai metode telah diterapkan, seperti metode maximum marginal relevance, lexical chain, berbasis graf dan vector space model. Tetapi salah satu metode yang banyak dikembangkan adalah metode TF-IDF (*Term Frequency-Inversed Document Frequency*).

Pada penelitian ini metode yang digunakan untuk peringkasan teks otomatis adalah TF-IDF (*Term Frequency-Inversed Document Frequency*), karena metode ini memiliki hasil akurasi ringkasan yang cukup akurat dari perhitungan pembobotan setiap kalimat dibandingkan dengan metode lainnya.

Kelebihan dari metode ini yaitu mengekstraksi kalimat-kalimat beranking tinggi untuk dimasukkan ke dalam rangkuman. Metode *Term Frequency-Inversed Document Frequency* (TF-IDF) adalah suatu pengukuran statistik untuk mengukur seberapa penting sebuah kata dalam suatu kumpulan dokumen. Metode ini digunakan untuk menghitung pembobotan *term* dan kalimat yang memiliki nilai tertinggi untuk dimasukkan ke dalam ringkasan [14].

Pada Metode ini pembobotan kata dalam sebuah dokumen dilakukan dengan mengalikan nilai TF dan IDF [14]. Tingkat kepentingan setiap kata akan menjadi meningkat, ketika sebuah kata yang sama muncul beberapa kali dalam suatu dokumen, tetapi diimbangi dengan frekuensi kemunculan kata tersebut dalam kumpulan dokumen. Rumus TF-IDF sebagai berikut :

$$TF\ IDF\ (tk,dj) = TF\ (tk, dj) * IDF\ (tk)$$

Dimana sebelumnya dihitung terlebih dahulu *Term Frequency* (TF) yaitu frekuensi kemunculan suatu *term* di setiap dokumen. Lalu dihitung *Inverse Document Frequency* (IDF) yaitu nilai bobot suatu *term* dihitung dari seberapa seringnya kemunculan kata pada kumpulan kalimat. Semakin sering suatu *term* muncul di dalam dokumen, maka nilai IDF nya akan kecil. Rumus TF dan IDF :

$$TF\ (tk,dj) = f\ (tk,dj)$$

$$IDF\ (tk) = log\ \frac{N}{df\ (t)}$$

Keterangan :

$$W_{dt} = tf_{dt}\ x\ IDF_t \quad \textbf{(3.1)}$$

Diketahui :

d = dokumen ke -d

t = kata ke -t dari kata kunci

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

tf = banyaknya kata yang dicari dalam sebuah dokumen.

IDF = *Inversed Document Frequency*

$$\textbf{IDF} = log_2\left(\frac{N}{df}\right) \quad \textbf{(3.2)}$$

D = total dokumen

df = banyaknya Dokumen yang mengandung kata yang dicari

2.6 *Temporary Folder*

Pada sistem peringkasan teks otomatis yang akan dibuat, untuk mencegah kerusakan data hasil ringkasan pada sistem , dibutuhkan *temporary folder* yang akan menyimpan file berupa JSON, sehingga apabila sistem terjadi *crash* saat

melakukan peringkasan data artikel, maka data tersebut dapat dikembalikan *(backup)*. Pada *temporary folder* ini, data artikel yang akan diringkas pada sistem peringkasan akan disimpan dengan file berupa JSON secara otomatis.

Temporary merupakan *file* yang digunakan secara sementara dan menjadi *redundant* saat ada suatu proses dilakukan sistem. *File* sementara tersebut dibuat untuk menyimpan data sementara ketika ada *file* sedang dibuat atau diproses atau digunakan. *Temporary file* dibuat oleh *Windows* selama kegiatan normal berjalan saat tidak ada cukup memori yang dialokasikan untuk sebuah tugas atau proses.

Beberapa aplikasi yang membutuhkan data besar seperti pengolah grafik, video, atau lainnya dapat membuat *temporary file* yang cukup banyak dan biasanya tetap tertinggal di dalam *harddsik* walaupun proses-proses yang dijalankan aplikasi sudah selesai, jika ini terjadi secara terus-menerus maka dapat memenuhi *harddisk* tanpa kita sadari. *Temporary file* dibuat untuk bertujuan *backup program.*