

Analisis Fitur Kalimat untuk Peringkas Teks Otomatis pada Bahasa Indonesia

B. Zaman dan E. Winarko

Abstract— *Automatic Text Summarization (ATS) is a technique to create a summary of the document automatically by using computer applications to produce the most important information from the original document. Features are required to perform weighting of sentences, including Log-TFISF (term frequency index sentence frequency), sentence location, sentence overlap, title overlap and sentence relative length. This research conducted an analysis of five features in order to determine the weights of each feature that will get the results of a coherent summary. The five features are implemented in automated text summarization system in Indonesian language that was developed using the method of relative importance of topics. Results from experiments show that sentence location feature has the highest F-Measures namely 0.46 and then consecutive sentence overlap, title overlap, sentence relative length and Log-TFISF, with a value of 0.42, 0.42, 0.35 and 0.32. Relative weights of feature extraction consecutive from the largest are sentence location, sentence overlap, title overlap, sentence relative length and Log-TFISF with a value of 0.25, 0.22, 0.22, 0.19 and 0.12. These relative weights are implemented on ATS, so we get accuracy of 70.62%. It is more accurate 2,86% than without relative weights which accuracy of 67,72%.*

Keywords— *Automatic Text Summarization (ATS), Log-TFISF, sentence location, sentence overlap, title overlap, sentence relative length, bahasa Indonesia*

1. PENDAHULUAN

Summary atau ringkasan didefinisikan sebagai penyajian bacaan dalam bentuk singkat dengan mempertahankan urutan isi dan sudut pandang / memendekkan bacaan dengan mengambil inti sari bacaan itu tanpa mengubah struktur wacana [1]. 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.

Badrus Zaman adalah staf pengajar, Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Airlangga, Kampus C Mulyosari, Surabaya, 61115, Telp. 031-5965257, e-mail: badruszaman@unair.ac.id

Edi Winarko adalah staf pengajar Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Sekip Utara, Bulak Sumur, Yogyakarta Telp 0274-5551333, e-mail: ewinarko@ugm.ac.id

Sebuah sistem peringkas diberi masukan berupa teks, kemudian melakukan peringkasan, dan menghasilkan keluaran berupa teks yang lebih singkat dari teks asli. Hasil peringkasan mengandung poin-poin penting atau informasi utama dari teks sumber. Peringkasan dapat dilakukan dengan 2 metode yakni ekstraksi kalimat yang menggunakan pendekatan statistik (*selection based*) dan abstraksi kalimat yang menggunakan pendekatan pengetahuan (*knowledge based*) [2]. Metode ekstraksi kalimat adalah ringkasan yang berisi kalimat-kalimat dari dokumen sumber tanpa ada perubahan kalimat-kalimat tersebut, atau dengan kata lain ekstrak adalah kumpulan kalimat representatif dari sumber dokumen. Metode abstraksi pada sisi lain, menggunakan metode linguistik untuk memeriksa dan menafsirkan dokumen, di mana sudah ada perubahan pada kalimat-kalimat hasil ringkasan [2].

Untuk menggunakan metode ekstraksi diperlukan fitur. Fitur ini digunakan untuk merepresentasikan data yang digunakan dalam proses ini, di mana data adalah kumpulan kalimat[3]. Fitur-fitur kalimat memiliki bermacam variabel, di antaranya menurut Suanmali dkk., *title feature, sentence length, term weight, sentence position, sentence to sentence similarity, proper noun, thematic word* [3].

Penelitian ini difokuskan pada analisis fitur kalimat untuk peringkas teks otomatis pada Bahasa Indonesia. Fitur yang akan dianalisis terdapat 5 jenis yaitu *Log-TFISF (term frequency index sentence frequency), Sentence Location, Sentence Overlap, Title Overlap* dan *Sentence Relative Length*. Dari hasil analisis tersebut diharapkan dapat ditentukan pembobotan masing-masing fitur sehingga akan didapatkan hasil ringkasan yang lebih koheren.

Menurut Mani dkk. dan Hassel metode untuk melakukan evaluasi terhadap hasil dari ATS secara umum dibagi 2 yaitu metode evaluasi intrinsik dan ekstrinsik [2,4]. Metode evaluasi intrinsik adalah evaluasi yang dilakukan oleh sistem peringkas itu sendiri, misalnya menggunakan *F-Measures*. Evaluasi ini difokuskan pada tingkat koheren dan informatif dari hasil ringkasan. Sedangkan evaluasi metode ekstrinsik adalah menghitung efektivitas dan

akseptabilitas dari hasil ringkasan untuk tugas-tugas tertentu, misalnya *assessment* terhadap hasil ringkasan. Dalam penelitian ini, metode evaluasi yang digunakan adalah penghitungan *F-Measures (F-Scores)* menggunakan perhitungan *Precision* dan *Recall* yang merupakan standar evaluasi dalam penghitungan *information retrieval* [2]. Menurut Mani dkk. standar evaluasi perhitungan *information retrieval* juga dapat digunakan dalam evaluasi perhitungan ATS [4].

2. METODE PENELITIAN

2.1. Ringkasan

Menurut [1], ringkasan adalah penyajian bacaan dalam bentuk singkat dengan mempertahankan urutan isi dan sudut pandang/memendekkan bacaan dengan mengambil inti sari bacaan itu tanpa mengubah struktur wacana. Sebuah ringkasan bermula dari karangan sumber yang panjang, kemudian dipangkas dengan mengambil hal-hal atau bagian yang pokok dengan membuang perincian serta ilustrasi. Meskipun begitu, sebuah ringkasan tetap mempertahankan pikiran pengarang serta pendekatannya yang asli. Jadi, meringkas merupakan keterampilan mereproduksi hasil karya yang sudah ada dalam bentuk yang singkat [1].

Menurut [1], ringkasan berbeda dengan ikhtisar. Walaupun kedua istilah itu sering disamakan, tapi sesungguhnya keduanya berbeda. Ringkasan merupakan penyajian singkat dari suatu karangan asli namun tetap mempertahankan urutan isi dan sudut pandang pengarang asli. Ikhtisar sebaliknya, tidak perlu mempertahankan sistematika penulisan sesuai dengan aslinya dan tidak perlu menyajikan isi dari seluruh karangan itu secara proporsional. Dalam ikhtisar, penulis dapat langsung mengemukakan pokok uraian, sementara bagian yang dianggap kurang penting dapat dibuang [1].

Ringkasan dapat disusun dalam dua bentuk, yaitu bentuk verbal uraian (paragraf) dan bentuk nonverbal berupa bagan atau skema. Meskipun ringkasan berbentuk bagan atau skema, tetapi harus mencerminkan gagasan atau seperti yang diungkapkan oleh teks sumbernya. Sebelum membuat bagan atau skema, harus dicatat terlebih dulu butir-butir informasi yang akan dijadikan unsur-unsur bagan atau skema [1].

2.2. Automatic Text Summarization (ATS)

ATS adalah teknik pembuatan ringkasan dari sebuah teks secara otomatis dengan

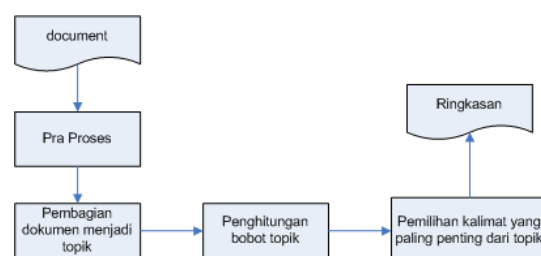
memanfaatkan aplikasi yang dijalankan pada komputer untuk menghasilkan informasi yang paling penting dari dokumen aslinya[5]. Sistem diberi masukan berupa teks, kemudian melakukan peringkasan, dan menghasilkan keluaran berupa teks yang lebih singkat dari teks asli. Hasil peringkasan mengandung poin-poin penting atau informasi utama dari teks sumber[5,6].

Penelitian ATS mulai dilakukan oleh Luhn di IBM pada tahun 1958 [6]. Beberapa peneliti kemudian melanjutkan penelitian ini dengan mencoba pendekatan-pendekatan baru sampai tahun 1980-an.

Pada dasarnya, tugas *summarization* dapat dibagi menjadi dua tahap utama: (a) pembangunan representasi teks; (b) membuat ringkasan. Pembangunan representasi teks di antaranya terdapat proses pembagian dokumen menjadi kalimat dan *stemming* [7]. Sedangkan pendekatan untuk mendapatkan ringkasan dapat dilakukan dengan 2 metode yakni ekstraksi kalimat yang menggunakan pendekatan statistik (*selection based*) dan abstraksi kalimat yang menggunakan pendekatan pengetahuan (*knowledge based*) [2,6]. Metode ekstraksi, ringkasan yang dihasilkan berisi kalimat-kalimat dari dokumen sumber tanpa ada perubahan kalimat-kalimat tersebut, atau dengan kata lain ekstrak adalah kumpulan kalimat representatif dari sumber dokumen, sedangkan abstraksi menggunakan metode linguistik untuk memeriksa dan menafsirkan dokumen, di mana sudah ada perubahan pada kalimat-kalimat hasil ringkasan [2,6]. Pendekatan statistik paling banyak dipakai karena relatif mudah dan cepat [6].

2.3. Gambaran Umum Sistem ATS

Menurut [8], ATS dapat dibagi menjadi 4 tahap yaitu pra proses, pembagian dokumen menjadi topik, penghitungan bobot topik, dan pemilihan kalimat yang paling penting dari topik. Tahapan ini ditunjukkan oleh gambar 1[8].



Gambar 1 Gambaran umum sistem ATS

2.3.1. Pra Proses

Pada tahap pra proses (*preprocessing*) dilakukan penyiapan dokumen mentah menjadi dokumen atau representatif dokumen yang siap diproses untuk langkah selanjutnya. Pada tahap ini proses yang dilakukan antara lain membagi dokumen menjadi kalimat, membagi dokumen menjadi kata, menghapus *stopword* dan melakukan proses *stemming* [8].

Stopword adalah daftar kata yang jumlahnya sangat besar yang mempunyai fungsi namun tidak perlu digunakan sebagai penciri dokumen [9]. Contohnya: dan, yang, atau, ini, itu dan lainnya. Daftar kata *stopword* juga sering disebut dengan *stoplist*. Hasil penelitian Tala, menyebutkan bahwa penghapusan *stopword* dapat meningkatkan kemampuan pemrosesan sistem temu kembali informasi [9]. *Stemming* adalah proses pemetaan dari penguraian berbagai bentuk kata baik itu prefix, suffix, maupun gabungan antara prefix dan suffix (*confix*), menjadi bentuk kata dasarnya (*stem*) [3,5,6,7].

2.3.2. Pembagian dokumen menjadi topik

Pembagian dokumen dilakukan menggunakan algoritma *texttiling* yang dimodifikasi oleh Neto dkk. [8]. Sebelum proses pembagian dokumen menjadi topik, langkah yang harus dilakukan adalah membagi dokumen menjadi blok-blok yang terdiri dari beberapa kalimat, biasanya tiap blok terdiri dari 3 sampai 5 kalimat. Menurut Chang dan Lee *short text* seperti berita internet yang terdiri dari 30-60 kalimat, tiap blok menggunakan 3 kalimat [10]. Pembobotan kata dilakukan dengan persamaan TFIBF (*term frequency index block frequency*). TFIBF hampir sama dengan TFIDF hanya saja istilah dokumen diganti dengan blok. Rumus ini ditunjukkan pada rumus 1 [8].

$$TFIBF(w,b) = tf(w,b) \cdot \log\left(\frac{N}{n_w}\right) \quad (1)$$

di mana $tf(w,b)$ adalah jumlah kata w pada blok b , N adalah jumlah blok, sedangkan n_w adalah jumlah blok yang mengandung kata w .

Algoritma *texttiling* yang digunakan pada penelitian ini dibagi menjadi 2 langkah yaitu penghitungan kemiripan antar blok yang berdekatan dan penentuan adanya pergantian topik. Penghitungan kemiripan antar blok yang berdekatan menggunakan persamaan kosinus. Kemiripan antara blok i dan blok $i+1$ menggunakan persamaan kosinus pada rumus 2 [8].

$$\cos(b_i, b_{i+1}) = \frac{\sum_{t=1}^n w_{t,b_i} w_{t,b_{i+1}}}{\sqrt{\sum_{t=1}^n w_{t,b_i}^2 \sum_{t=1}^n w_{t,b_{i+1}}^2}} \quad (2)$$

di mana b_i adalah blok ke- i , b_{i+1} adalah blok ke- $i+1$ dan t adalah indeks kata.

Untuk melakukan penentuan adanya pergantian topik maka perlu dibuat nilai *threshold* (batas pergantian topik) yang merepresentasikan beda kemiripan antar blok. Nilai *threshold* yang digunakan antara 0.1 dan 0.2 [9]. Sedangkan untuk kasus *short text* nilai *threshold* ditentukan sebesar 0.1 [10].

Algoritma untuk menemukan topik ini dijelaskan pada gambar 2, di mana $sim(p)$ adalah nilai kemiripan antara blok kalimat posisi ke- p dengan blok kalimat posisi ke- $(p+1)$ [8].

```

scan the graph, beginning from position p := 0; direction := none; counter := 0;
F sim(p+1) - sim(p) > 0
  THEN the line of similarity between blocks is rising;
    IF direction=falling and counter > threshold
      THEN add current block to the list of topics;
        direction := rising;
    ELSE the line of similarity between blocks is falling;
      IF direction=rising
        THEN counter := sim(p+1) - sim(p);
          direction := falling;

```

Gambar 2 Pseudocode untuk mendeteksi pergantian topic [8]

Algoritma ini memungkinkan untuk mendeteksi topik di mana terjadi pada blok-blok yang mengalami penurunan tingkat kemiripan sebesar nilai *threshold* yang telah ditentukan.

2.3.3. Penghitungan bobot topik

Setelah dokumen dibagi menjadi topik-topik kemudian dihitung bobot tiap topik tersebut, dengan tujuan untuk mengekstrak lebih banyak kalimat dari topik yang berbobot tinggi [8]. Bobot suatu topik adalah total bobot tiap kalimat dalam topik tersebut. Untuk menghitung bobot topik dilakukan dengan 2 langkah. Pertama, bobot tiap kalimat dihitung dengan rumus TFISF dan posisi dari topik. TFISF ini seperti juga TFIBF adalah modifikasi dari TFIDF dengan mengganti dokumen menjadi kalimat. Untuk menghitung bobot topik ditunjukkan dengan *TopImp*, dengan rumusan pada rumus 3 [8].

$$TopImp(t) = \sum_{s=1}^{|S(t)|} \left(\frac{\sum_{w=1}^{|W(s)|} TFISF(w,s)}{|W(s)|} \right) * (N+1-t) \quad (3)$$

di mana t adalah indeks topik, w adalah kata, s adalah kalimat, $|W(s)|$ adalah jumlah kata dalam kalimat s , dan $|S(t)|$ adalah jumlah kalimat dalam topik t . Sedangkan N adalah jumlah topik dalam dokumen.

Kedua, setelah semua topik dihitung bobotnya, kemudian dihitung bobot relatif topik (*RelTopImp*) untuk menormalisasi rumus di atas sehingga didapatkan hasil antara 0 dan 1, dengan rumus 4.

$$RelTopImp(t) = \frac{TopImp(t)}{\sum_{i=1}^{|T|} TopImp(i)} \quad (4)$$

di mana *t* adalah indeks dari topik dan *|T|* adalah jumlah topik pada dokumen.

2.3.4. Pemilihan kalimat paling penting dari topik

Jumlah kalimat yang harus diekstrak dari tiap topik disesuaikan dengan bobot relatif topik tersebut [8]. Pada tahap sebelumnya dokumen telah dibagi menjadi topik-topik dan dihitung bobot relatif tiap topik. Misalkan dokumen yang terdiri dari 25 kalimat dibagi menjadi 7 topik (lihat Tabel 1).

Tabel 1 Perhitungan bobot relatif topik suatu dokumen

Topik	Relative Importance	Kalimat
1	0,25	1,2,3,4,5
2	0,21	6,7,8,9
3	0,03	10,11,12
4	0,02	13,14,15
5	0,13	16,17,18,19
6	0,19	20,21,22
7	0,17	23,24,25
Jumlah	1	

Untuk menghitung jumlah kalimat yang diekstrak dari tiap topik disesuaikan dengan bobot relatif topik tersebut. Algoritma yang digunakan mengacu pada gambar 3. Variabel *t* adalah jumlah topik, *N* adalah jumlah kalimat, *tot_sel* adalah jumlah kalimat yang dipilih, dan *sel[i]* adalah jumlah kalimat yang dipilih dari topik ke-*i*.

```

/* t = number of topics */
/* N = user-specified number of sentences to be selected for the summary */
/* tot_sel = number of sentences to be selected for the summary */
/* sel_i = (not necessarily integer) number of sentences from topic i to be selected
   for the summary, computed by RelTopImp(i) * tot_sel. */
/* int_sel_i = integer number of sentences from topic i to be selected for summary */
input: sel, i = 1,...,t
output: int_sel, i = 1,...,t
tot_sel = 0;
FOR i = 1,...,t
  int_sel_i = truncate(sel_i); leftover_i = sel_i - int_sel_i; tot_sel = tot_sel + int_sel_i;
ENDFOR
WHILE (tot_sel < N)
  find topic t such that leftover_t is the maximum over all leftover_t, i = 1,...,t
  int_sel_t = int_sel_t + 1; leftover_t = 0; tot_sel = tot_sel + 1;
ENDWHILE
    
```

Gambar 3 Pseudocode menghitung jumlah kalimat yang diekstrak per topik [8]

Misalkan pengguna menginginkan ringkasan sebanyak 6 kalimat, maka langkah-langkah yang dilakukan adalah, pertama, bobot relatif topik dikalikan dengan jumlah kalimat ringkasan yaitu dikalikan dengan 6. Jadi untuk topik 1 sampai 7 masing-masing akan mendapat nilai 1.5, 1.26, 0.18, 0.12, 0.78, 1.14 dan 1.02. Langkah

berikutnya nilai-nilai tersebut dibulatkan, sehingga didapatkan nilai 2, 1, 0, 0, 1, 1, dan 1. Pada iterasi pertama *statement* WHILE, didapatkan *int_sel[2] = 2*, lalu pada iterasi kedua diperoleh *int_sel[3] = 1* dan *N = 5* sehingga algoritma selesai. Jadi untuk topik 1 dipilih 1 kalimat, topik 2 dipilih 2 kalimat, topik 3 dipilih 1 kalimat, dan topik 5 sebanyak 1 kalimat. Sebagai ringkasan dari ilustrasi di atas dapat dilihat tabel 2.

Tabel 2 Perhitungan bobot relatif topik untuk ekstrak kalimat

Topik	Relative Importance (RI)	Kalimat	RI*6	N
1	0,25	1,2,3,4,5	1,5	2
2	0,21	6,7,8,9	1,26	1
3	0,03	10,11,12	0,18	0
4	0,02	13,14,15	0,12	0
5	0,13	16,17,18,19	0,78	1
6	0,19	20,21,22	1,14	1
7	0,17	23,24,25	1,02	1
Jumlah	1		6	6

N = Jumlah kalimat yang diambil

Algoritma tersebut hanya menghitung jumlah kalimat yang harus dipilih dari tiap topik tetapi belum diketahui kalimat mana yang harus dipilih dari tiap topik tersebut. Oleh karena itu diperlukan pembobotan kalimat agar didapatkan kalimat ringkasan yang berbobot lebih tinggi berdasarkan fitur ekstraktor tertentu.

2.4 Fitur Ekstraktor

Fitur ekstraktor adalah fitur yang digunakan untuk merepresentasikan data yang digunakan dalam proses ATS dengan metode ekstraksi [3]). Contoh fitur ekstraktor antara lain Log-TFISF, posisi kalimat, kemiripan dengan kalimat pertama, kemiripan dengan judul, dan panjang kalimat.

2.4.1. Log TFISF

Metode ini menghitung pembobotan kalimat berdasarkan nilai Log-TFISF pada masing-masing kata dalam kalimat tersebut. Rumusan ini diadopsi dari Neto dkk, namun ditambah fungsi logaritma didepannya [11] sehingga rumusnya menggunakan rumus 5.

$$Log - TFISF(i) = \log tf(i, s) \cdot \log \left(\frac{N}{S_i} \right) \quad (5)$$

di mana *tf(i,s)* adalah frekuensi kata *i* dalam kalimat *s*, *N* adalah jumlah kalimat dalam dokumen dan *S_i* adalah jumlah kalimat yang mengandung kata *i*.

Langkah selanjutnya dihitung rata-rata *TFISF* tiap kalimat menggunakan rumus 6.

$$mLogTFISF(j) = l \frac{\sum_{j=1}^n \text{Log} - TFISF(i, j)}{|S|} \quad (6)$$

di mana $|S|$ adalah jumlah kata dalam kalimat j .

2.4.2. Posisi kalimat

Metode ini menghitung bobot kalimat dari posisi kalimat dalam dokumen dengan menggunakan rumus 7 [6]. Kalimat yang terletak pada awal dokumen cenderung memiliki bobot yang lebih tinggi daripada kalimat yang terletak pada akhir dokumen.

$$wP_i = \frac{n-i+1}{n} \quad (7)$$

di mana wP_i adalah bobot kalimat, i adalah posisi kalimat sedangkan n adalah jumlah kalimat dalam dokumen.

2.4.3 Kemiripan dengan kalimat pertama

Bobot suatu kalimat yang dihitung dengan metode ini adalah hasil perkalian vektor kalimat pertama dengan kalimat lain. Hal ini didasari oleh pemikiran bahwa pokok pikiran suatu dokumen seringkali diletakkan oleh penulis berita di bagian awal. Rumus yang digunakan pada rumus 8 [6].

$$w \circ v(1, j) = \frac{\sum_{w=1}^{|w(1)|} f(w, 1) \cdot f(w, j)}{\text{Max}(w \circ v(1, j))} \quad (8)$$

di mana $|w(1)|$ adalah jumlah kata dalam kalimat ke-1, sedangkan $f(w, 1)$ dan $f(w, j)$ masing-masing adalah jumlah kata w dalam kalimat ke-1 dan j .

2.4.4 Kemiripan dengan judul

Metode ini hampir sama dengan metode kemiripan dengan kalimat pertama hanya saja tiap kalimat dibandingkan dengan judul dokumen [6]. Rumus 9 merupakan rumusan dari fitur ini.

$$w \circ v(t, i) = \frac{\sum_{w=1}^{|w(t)|} f(w, t) \cdot f(w, i)}{\text{Max}(w \circ v(t, i))} \quad (9)$$

di mana $|w(t)|$ adalah jumlah kata dalam judul, sedangkan $f(w, t)$ dan $f(w, i)$ masing-masing adalah jumlah kata w dalam judul dan jumlah kata w dalam kalimat ke- i .

2.4.5 Panjang relatif kalimat

Metode ini digunakan untuk memberikan bobot rendah pada kalimat-kalimat pendek, sehingga kemungkinan untuk dimasukkan ke ringkasan menjadi kecil [6]. Rumus yang digunakan pada rumus 10.

$$wLen(w_i) = \frac{|S_i|}{\max_{S_j \in D(S_i)} |w_z|} \quad (10)$$

di mana $|S_i|$ adalah jumlah kata dalam kalimat S_i dan $\max_{S_j \in D(S_i)} |w_z|$ adalah jumlah kata terbanyak dalam suatu kalimat dalam dokumen $D(S_i)$.

2.5. Evaluasi Hasil ATS

Metode yang digunakan untuk mengevaluasi sistem ATS sampai saat ini masih diperdebatkan oleh para pakar *Information Retrieval* (IR). Hal ini dikarenakan banyaknya parameter dan variabel yang terlibat di dalamnya. Secara umum evaluasi sistem ATS dapat dilakukan dengan metode intrinsik maupun ekstrinsik. Metode evaluasi intrinsik mengukur kualitas sistem ATS itu sendiri yang difokuskan pada seberapa koheren dan informatif dari suatu ringkasan, biasanya dibandingkan dengan bantuan ringkasan yang dibuat oleh manusia. Sedangkan metode ekstrinsik mengukur bagaimana pengaruhnya (efektivitas dan akseptabilitas) hasil ringkasan tersebut pada sistem lain misalnya pencarian dokumen, *indexing*, atau klasifikasi dokumen [2]. Metode evaluasi yang biasanya digunakan pada *information retrieval* adalah metode intrinsik dengan *Precision / Recall* dan *F-Measures* yang dirumuskan dengan rumusan [2,6,12].

$$\text{Recall} = \frac{\text{jumlah ka lim at relevan yang diha silkan sistem}}{\text{jumlah ka lim at relevan}} \quad (11)$$

$$\text{Precision} = \frac{\text{jumlah kãimat relevanyang dihasilkan sistem}}{\text{jumlah kãimat yang dhasilkan sistem}} \quad (12)$$

$$F\text{-Measures} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (13)$$

Dalam metode intrinsik, *precision* dan *recall* digunakan untuk mengukur kualitas ringkasan otomatis dengan cara membandingkan ringkasan otomatis dengan ringkasan manual (buatan manusia). Kemudian hasil akhir akan didapatkan dengan cara penggabungan nilai *recall* (rumus 11) dan *precision* (rumus 12) yang disebut dengan nilai *F-measures* (rumus 13). Masalah dalam metode ini adalah dalam menentukan kalimat relevan karena pasti terdapat perbedaan pendapat antar subyek pembuat ringkasan dalam memilih kalimat. Misal subyek A menganggap kalimat x sebagai kalimat utama tetapi bisa saja subyek B berpendapat sebaliknya bahwa kalimat x kurang begitu penting. Untuk mengatasi hal ini digunakan beberapa metode seperti suara terbanyak (*majority opinion*), gabungan (*union*), atau irisan (*intersection*).

2.6. Rancangan Proses dengan DFD

Perancangan proses perlu dilakukan untuk mengetahui proses-proses yang diperlukan dalam pembuatan aplikasi, aliran data pada tiap-tiap proses hingga aktor yang terlibat di dalamnya. Perancangan ini bertujuan untuk mengetahui proses transformasi data dari input berupa dokumen hingga menjadi output berupa hasil ringkasan.

2.6.1. Diagram konteks

Diagram konteks dibuat untuk menggambarkan sistem secara umum dan entitas-entitas yang terlibat di dalamnya. Dalam konteks diagram Peringkas Teks Otomatis pada Bahasa Indonesia terdapat 1 eksternal yakni pengguna, yakni orang yang menggunakan sistem. Diagram konteks ini seperti gambar 4. Pengguna memasukkan dokumen, judul dokumen, fitur dan persentase hasil ringkasan, kemudian sistem akan memberikan output berupa hasil ringkasan.

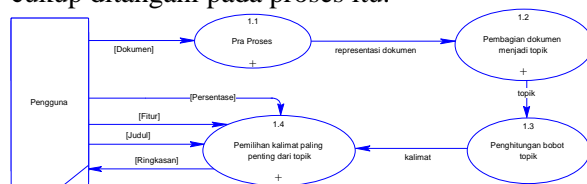


Gambar 4 Konteks diagram peringkas teks otomatis pada bahasa Indonesia

2.6.2. DFD level 1

Dari konteks diagram gambar 5, dijabarkan menjadi DFD Level 1 seperti gambar 5. Dari gambar tersebut terdapat 5 proses yakni Pra Proses, Pembagian dokumen menjadi topik, penghitungan bobot relatif topik, dan pemilihan kalimat paling penting dari topik.

Dokumen dari pengguna akan diproses pada pra proses, kemudian representasi dokumen yang dihasilkan pada proses ini akan dijadikan masukan pada proses pembagian dokumen menjadi topik. Keluaran dari proses sebelumnya yang berupa topik dijadikan sebagai masukan untuk dilakukan proses perhitungan bobot topik. Kemudian proses terakhir pada level ini adalah pemilihan kalimat paling penting dari topik. Masing-masing proses terdapat sub proses yang merupakan dekomposisi dari proses tersebut, kecuali untuk proses penghitungan bobot topik yang tidak memiliki sub proses karena sudah cukup ditangani pada proses itu.



Gambar 5 DFD level 1 peringkas teks otomatis pada bahasa Indonesia

3. HASIL DAN PEMBAHASAN

3.1. Data Penelitian

Untuk menunjang penelitian ini diperlukan data. Data yang digunakan dalam penelitian ini secara umum ada 3, yaitu dokumen, kamus dan stopword. Dokumen yang digunakan pada penelitian ini merupakan data sekunder yang dikumpulkan oleh Kurniawan (2003). Dokumen tersebut bersumber dari situs harian Duta Masyarakat Online mulai tanggal 15 Maret 2003 sampai 3 Juni 2003 sebanyak 50 dokumen yang dipilih secara acak. Dari dokumen tersebut dibuat ringkasannya oleh 3 orang sehingga terdapat 150 dokumen hasil ringkasan. Dokumen hasil ringkasan tersebut akan digunakan untuk mengevaluasi ringkasan yang dihasilkan oleh sistem pada penelitian ini. Data kamus merupakan kumpulan kata-kata dasar pada bahasa Indonesia yang disesuaikan dengan imbuhan. Data kamus yang digunakan antara lain kamus yang berhubungan dengan imbuhan partikel, milik, prefiks1, prefiks2, dan sufiks.

3.2. Ringkasan Manual

Ringkasan manual adalah ringkasan yang dibuat secara manual oleh orang. Ringkasan manual ini digunakan untuk membandingkan hasil ringkasan yang dihasilkan oleh sistem. Dalam hal ini, data yang digunakan adalah data sekunder dari penelitian yang dilakukan oleh Kurniawan (2003) di mana ringkasan manual dibuat oleh 3 orang peringkas di mana setiap orang meringkas 50 dokumen tersebut dengan hasil ringkasan sebesar 20% dari dokumen asal atau sekitar 6 kalimat [6]. Untuk menentukan kalimat hasil ringkasan yang relevan ditentukan dengan metode suara terbanyak (majority opinion), artinya bila suatu kalimat dipilih oleh minimal 2 orang peringkas maka kalimat itu relevan untuk dijadikan hasil ringkasan. Tabel 6.1 merupakan hasil rekapitulasi untuk mendapatkan kalimat relevan dari hasil ringkasan manual oleh 3 orang. Pada kolom Majority Opinion merupakan indeks kalimat relevan dari ringkasan manual yang nantinya digunakan untuk membandingkan dengan indeks kalimat yang dihasilkan oleh sistem. Sedangkan pada kolom Jml merupakan jumlah kalimat relevan yang dihasilkan hasil dari suara terbanyak. Jumlah ini nantinya digunakan sebagai pembanding untuk menentukan evaluasi recall dari tiap-tiap fitur ekstraktor. Sebagai contoh untuk dokumen dengan dok_id = 6, user1 memilih kalimat dengan indeks 0, 1, 2, 3, 4, 5 sebagai kalimat ringkasan, kemudian user2 memilih kalimat dengan indeks 0, 3, 6, 20, 23, 24

dan user3 memilih kalimat dengan indeks 0, 3, 4, 16, 18, 19. Dari hasil ringkasan dari ke-3 user ini dapat disimpulkan bahwa suara terbanyak adalah kalimat dengan indeks 0, 3, 4, 16, dengan jumlah kalimat relevan sebesar 4.

3.3. Evaluasi Sistem Berdasarkan Fitur

Evaluasi sistem dilakukan dengan cara membandingkan hasil ringkasan manual dengan hasil ringkasan oleh sistem pada tiap fitur. Untuk mengevaluasi ini menggunakan notasi perhitungan *precision*, *recall* dan *F-measures* yang ditunjukkan pada rumus 11, 12 dan 13.

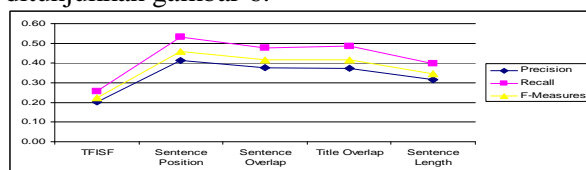
Rata-rata nilai *precision*, *recall* dan *F-measures* pada masing-masing fitur dapat dilihat pada tabel 3. Dari tabel tersebut dapat disimpulkan antara lain bahwa nilai *precision* tertinggi adalah fitur posisi kalimat sebesar 0.41. Hal ini menunjukkan bahwa fitur ini mengekstrak sebanyak sekitar 41% kalimat relevan dari jumlah kalimat yang dihasilkan oleh sistem. Dan nilai *recall* tertinggi adalah 0.53 dari fitur posisi kalimat. Hal ini berarti sebesar sekitar 53% kalimat relevan dihasilkan dari jumlah kalimat relevan.

Dari tabel 3 juga dapat disimpulkan bahwa nilai *F-measures* yang tertinggi adalah fitur posisi kalimat sebesar 0.46 sehingga dapat diartikan bahwa fitur ini paling berpengaruh terhadap hasil ringkasan yang koheren dibandingkan dengan fitur lainnya. Nilai *F-measures* tertinggi selanjutnya kemiripan dengan kalimat pertama dan kemiripan dengan judul sebesar 0.42, dan berturut-turut pada posisi 3 dan 4 fitur panjang relatif kalimat dan Log-TFISF sebesar 0.35 dan 0.22. Hal ini berarti untuk fitur Log-TFISF sedikit pengaruhnya terhadap hasil ringkasan yang koheren.

Tabel 3 Rata-rata evaluasi terhadap fitur

Fitur	Precision	Recall	F-Measures
Log-TFISF	0.20	0.26	0.22
Sentence position	0.41	0.53	0.46
Sentence overlap	0.38	0.48	0.42
Title overlap	0.37	0.49	0.42
Sentencelength	0.31	0.40	0.35

Untuk lebih jelasnya, perbandingan rata-rata nilai *precision*, *recall* dan *F-measures* tiap-tiap fitur digambarkan dalam bentuk grafis yang ditunjukkan gambar 6.



Gambar 6 Grafik perbandingan nilai *Precision*, *Recall* dan *F-Measures* masing-masing fitur

3.4. Penentuan Bobot Fitur

Pembobotan fitur dilakukan dengan cara membandingkan nilai *F-Measures* dengan total nilai *F-Measures* dari 5 fitur, sehingga akan didapatkan bobot relatif suatu fitur terhadap fitur lainnya. Rumus yang digunakan untuk mendapat bobot relatif ini ditunjukkan dengan rumus 14, di mana i adalah indeks fitur, $|I|$ adalah jumlah fitur dan $FM(i)$ adalah *F-Measure* fitur ke- i .

$$BobotRelatif(i) = \frac{FM(i)}{\sum_{i=1}^{|I|} FM(i)} \quad (14)$$

Dengan menggunakan data pada tabel 5 dan rumus 14 akan didapatkan bobot relatif untuk fitur Log-TFISF sebesar 0.12 karena nilai *F-measures* Log-TFISF 0.22 dibandingkan dengan nilai total *F-measures* sebesar 1.87. Hasil lengkap dari perhitungan bobot relatif dari fitur dapat dilihat pada tabel 5.

Dari tabel 4 dapat disimpulkan bahwa fitur posisi kalimat memiliki bobot relatif paling signifikan dibandingkan fitur lainnya yakni sebesar 0.25, kemudian berturut-turut fitur kemiripan dengan kalimat pertama sebesar 0.22, kemiripan dengan judul sebesar 0.22, panjang relatif kalimat sebesar 0.19 dan Log-TFISF 0.12.

Tabel 4 Pembobotan fitur berdasarkan nilai *F-Measures*

Fitur	F-Measures	Bobot Relatif
Log-TFISF	0.22	0.12
Sentence Position	0.46	0.25
Sentence Overlap	0.42	0.22
Title Overlap	0.42	0.22
Sentence Length	0.35	0.19
Jumlah	1.87	1.00

3.5. Analisis Fitur

Pada subbab ini akan dilakukan analisis fitur terhadap sistem peringkasan otomatis dengan cara membandingkan hasil ringkasan pada ATS dengan bobot fitur sama dengan dengan ATS dengan bobot fitur seperti tabel 6. Dokumen yang digunakan sebagai bahan ujicoba bertemakan manajemen pemasaran yang diambil dari Kompas.com pada tanggal 15 Desember 2010 sebanyak 5 dokumen. Sebagai bahan evaluasi sistem peringkasan otomatis ini, dokumen-dokumen ini diringkaskan oleh orang yang berkecimpung pada bidang manajemen pemasaran. Dokumen dilakukan peringkasan sebesar 20% dari dokumen sumber atau sekitar 6-10 kalimat. Rincian hasil ringkasan dari dokumen ini ditunjukkan tabel 5. Untuk melakukan evaluasi, sistem menggunakan peringkasan sebesar 20% juga.

Tabel 5 Dokumen sumber sebagai bahan evaluasi untuk ATS

No Dok	Jumlah Kalimat	Jumlah Ringkasan	Indeks Kalimat Ringkasan
1	39	8	1, 2, 3, 5, 6, 30, 31, 32
2	38	8	1, 2, 3, 8, 18, 29, 21, 22
3	48	10	1, 2, 5, 7, 12, 17, 20, 27, 29, 37
4	33	7	1, 2, 8, 9, 10, 23, 26
5	28	6	1, 2, 3, 5, 6, 7

3.5.1 Evaluasi ATS dengan bobot sama

Analisis dilakukan dengan cara menggunakan fitur dengan bobot sama. Hasil evaluasi ini ditunjukkan pada tabel 7. Dari tabel tersebut untuk dokumen ke-1 akurasi yang dihasilkan sistem sebesar 75,00%, dokumen ke-2 sebesar 50,00%, dokumen ke-3 sebesar 90,00%, dokumen ke-4 sebesar 57,14%, dan dokumen ke-5 sebesar 67,67%. Dari hasil akurasi masing-masing dokumen maka didapatkan rata-rata akurasi sebesar 67,76%.

Tabel 6 Evaluasi terhadap ATS tanpa pembobotan fitur

No	KR	Σ KR	KRS	Σ KRS	Akurasi
1	1, 2, 3, 5, 6, 30, 31, 32	8	1, 2, 3, 25, 30, 31, 32, 35	6	75,00%
2	1, 2, 3, 8, 18, 29, 21, 22	8	1, 2, 4, 7, 8, 22, 31, 32	4	50,00%
3	1, 2, 5, 7, 13, 17, 20, 27, 29, 37	10	1, 2, 5, 7, 13, 17, 20, 29, 31, 37	9	90,00%
4	1, 2, 8, 9, 10, 23, 26	7	2, 9, 10, 16, 18, 23, 27	4	57,14%
5	1, 2, 3, 5, 6, 7	6	1, 2, 3, 7, 18, 19	4	67,67%
	Rata-rata				67,76%

KR=Kalimat relevan

KRS = Kalimat relevan oleh sistem

3.5.2 Evaluasi ATS dengan pembobotan fitur

Analisis dilakukan dengan cara menambahkan bobot relatif fitur sesuai dengan tabel 4. Hasil evaluasi ini ditunjukkan pada tabel 7. Dari tabel tersebut untuk dokumen ke-1 akurasi yang dihasilkan sistem sebesar 75,00%, dokumen ke-2 sebesar 50,00%, dokumen ke-3 sebesar 90,00%, dokumen ke-4 sebesar 57,14%, dan dokumen ke-5 sebesar 67,67%. Dari hasil akurasi masing-masing dokumen maka didapatkan rata-rata akurasi sebesar 70,62%. Hal ini menunjukkan bahwa hasil ringkasan dari ATS yang menerapkan pembobotan fitur memiliki tingkat akurasi lebih tinggi dibandingkan tanpa menggunakan pembobotan dengan selisih sekitar 2,86%.

Tabel 7 Evaluasi terhadap ATS dengan pembobotan fitur

No	KR	Σ KR	KRS	Σ KRS	Akurasi
1	1, 2, 3, 5, 6, 30, 31, 32	8	1, 2, 3, 25, 30, 31, 32, 35	6	75,00%
2	1, 2, 3, 8, 18, 29, 21, 22	8	1, 2, 4, 7, 8, 22, 31, 32	4	50,00%
3	1, 2, 5, 7, 13, 17, 20, 27, 29, 37	10	1, 2, 5, 7, 13, 17, 20, 29, 31, 37	9	90,00%
4	1, 2, 8, 9, 10, 23, 26	7	1, 2, 9, 10, 18, 23, 27	5	71,43%
5	1, 2, 3, 5, 6, 7	6	1, 2, 3, 7, 18, 19	4	67,67%
	Rata-rata				70,62%

KR=Kalimat relevan

KRS = Kalimat relevan oleh sistem

4. KESIMPULAN

1. Nilai *precision* fitur berurutan dari yang tertinggi adalah posisi kalimat, kemiripan dengan kalimat pertama, kemiripan dengan judul, panjang relatif kalimat dan Log-TFISF dengan nilai 0.41, 0.38, 0.37, 0.31, dan 0.20. Dari data ini dapat disimpulkan bahwa fitur posisi kalimat mengekstrak kalimat relevan terbanyak dari jumlah kalimat yang dihasilkan oleh sistem yakni sekitar 41%, sedangkan fitur Log-TFISF yang paling sedikit yakni sekitar 20%.
2. Nilai *recall* fitur berurutan dari yang tertinggi adalah posisi kalimat, kemiripan dengan judul, kemiripan dengan kalimat pertama, panjang relatif kalimat, dan Log-TFISF dengan nilai 0.53, 0.49, 0.48, 0.40, dan 0.26. Dari data ini dapat disimpulkan bahwa fitur posisi kalimat mengekstrak kalimat relevan terbanyak dari jumlah kalimat relevan majority opinion yakni sekitar 53%, sedangkan fitur Log-TFISF yang paling sedikit yakni sekitar 26%.
3. Nilai *F-Measures* fitur berurutan dari yang tertinggi adalah posisi kalimat, kemiripan dengan kalimat pertama, kemiripan dengan judul, panjang relatif kalimat dan Log-TFISF dengan nilai 0.46, 0.42, 0.42, 0.35 dan 0.22. Dari data ini dapat disimpulkan bahwa fitur posisi kalimat paling berpengaruh terhadap hasil ringkasan yang koheren dibandingkan fitur lainnya. Sedangkan fitur Log-TFISF paling sedikit pengaruhnya terhadap hasil ringkasan yang koheren.
4. Bobot relatif masing-masing fitur dari yang tertinggi adalah posisi kalimat sebesar 0.25, kemiripan dengan kalimat pertama sebesar 0.22, kemiripan dengan judul sebesar 0.22, panjang relatif kalimat sebesar 0.19 dan Log-TFISF sebesar 0.12. Hal ini menunjukkan nilai bobot yang perlu digunakan dalam

proses pembobotan fitur untuk peringkasan suatu dokumen.

5. Nilai akurasi dari sistem peringkasan teks otomatis pada fitur tanpa pembobotan sebesar 67,76%, sedangkan pada fitur dengan pembobotan sesuai dengan kesimpulan nomor 4 adalah 70,62%. Hal ini menunjukkan bahwa tingkat akurasi peringkasan teks otomatis menggunakan pembobotan fitur lebih tinggi dibandingkan dengan fitur tanpa pembobotan, dengan selisih akurasi sekitar 2,86%.

5. SARAN

1. Melibatkan fitur lainnya seperti keberadaan kata benda, query oleh pengguna, dan lain-lain dalam melakukan analisis bobot fitur.
2. Perlu dilakukan teknik pembobotan fitur ekstraktor untuk menganalisis hasil ringkasan dari suatu sistem.

DAFTAR PUSTAKA

- [1] Irman, M., Prastowo, T. W., dan Nurdin, 2008, *Bahasa Indonesia 2: Untuk SMK/MAK Semua Program Keahlian Kelas XI*, Pusat Perbukuan, Departemen Pendidikan Nasional, 2008.
- [2] Hassel, M., 2004, *Summaries and the Process of Summarization from Evaluation of Automatic Text Summarization – A practical Implementation*, Licentiate Thesis, KTH NADA.
- [3] Suanmali, L., Salim N., dan Binwahlan, M. S., 2008, *Automatic Text Summarization Using Feature-Based Fuzzy Extraction*, *Jurnal Teknologi Maklumat* Jilid 20 Bil 2 p.105-115.
- [4] Mani, I., Klein G., House D., dan Hirschman L., 2002, *SUMMAC: a text summarization evaluation*, Cambridge University Press, *Natural Language Engineering* 8 (1): 43-68.
- [5] Fattah, M. A. dan Ren, F., 2008, *Automatic Text Summarization*, *World Academy of Science, Engineering and Technology* 37 p.192-195.
- [6] Kurniawan, A., 2003, *Perancangan dan Pembuatan Perangkat Lunak Peringkasan Dokumen Berita Berbahasa Indonesia dengan Metode Ekstraksi*, Laporan Tugas Akhir, Jurusan Teknik Informatika, Fakultas Teknologi Informasi, ITS, Surabaya.
- [7] Budhi, G. S., Intan, R., Silvia, R., dan Stevanus, R. R., 2007, *Indonesian Automated Text Summarization*, Information Engineering Dept. Petra Christian University, Surabaya.
- [8] Neto, J. L., Santos, A. D., Kaestner, C. A. A., dan Freitas A. A., 2003, *Generating Text Summaries through the Relative Important of Topics*, PUC-PR, Rua Imaculada Conceicao, Curitiba, Brazil.
- [9] Tala, F. Z., 2003, *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*, Institute for Logic, Language and Computation, Universiteit van Amsterdam; Amsterdam; Netherlands.
- [10] Chang, T. H., dan Lee, H. L., 2007, *Topic segmentation for short texts*, Department of Computer and Information Science, National Chiao Tung University.
- [11] Wibisono, Y., dan Khodra, M. L., 2006, *Clustering Berita Berbahasa Indonesia*, FPMIPA UPI, Sekolah Teknik Elektro dan Informatika ITB, Bandung.
- [12] Zaman, B., 2005, *Pengembangan Perangkat Lunak Peringkasan Dokumen Berbahasa Indonesia dengan Metode Hybrid Stemming*, Laporan Tugas Akhir, Jurusan Teknik Informatika, Fakultas Teknologi Informasi, ITS, Surabaya.