BAB II

LANDASAN TEORI

2.1 Text Summarization

Automatic Text Summarization adalah sebuah proses untuk menghasilkan ringkasan dengan menggunakan aplikasi, dengan mengambil beberapa informasi penting yang tersedia dari sumber dokumen. *Automatic Text Summarization* menjadi penting, karena keterbatasan manusia untuk menangkap maksud atau informasi penting dari banyaknya informasi yang tersedia secara tepat dan cepat (Goldstein dkk., 2000).

Terdapat dua pendekatan yang ada pada *automatic text summarization*. Pendekatan pertama adalah *extraction-based summarization*, yaitu pendekatan yang dilakukan dengan mengambil kalimat-kalimat yang mengandung informasi penting tanpa melakukan pengubahan terhadap kalimat-kalimat tersebut. Setelah semua kalimat yang mengandung informasi penting telah didapatkan, ringkasan dapat dibentuk dengan menyusun kalimat tersebut sesuai dengan urutan pada dokumen aslinya (Sizov, 2010).

Pendekatan kedua adalah *abstraction-based summarization*, yaitu pendekatan yang dilakukan dengan mengambil kalimat-kalimat yang mengandung informasi penting, kemudian dilakukan pengubahan terhadap struktur kalimat atau paragraf, sehingga akan menghasilkan suatu ringkasan yang mudah dipahami oleh pengguna. Untuk menghasilkan ringkasan dengan pendekatan ini, perlu adanya pengetahuan mengenai *natural language generation technology*. Sehingga, pengembangan sistem *automatic text summarization* dengan menggunakan pendekatan ini lebih sulit (Bhargava dkk., 2016).

Dari kedua pendekatan diatas, sistem yang banyak tersedia saat ini adalah dengan pendekatan *extractive-based summarization*. Pendekatan tersebut akan menghasilkan ringkasan lebih cepat dan tidak akan menggunakan banyak sumber daya, baik waktu ataupun manusia. Banyak algoritma yang dapat digunakan untuk menghasilkan ringkasan dengan pendekatan ini, seperti, *Maximum Marginal Relevance*, *Genetic Algorithm, Graph-Based Algorithm*, *Lexical Chaining*, dan *Shortest Path Algorithm* (Pradnyana dan Mogi, 2014).

2.2 Text Preprocessing

Tahapan pertama untuk memperoleh ringkasan otomatis adalah dengan melakukan *text preprocessing*, yaitu dengan melakukan pengubahan terhadap *input* teks berupa paragraf ke dalam kata-kata yang akan digunakan untuk penghitungan bobot masing-masing kata. Bobot tersebut akan menjadi nilai dari tingkat kepentingan kata tersebut di dalam teks. *Text preprocessing* mengandung beberapa tahapan, yaitu pemisahan kalimat, *case folding*, *tokenizing*, *filtering*, dan *stemming*. Tahapan terakhir dari proses ini menghasilkan kata-kata yang berupa akar kata (Yulita dan Pribadi, 2015). Berikut ini adalah tahapan-tahapan dari proses ini.

1. Pemisahan Kalimat

   Pemisahan kalimat adalah tahapan pertama dalam *text preprocessing*. Pada tahapan ini, setiap paragraf dalam suatu teks atau berita akan dipisah menjadi beberapa kalimat. Pemisahan kalimat ini dilakukan berdasarkan tanda baca akhir kalimat, yaitu titik.

2. Case Folding

Case Folding adalah pemrosesan semua kata-kata ke dalam karakter dengan huruf kecil dan mengabaikan karakter selain dari kata yang ada dalam alfabet. Tanda baca, angka, dan simbol juga akan diabaikan.

3. Filtering

Filtering adalah proses dari penghilangan kata henti. Kata henti adalah kumpulan dari kata-kata yang tidak bermakna penting dalam sebuah kalimat. Proses ini dilakukan dengan melakukan pengecekan terhadap kamus kata henti. Jika kata yang sedang diproses tidak dihilangkan, kata tersebut akan memiliki bobot. Contoh dari kata henti adalah "jika", "maka", "di", dan lain sebagainya.

4. Tokenizing

Tokenizing adalah proses dari pengubahan kalimat menjadi kata akar (*root words*). Pengubahan kalimat pada tahap ini adalah dengan memisahakan setiap kata pada kalimat berdasarkan spasi. Selain itu, setiap kata tersebut akan ditambahkan jumlah kemunculan dalam suatu teks (Kaplan, 2005).

5. Stemming

Stemming adalah proses pengubahan kata ke dalam kata akar atau kata dasar dengan beberapa aturan, sehingga setiap kata akan memiliki representasi yang sama. Metode *stemming* yang akan dilakukan dalah penelitian ini adalah dengan menggunakan metode Porter Stemmer untuk bahasa Indonesia.

2.3 Porter Stemming

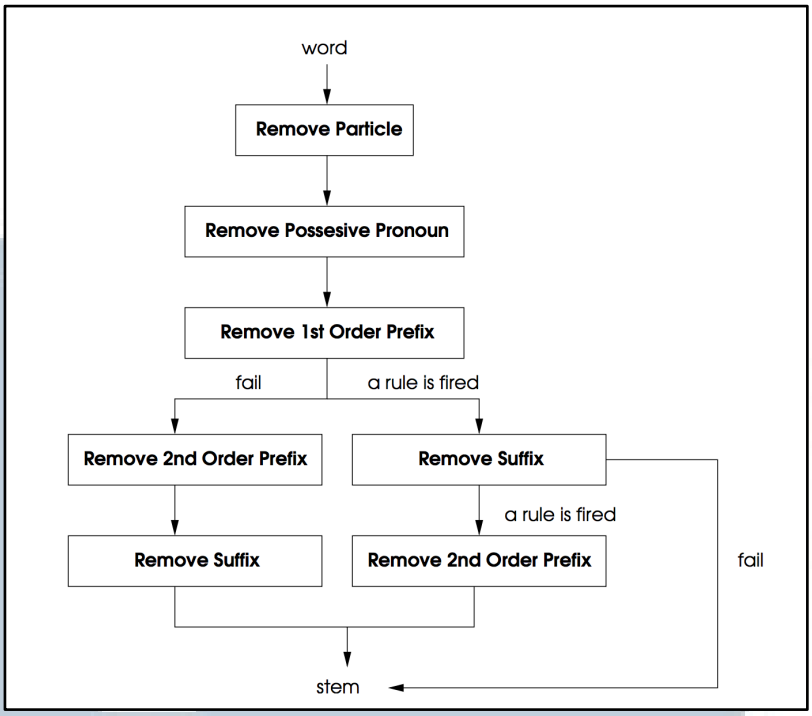
Stemming adalah proses pemetaan dan penguraian berbagai bentuk kata menjadi kata dasarnya (Porter, 1980). Proses *stemming* secara luas sudah digunakan

di dalam proses pencarian informasi (*information retrieval*) untuk meningkatkan kualitas informasi yang didapatkan (Frakes dan Baeza, 1992).

Algoritma Porter Stemming itu sendiri adalah algoritma untuk menghilangkan akhiran dari kata dalam Bahasa Inggris yang dikembangkan oleh M.F. Porter pada tahun 1980. Algoritma Porter Stemming dapat digunakan untuk Bahasa Indonesia karena adanya kesamaan struktur morfologi dari kata-kata dalam Bahasa Indonesia. Struktur morfologi dalam kata-kata Bahasa Indonesia mengandung kombinasi dari suku kata dengan berbagai imbuhan prefiks dan sufiks. Struktur tersebut menjadikan penggunaan Porter Stemming untuk Bahasa Indonesia dapat digunakan (Tala, 2013).

Modifikasi terhadap algoritma Porter Stemming agar sesuai dengan Bahasa Indonesia harus dilakukan karena Bahasa Inggris dan Bahasa Indonesia memiliki kandungan kata yang berbeda-beda. Modifikasi yang dilakukan adalah dengan mengubah *cluster rules* dan *measure condition*. *Measure condition* adalah satuan terkecil bagaimana sebuah kata dapat dibaca. Dalam Bahasa Indonesia, satuan terkecil tersebut adalah suku kata (Tala, 2013).

Dikarenakan algoritma Porter Stemming hanya menghilangkan sufiks pada kata, beberapa tahap ditambahakan untuk menghilangkan prefiks, dan konfiks kata, serta penyesuaian ejaan pada karakter pertama setelah penghilangan imbuhan (Tala, 2013). Proses algoritma Porter Stemming untuk Bahasa Indonesia ditunjukkan pada Gambar 2.1 di halaman selanjutnya.



Gambar 2.1 Proses Algoritma Porter Stemming untuk Bahasa Indonesia

Tahap pertama pada algoritma Porter Stemming ini dengan menghilangkan sufiks kata “-kah”, “-lah”, dan “-pun”. Sebagai contoh, kata “bukukah” menjadi “buku”, “adalah” menjadi “ada”, dan “bukupun” menjadi “buku”. Kemudian, tahap berikutnya adalah dengan menghilangkan sufiks kepemilikan, yaitu sufiks “-ku”, “-mu”, dan “-nya”. Sebagai contoh, kata “sepatumu” menjadi “sepatu” (Tala, 2013).

Tahap ketiga adalah menghilangkan prefiks pertama yaitu “meng-“, “meny-“, “men-“, “mem-“, “me-“, “peng-“, “peny-“, “pen-“, “pem-“, “di-“, “ter-“, dan “ke-“. Khusus untuk prefiks kata “meny-“, terdapat aturan tambahan setelah menghilankan prefiks tersebut, yaitu menggantinya dengan huruf “s”, jika huruf pertama setelah prefiks tersebut dihilangkan adalah huruf vokal (*vowel*). Berlaku pula aturan yang sama pada prefiks “mem-“ dengan menggantinya menjadi “p”, “peny-“ dengan menggantinya menjadi “s”, dan “pem-“ dengan menggantinya menjadi “p” (Tala, 2013).

Tahap berikutnya adalah menghilangkan prefiks kedua, yaitu prefiks “ber-“, “bel-“, “be-“, “per-“, “pel-“, dan “pe-“. Menghilangkan prefiks “bel-“ dan “pel-“ hanya berlaku untuk kata dengan suku kata “ajar”. Kemudian, prefiks “be-“ akan dihilangkan jika huruf setelah prefiks tersebut adalah huruf konsonan diikuti oleh “-er”. Sebagai contoh kata “bekerja” menjadi “kerja” dan kata “bepergian” menjadi “pergian”. Tahap terakhir adalah menghilangkan sufiks “-kan” dengan aturan prefiks tidak termasuk dalam “ke-“ dan “peng-“, sufiks “-an” dengan aturan prefiks tidak termasuk dalam “di-“, “meng-“, dan “ter-“, dan sufiks “-i” dengan aturan prefiks tidak termasuk dalam “ber-“, “ke-“, dan “peng-“.

2.4 Algoritma Maximum Marginal Relevance

Maximum Marginal Relevance adalah suatu cara pengukuran yang digunakan untuk mengukur derajat keberagaman dari kalimat yang akan dipilih dengan kalimat yang sudah dihasilkan dari proses ekstraksi. Tujuan dari penggunaan metode ini adalah untuk mendapatkan kalimat sebagai sumber informasi penting yang tidak sama dengan kalimat yang sudah menjadi informasi penting (Carbonell dan Goldstein, 1998).

Metode ini merupakan salah satu metode yang digunakan untuk mendapatkan kalimat atau teks dari sebuah atau banyak dokumen dengan melakukan pengurutan dan pembobotan antar kalimat dalam dokumen. *Maximum Marginal Relevance* melakukan perhitungan dengan mengurangkan dua buah *cosine similiarity matrix*, yaitu matriks yang berisi semua kalimat dan matriks yang berisi kalimat pilihan untuk ringkasan.

Cosine similarity adalah sebuah perhitungan yang biasa digunakan dalam proses penerimaan informasi (*information retrieval*) dengan tujuan untuk menghitung derajat kesamaan antar dua buah matriks yang berasal dari dokumen teks. Rumus yang digunakan untuk mendapatkan nilai *cosine similarity* ini adalah sebagai berikut (Korenius dkk., 2007).

$$similarity = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i{}^2} \times \sqrt{\sum_{i=1}^{n} B_i{}^2}} \quad \ldots(2.1)$$

Rumus tersebut akan membagi hasil dari *dot product* atau perkalian matriks A_i dan matriks B_i dengan perkalian dari akar jumlah setiap elemen matriks A_i kuadrat dan akar jumlah setiap elemen matriks B_i kuadrat.

Setelah mendapatkan nilai kesamaan dari setiap kalimat, perhitungan dari metode Maximum Marginal Relevance ini adalah dengan mengambil kalimat dengan nilai tertinggi dari setiap iterasi perhitungan. Iterasi perhitungan akan berhenti, jika nilai maksimal yang dihasilkan adalah ≤ 0. Berikut adalah rumus yang digunakan pada metode ini (Carbonell dan Goldstein, 1998).

$$MMR(Si) = \lambda \cdot Sim1(Si, Q) - (1 - \lambda) \cdot \max Sim2(Si, S') \quad \ldots(2.2)$$

Dimana λ merupakan angka konstan bernilai 0.7 (Carbonel dan Goldstein, 1998), kemudian Sim1 (Si, Q) adalah kalimat yang akan dihitung persamaannya dan $(1 - \lambda)$. $max\ Sim2\ (Si\ ,\ S')$ adalah nilai maksimum persamaan dari kalimat yang telah dipilih menjadi ringkasan.

Tujuan pengurangan antara nilai *cosine similarity* kalimat yang sedang dihitung dengan *cosine similarity* kalimat yang telah menjadi pilihan ringkasan adalah untuk menghindari redudansi, atau kalimat dengan makna yang sama (Carbonell dan Goldstein, 1998).

2.5 System Usability Scale

System Usability Scale merupakan metode yang digunakan untuk mengukur tingkat kegunaan dari sistem yang *user* sedang atau akan gunakan. Tiga hal utama yang dijadikan pertimbangan untuk menilai sebuah sistem dapat digunakan dengan baik atau tidak adalah efektifitas, efisiensi, dan tingkat kepuasan (Brooke, 2003).

Sebuah sistem dibangun untuk membantu para penggunanya menyelesaikan pekerjaan-pekerjaan mereka. Efektifitas dari sebuah sistem dapat dinilai berdasarkan tujuan dari sebuah sistem itu dibangun. Jika sistem yang dibangun tepat sasaran, nilai efektifitas dari sistem tersebut akan tinggi (Brooke, 2003).

Penilaian terhadap tingkat efisiensi dari sebuah sistem akan selalu berhubungan dengan sumber daya, terutama sumber daya waktu. Sebuah sistem dapat dikatakan efisien jika dapat mengerjakan pekerjaan secara cepat dan tepat, termasuk dalam penanganan *error* yang diperoleh dari pengguna sistem (Brooke, 2003).

Pernilaian terhadap tingkat kepuasan dari pengguna kepada sebuah sistem dapat dilihat dari bagaimana pengguna merasa nyaman untuk menggunakan sistem dalam jangka waktu yang lama. Penilaian terhadap tingkat kepuasan pengguna pada sistem juga dapat diperoleh dengan melakukan penilaian serupa pada sistem lain yang sejenis (Brooke, 2003).

Berdasarkan ketiga pertimbangan di atas, System Usability Scale digunakan untuk mendapatkan pandangan atau penilaian subjektif pengguna terhadap sebuah sistem dan secara cepat dapat mengevaluasi sistem yang telah dibangun (Brooke, 2003).

Terdapat 10 pertanyaan pada System Usability Scale dengan masing-masing pertanyaan akan dijawab memiliki skala nilai dari 0 sampai dengan 4 (Brooke, 2003). Tabel 2.1 merupakan daftar pertanyaan dari System Usability Scale.

Tabel 2.1 Daftar Pertanyaan System Usability Scale

No.	Pertanyaan
1.	Saya akan sering menggunakan aplikasi ini
2.	Saya merasa aplikasi ini sangat kompleks
3.	Saya sangat mudah menggunakan aplikasi ini
4.	Saya membutuhkan bantuan dari sisi teknis agar dapat menggunakan aplikasi ini
5.	Saya merasa setiap fungsi yang ada dapat berjalan dengan baik
6.	Saya merasa ada banyak inkonsistensi selama penggunaan aplikasi ini
7.	Saya yakin banyak orang akan mudah mempelajari aplikasi ini dengan cepat
8.	Saya merasa aplikasi ini sangat merepotkan untuk digunakan
9.	Saya merasa percaya diri menggunakan aplikasi ini
10.	Saya perlu mempelajari banyak hal terlebih dahulu sebelum saya dapat menggunakan aplikasi ini

2.6 Skala Likert

Skala Likert merupakan adalah sebuah respons psikometrik yang digunakan dalam sebuah kuisioner untuk mendapatkan pilihan atau tingkat kesepakatan responden terhadap suatu pernyataan. Skala Likert merupakan sebuah teknik non-komparatif dan unidimensional (hanya mengukur suatu sifat). Responden diminta untuk menunjukkan tingkat kesepakatan mereka dengan pernyataan yang diberikan dengan menggunakan skala ordinal (Bertram, 2008).

Variasi dari skala ordinal yang diberikan pada Skala Likert ini adalah sebuah skala lima poin yang ditampilkan dari “sangat tidak setuju” sampai dengan “sangat

setuju". Namun, ada beberapa orang yang menggunakan skala hingga tujuh atau sembilan poin untuk menambahkan perincian terhadap sesuatu yang sedang diteliti. Setiap poin yang ada pada Skala Likert diberikan nilai atau kode, biasanya dimulai dari angka 1 dan akan bertambah 1 sampai poin terakhir (Bertram, 2008). Gambar 2.2 menunjukkan skala yang digunakan dalam Skala Likert.

Gambar 2.2 Skala dalam Skala Likert

Untuk mendapatkan nilai rata-rata dari setiap pertanyaan yang ada dalam Skala Likert, setiap nilai atau bobot yang telah ada dalam masing-masing poin dalam Skala Likert akan dikalikan dengan banyaknya responden yang memilih poin tersebut. Rumus yang digunakan adalah sebagai berikut (Simpson, 2010).

$$f_{ave} = \frac{\sum_i f_i w_i}{\sum_i f_i} \quad \ldots(2.3)$$

Dimana f_i merupakan jumlah dari banyaknya responden yang memilih poin dengan bobot w_i. Kemudian total dari perhitungan tersebut akan dibagi dengan jumlah responden secara keseluruhan. Nilai yang didapatkan tersebut merupakan nilai dari sebuah pertanyaan. Perhitungan dapat dikembangkan untuk mencari bobot rata-rata dari setiap pertanyaan yang ada untuk mendapatkan hasil akhir atau pendapat responden mengenai suatu penelitian yang berjalan (Simpson, 2010).