

BAB 3 PERANCANGAN

Pada bab ini akan dijelaskan mengenai rancangan percobaan pada penelitian segmentasi dokumen ini. Pembahasan akan dimulai dengan penjelasan mengenai gambaran umum proses segmentasi yang dilakukan, lalu akan dijelaskan mengenai data yang digunakan pada percobaan beserta pengolahan yang dilakukan sebelum data tersebut dapat digunakan sebagai data percobaan. Selanjutnya akan dijelaskan mengenai metode-metode segmentasi dokumen yang digunakan pada percobaan ini beserta proses pengukuran hasil segmentasi.

3.1 GAMBARAN UMUM PROSES SEGMENTASI DOKUMEN

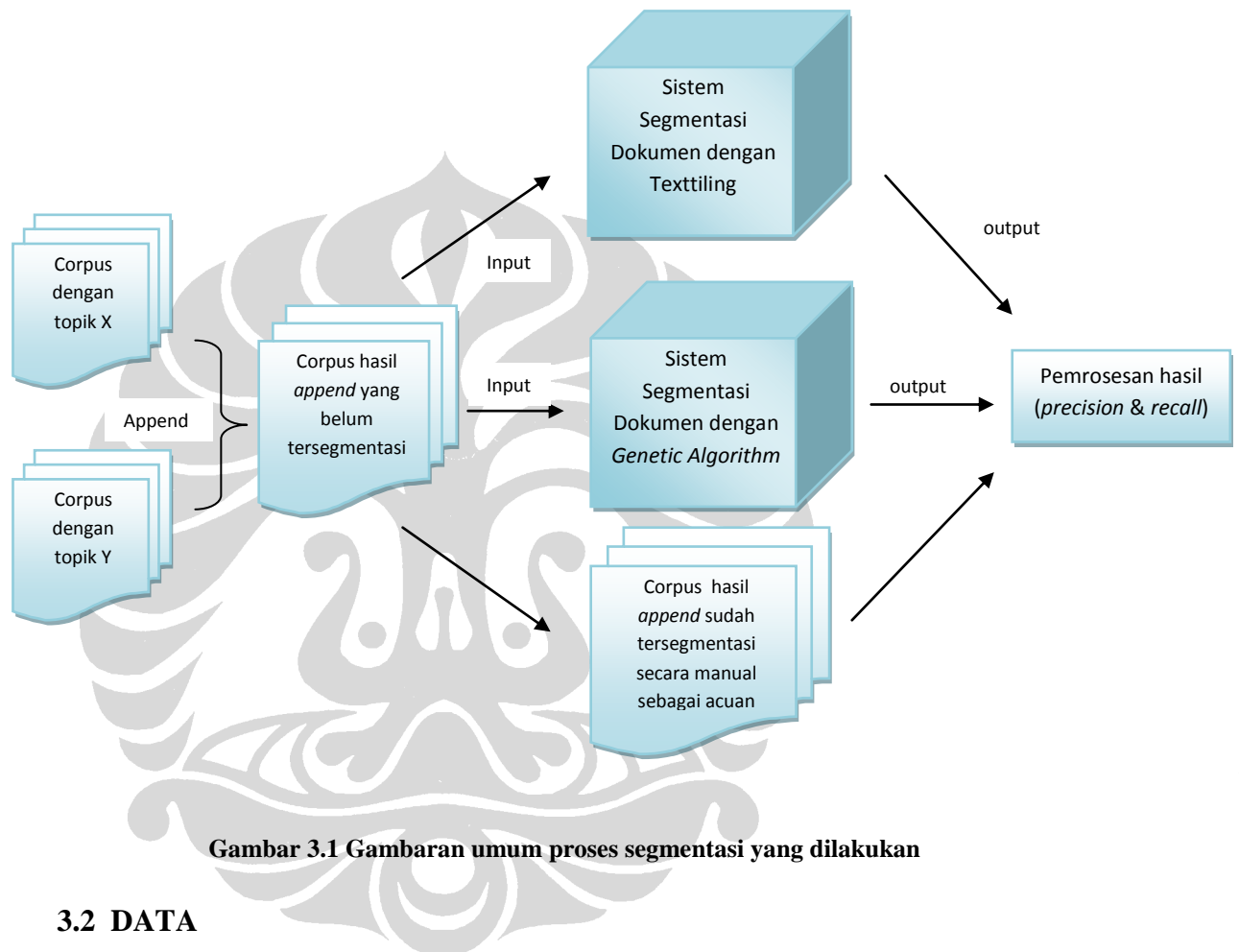
Pada tugas akhir ini, segmentasi dokumen akan dilakukan dengan pendekatan *local search* dengan menggunakan metode *genetic algorithm* yang telah dijelaskan pada subbab 2.2. Pada penelitian SegGen (Lamprier et al., 2007), dilakukan percobaan untuk membandingkan antara metode *genetic algorithm* dengan beberapa metode segmentasi dokumen lainnya. Hasil dari penelitian ini membuktikan bahwa *genetic algorithm* memberikan hasil yang baik dibandingkan dengan metode-metode lainnya. Pada tugas akhir ini akan dilakukan percobaan segmentasi dokumen dengan menggunakan metode *genetic algorithm* terhadap dokumen bahasa Indonesia. Pada percobaan ini akan dilihat tingkat akurasi dan ketepatan dari segmen pada dokumen bahasa Indonesia.

Pada percobaan dengan *genetic algorithm* ini digunakan beberapa *fitness function* untuk membantu menentukan batas segmen, yaitu optimisasi *internal cohesion*, optimisasi *dissimilarity*, optimisasi *internal cohesion* dan *dissimilarity* dengan pendekatan kombinasi linear, dan optimisasi *internal cohesion* dan *dissimilarity* dengan pendekatan SPEA 2. Proses penghitungan nilai *internal cohesion* dan *dissimilarity* dilakukan menggunakan beberapa metode penghitungan nilai *similarity* antar kalimat, diantaranya *dice coefficient*, *cosine similarity* dengan pembobotan TF-IDF, dan *cosine similarity* dengan pembobotan TF-IDF (Word Frequency). Penjelasan lebih detail mengenai metode segmentasi dokumen beserta variabelnya akan dibahas pada subbab 3.3.

Koleksi dokumen yang digunakan pada percobaan ini adalah koleksi dokumen yang telah digunakan pada penelitian tugas akhir lain, yaitu tugas akhir yang dilakukan oleh Dyta Anggraini dengan judul “Klasifikasi Topik Menggunakan Metode Naive Bayes dan Maximum Entropy Pada Artikel Media Massa dan Abstrak Tulisan” (Anggraeni, 2008). Koleksi dokumen tersebut terdiri dari artikel media massa Indonesia dan abstrak tulisan ilmiah dari beberapa kategori topik. Sebelum dokumen-dokumen tersebut digunakan pada tahap percobaan, terlebih dahulu dilakukan proses pembuangan *stopword*, *stemming*, dan pemisahan antar kalimat. Proses tersebut dilakukan untuk membantu mengoptimalkan hasil segmentasi dengan *genetic algorithm*. Pada kumpulan dokumen tersebut akan dilakukan proses *append* untuk menghasilkan suatu dokumen yang terdiri dari beberapa dokumen sebagai penyusunnya. Proses segmentasi dengan metode *genetic algorithm* ini akan dilakukan pada dokumen-dokumen hasil proses *append* tersebut. Untuk penjelasan lebih detil mengenai proses pengolahan data percobaan dan pembuatan *test ase* akan dijelaskan pada subbab 3.2. Pada percobaan terakhir akan dilakukan percobaan untuk membandingkan hasil segmentasi menggunakan metode *genetic algorithm* dengan hasil segmentasi menggunakan metode Texttiling yang disediakan pada *website* <http://morphadorner.northwestern.edu>. Penjelasan lebih detil mengenai perbandingan antara metode *genetic algorithm* dan Texttiling akan dibahas pada subbab 3.4. Proses segmentasi ini bertujuan untuk menemukan batas-batas antar dokumen yang menyusun dokumen hasil *append* tersebut. Letak segmentasi sebenarnya dari dokumen hasil *append* tersebut akan disimpan sebagai acuan penilaian hasil segmentasi oleh *genetic algorithm*. Untuk setiap percobaan akan digunakan lebih dari satu jenis *test case* yang berbeda.

Proses segmentasi ini akan menghasilkan konfigurasi letak segmen untuk dokumen yang menjadi *input* proses ini. Tahap penilaian hasil segmentasi pada percobaan ini akan dilakukan dengan melakukan penghitungan *precision* dan *recall* dari konfigurasi segmen tersebut dengan mengacu pada letak segmen sebenarnya yang telah disimpan pada proses sebelumnya. *Genetic algorithm* pada dasarnya menggunakan metode *random*, oleh karena itu percobaan untuk masing-masing dokumen akan dilakukan sebanyak lebih dari satu kali, untuk kemudian

dilakukan penghitungan rata-rata dari nilai *precision* dan *recall* untuk masing-masing dokumen tersebut sebagai hasil evaluasi. Hal ini dilakukan untuk menguji ketepatan dan keakuratan hasil segmentasi yang telah dilakukan. Gambaran umum mengenai keseluruhan proses segmentasi dokumen dengan menggunakan metode *genetic algorithm* ini dapat dilihat pada Gambar 3.1 sebagai berikut:



Gambar 3.1 Gambaran umum proses segmentasi yang dilakukan

3.2 DATA

Pada penelitian segmentasi dokumen bahasa Indonesia ini, akan digunakan kumpulan dokumen dari dua domain yang cukup berbeda, yaitu artikel media massa dan abstrak tulisan ilmiah. Topik atau tema dari dokumen-dokumen tersebut sudah diketahui sebelumnya, sehingga dapat memudahkan pemilihan topik dokumen yang akan digunakan pada percobaan. Kedua jenis data ini akan dikombinasikan untuk menciptakan berbagai dokumen hasil *append* yang akan digunakan pada percobaan (*test case*).

Keseluruhan dokumen yang digunakan pada percobaan ini berasal dari dokumen yang telah digunakan pada penelitian sebelumnya (Anggraeni, 2008). Data artikel media massa Indonesia diperoleh dari www.kompas.com, sedangkan data abstrak tulisan ilmiah diperoleh dari basis data sistem Lontar yang dimiliki oleh perpustakaan Fakultas Ilmu Komputer Universitas Indonesia. Klasifikasi jenis topik untuk dokumen-dokumen artikel media massa tersebut mengikuti klasifikasi yang sudah dilakukan pada website sumber dokumen, dan untuk klasifikasi jenis topik untuk dokumen-dokumen abstrak tulisan ilmiah telah dilakukan oleh Dyta Anggraini. Domain topik untuk data artikel media massa terdiri dari ekonomi, kesehatan, olahraga, properti, dan travel, sedangkan domain topik pada data abstrak tulisan ilmiah terdiri dari *information retrieval* (IR), pengolahan citra, dan rekayasa perangkat lunak (RPL). Jumlah total koleksi untuk dokumen artikel media massa adalah berkisar 1500 dokumen, dan untuk dokumen abstrak tulisan ilmiah berjumlah sekitar 350 dokumen.

3.2.1 PENGOLAHAN DATA

Seperti telah dijelaskan pada subbab 3.1, dokumen yang digunakan pada percobaan ini adalah dokumen hasil *append* dari beberapa dokumen. Oleh karena itu kumpulan dokumen artikel media massa dan abstrak tulisan ilmiah tersebut akan digunakan sebagai bahan penyusun dokumen hasil *append*. Proses pemilihan dokumen yang akan digunakan sebagai dokumen penyusun dilakukan secara *random* berdasarkan domain yang akan digunakan, lebih jelas mengenai hal ini akan dijelaskan pada subbab 3.2.2. Bersamaan dengan tahap pembuatan dokumen hasil *append* ini, dilakukan pula pembuatan dokumen yang sudah terdefinisi letak segmennya (kunci) sebagai acuan penilaian hasil proses segmentasi dengan *genetic algorithm*.

Sebelum koleksi dokumen hasil *append* tersebut dapat digunakan dalam percobaan ini, perlu dilakukan beberapa pemrosesan terhadap data-data tersebut. Beberapa proses yang dilakukan sebagai bentuk persiapan data sebelum digunakan pada percobaan ini diantaranya adalah melakukan pembuangan *stopword*, melakukan *stemming*, dan melakukan pemisahan antar kalimat. Proses pembuangan *stopword* dilakukan agar kata-kata yang sangat umum digunakan

tidak dilibatkan dalam proses segmentasi, dikarenakan *stopword* tidak memiliki karakteristik sebagai penentu topik dari suatu dokumen. Beberapa contoh kata-kata yang termasuk dalam *stopword* adalah “yang”, “di”, “dan”, “itu”, “dengan”, dsb. *Stemming* adalah suatu proses untuk membentuk suatu kata menjadi bentuk dasar dari kata tersebut (kata dasar). Proses *stemming* dilakukan dengan menggunakan *stemmer* yang telah dikembangkan oleh Muhammad Ichsan dan Mirna Adriani (Ichsan & Adriani, 2006). Proses *stemming* ini dilakukan agar kata-kata yang berbeda (karena penambahan imbuhan/akhiran/partikel) namun memiliki makna terkait tidak dianggap berbeda pada proses segmentasi. Contoh manfaat penggunaan proses *stemming* misalnya antara kata “memakan” dan “makan”, kedua kata tersebut memiliki makna yang relatif sama, tetapi tanpa adanya proses *stemming* maka kedua kata tersebut akan dianggap dua buah kata yang sama sekali berbeda, namun dengan adanya proses *stemming* maka kata “memakan” akan direduksi menjadi “makan” sedangkan kata “makan” tidak akan direduksi karena sudah merupakan kata dasar, sehingga pada tahap pemrosesan kata, kedua kata tersebut akan dianggap kata yang sama. Proses pemisahan kalimat dilakukan untuk membuat setiap kalimat penyusun dokumen tersebut menempati satu baris pada dokumen tersebut (jika terdapat suatu dokumen dengan tiga kalimat, maka dokumen tersebut akan diubah menjadi dokumen yang terdiri dari tiga baris dimana setiap baris berisi satu kalimat). Pemisahan kalimat tersebut dilakukan untuk mempermudah representasi input dan output pada proses segmentasi dokumen. Penjelasan lebih detil mengenai proses ini akan dijelaskan pada subbab 4.1.

3.2.2 PEMBUATAN TEST CASE

Pada percobaan ini dilakukan percobaan-percobaan dengan berbagai variasi data yang digunakan. Tujuan melakukan variasi pada jenis data ini adalah untuk mengetahui kualitas suatu metode terhadap berbagai jenis data. Variasi data yang akan digunakan pada percobaan ini terdiri dari dua variasi, yaitu variasi jumlah segmen dan variasi kemiripan antar topik penyusun dokumen. Kedua variabel ini dipilih untuk mewakili dua permasalahan mendasar, yaitu seberapa akurat segmentasi yang dilakukan untuk dokumen dengan jumlah segmen yang

banyak dan seberapa akurat segmentasi yang dilakukan ketika dihadapkan pada topik-topik penyusun yang cukup mirip.

Pada variabel jumlah segmen, akan diterapkan empat buah variasi jumlah segmen yaitu 2 buah segmen (3 dokumen penyusun), 5 buah segmen (6 dokumen penyusun), 10 buah segmen (11 dokumen penyusun), 30 buah segmen (31 dokumen penyusun). Pada variabel kemiripan antar topik akan digunakan variasi dokumen penyusun dari domain yang tidak mirip, domain yang mirip, dan domain yang sangat mirip. Tingkat kemiripan suatu dokumen tersebut dilihat berdasarkan tipe dokumen (artikel atau abstrak) dan topik dokumen. Untuk variasi dokumen yang tidak mirip, dokumen-dokumen penyusun akan dipilih secara *random* dari seluruh koleksi dokumen yang dimiliki. Untuk variasi dokumen yang mirip, dokumen-dokumen penyusun akan dipilih secara *random* dari koleksi dokumen abstrak dimana keseluruhan dokumen bertemakan bidang-bidang ilmu komputer. Untuk variasi dokumen yang sangat mirip, dokumen-dokumen penyusun akan dipilih secara *random* dari koleksi dokumen yang termasuk kedalam salah satu topik saja, pada percobaan ini digunakan kumpulan dokumen abstrak dengan topik rekayasa perangkat lunak (RPL). Contoh *test case* dapat dilihat pada Lampiran 2, Lampiran 3, dan Lampiran 4.

3.3 SEGMENTASI DOKUMEN DENGAN GENETIC ALGORITHM

Pada percobaan ini akan digunakan *genetic algorithm* sebagai metode untuk melakukan segmentasi dokumen. Beberapa pendekatan yang akan digunakan pada *genetic algorithm* untuk melakukan segmentasi dokumen ini diantaranya adalah optimisasi *internal cohesion*, optimisasi *dissimilarity*, optimisasi *internal cohesion* dan *dissimilarity* dengan kombinasi linear, dan optimisasi *internal cohesion* dan *dissimilarity* dengan pendekatan SPEA 2. Pendekatan yang dilakukan pada percobaan ini cukup berbeda dengan pendekatan yang dilakukan pada penelitian SegGen (Lamprier et al., 2007), dimana pada penelitian SegGen tersebut hanya digunakan pendekatan SPEA.

Seperti telah dijelaskan pada subbab 2.2.1 mengenai penelitian SegGen, fungsi *internal cohesion* dan fungsi *dissimilarity* merupakan dua fungsi yang akan dioptimalkan untuk menghasilkan segmentasi yang tepat. Fungsi *internal*

cohesion akan mengoptimalkan keterkaitan antar kalimat pembentuk suatu segmen, sedangkan fungsi *dissimilarity* akan mengoptimalkan ketidakterhubungan antara dua segmen yang berdekatan. Pada umumnya suatu metode segmentasi dokumen hanya mempertimbangkan fungsi *internal cohesion* untuk melakukan segmentasi, namun sebenarnya fungsi *dissimilarity* ini akan membantu meningkatkan ketepatan hasil segmentasi dengan melihat hubungan antar segmen-segmen yang ada. Memandang permasalahan segmentasi dokumen ini sebagai permasalahan untuk mengoptimalkan fungsi *internal cohesion* dan fungsi *dissimilarity* yang sifatnya saling bertolak belakang (semakin banyak segmen maka fungsi *internal cohesion* akan semakin meningkat, namun fungsi *dissimilarity* akan semakin menurun), maka permasalahan ini dapat dikategorikan sebagai *multiobjective problem*.

Untuk menemukan solusi dari suatu *multiobjective problem*, terdapat beberapa pendekatan yang dapat dilakukan. Dua pendekatan yang digunakan pada percobaan ini adalah kombinasi linear dan SPEA 2. Kedua pendekatan tersebut sudah sangat lazim digunakan untuk pemecahan solusi *multiobjective problem*. Solusi dari *multiobjective problem* biasanya bukan merupakan solusi tunggal, melainkan himpunan solusi. Dengan mengoptimisasi dua fungsi tersebut dan mencari solusi optimal dengan pendekatan kombinasi linear atau SPEA 2 diharapkan dapat menghasilkan segmentasi yang baik.

3.3.1 PARAMETER GENETIC ALGORITHM

Pada dasarnya metode *genetic algorithm* terdiri dari enam komponen utama, yaitu individu, populasi, *fitness function*, seleksi, *crossover*, dan mutasi. Dengan mendefinisikan permasalahan segmentasi dokumen ini menjadi enam komponen tersebut, maka implementasi *genetic algorithm* untuk proses segmentasi dokumen ini sudah dapat dilakukan. Representasi permasalahan yang baik dan benar akan sangat membantu untuk mempermudah proses pencarian solusi. Beberapa representasi permasalahan segmentasi dokumen pada *genetic algorithm* pada percobaan ini mengacu pada penelitian SegGen (Lamprier et al., 2007).

Representasi individu yang digunakan pada percobaan ini adalah suatu *binary string*, dimana untuk dokumen dengan ns kalimat akan direpresentasikan dengan vektor biner \vec{x} dengan $(ns - 1)$ elemen. $x_i = 1$ menunjukkan bahwa terdapat segmen diantara kalimat ke i dan ke $i + 1$. Contoh representasi individu yang digunakan pada percobaan dapat dilihat pada Gambar 3.2 sebagai berikut:

Dokumen:

Pagi hari ibu pergi kepasar bersama adik
 Ketika ibu pergi ayah sedang minum kopi
 Presiden Indonesia sekarang adalah SBY
 SBY merupakan presiden ke enam Indonesia

Contoh Representasi Individu : 0 0 1

Penjelasan :

Terdapat segmen diantara kalimat ke 3 dan ke 4 pada dokumen tersebut, sehingga dokumen hasil segmentasi adalah sebagai berikut:

pagi hari ibu pergi ke pasar bersama adik
 ketika ibu pergi ayah sedang minum kopi
 presiden Indonesia sekarang adalah sby

 sby merupakan presiden keenam Indonesia

Gambar 3.2 Contoh representasi individu/*chromosome* yang digunakan

Populasi awal dipilih secara *random* untuk menjamin keragaman individu-individu pada populasi. Ukuran populasi yang digunakan merupakan suatu variabel percobaan yang nantinya akan menjadi bahan uji coba. *Fitness function* yang digunakan adalah optimisasi fungsi *internal cohesion* (persamaan 2.5), optimisasi fungsi *dissimilarity* (persamaan 2.7), optimisasi fungsi *internal cohesion* dan *dissimilarity* dengan kombinasi linear (persamaan 2.10), dan optimisasi fungsi *internal cohesion* dan *dissimilarity* dengan SPEA 2. Untuk proses segmentasi dokumen dengan menggunakan pendekatan SPEA 2, proses seleksi konfigurasi hasil segmentasi pada iterasi terakhir akan digunakan metode

kombinasi linear untuk mencari konfigurasi akhir yang terbaik berdasarkan kedua fungsi objektifnya. Proses seleksi pada *genetic algorithm* dilakukan dengan menggunakan metode seleksi *tournament selection*, proses mutasi dilakukan dengan mengganti individu lama dengan suatu individu baru secara *random*, dan proses *crossover* dilakukan dengan menggunakan metode *single point crossover* dan *two point crossover* (variabel percobaan).

3.3.2 VARIABEL EKSPERIMENTASI

Pada percobaan ini akan dilakukan percobaan-percobaan dengan variasi pada beberapa variabel komponen *genetic algorithm*. Tujuan melakukan variasi pada variabel komponen *genetic algorithm* ini adalah untuk mengetahui konfigurasi komponen *genetic algorithm* yang paling optimal untuk menghasilkan segmentasi yang akurat. Variabel komponen yang digunakan pada percobaan ini terdiri dari lima buah variabel, yaitu *fitness function*, probabilitas mutasi, tipe *crossover*, jumlah iterasi, dan ukuran populasi.

Pada variabel *fitness function* akan diterapkan beberapa fungsi yang akan digunakan sebagai fungsi pencarian solusi, diantaranya optimisasi *internal cohesion* (persamaan 2.5), optimisasi *dissimilarity* (persamaan 2.7), optimisasi *internal cohesion* dan *dissimilarity* dengan kombinasi linear, dan optimisasi *internal cohesion* dan *dissimilarity* dengan SPEA 2. Persamaan kombinasi linear dan proses ekstraksi konfigurasi akhir dari populasi terakhir pada SPEA 2 mengacu pada persamaan 2.10 dengan $\alpha \in 1, 2, 6, 8$ untuk *fitness function* kombinasi linear dan $\alpha \in 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 100$ untuk *fitness function* SPEA 2. Metode penghitungan nilai *similarity* pada proses penghitungan nilai *internal cohesion* dan nilai *dissimilarity* juga merupakan suatu variabel percobaan, dimana terdapat 3 metode yang diujikan, yaitu *dice coefficient*, *cosine similarity* dengan pembobotan kata TF-IDF, *cosine similarity* dengan pembobotan kata TF-IDF (Word Frequency). Berikut adalah proses penghitungan *similarity* dengan metode *cosine similarity* seperti yang digunakan pada penelitian (Lamprier, 2007):

$$Sim(s_1, s_2) = \frac{\sum_{i=1}^t w_{i,s_1} \times w_{i,s_2}}{\sqrt{\sum_{i=1}^t w_{i,s_1}^2 \times \sum_{i=1}^t w_{i,s_2}^2}} \quad (3.1)$$

dimana t merupakan jumlah kata bermakna, dan w_{i,s_j} adalah bobot kata bermakna i pada kalimat s_j . Berikut adalah proses penghitungan *similarity* dengan menggunakan *dice coefficient*¹ :

$$Sim(s_1, s_2) = \frac{2 \times \text{common terms}}{|s_1| + |s_2|} \quad (3.2)$$

Berikut adalah proses penghitungan bobot kata pada metode pembobotan TF-IDF:

$$\text{Bobot}(w_i) = TF(w_i) \times \frac{1}{DF(w_i)} \quad (3.3)$$

Dimana w_i adalah bobot kata i pada dokumen, $TF(w_i)$ adalah frekuensi kata i pada dokumen, $DF(w_i)$ adalah jumlah kata i pada seluruh dokumen. Rumus ini diperoleh berdasarkan pengertian harafiah dari proses pembobotan dimana untuk menilai bobot kata i pada satu dokumen dilakukan perbandingan antara jumlah kata i pada dokumen tersebut dengan jumlah kata i pada keseluruhan koleksi dokumen. Berikut adalah proses penghitungan bobot kata pada metode pembobotan TF-IDF (*Word Frequency*) (Siahaan, 2008):

$$\begin{aligned} \text{Bobot}(w_i) &= TF(w_i) \times IDF_{wf}(w_i) \\ &= TF(w_i) \times \text{Log} \frac{TF(w)}{TF(w_i)} \end{aligned} \quad (3.4)$$

Dimana w_i adalah bobot kata i pada dokumen, $TF(w_i)$ adalah frekuensi kata i pada dokumen, $TF(w)$ adalah frekuensi seluruh kata pada dokumen.

Pada variabel probabilitas mutasi diterapkan 3 buah variasi probabilitas mutasi, yaitu 0.05, 0.09, 0.2. Pada variabel tipe *crossover* diterapkan 2 buah variasi, yaitu *single point crossover* dan *two point crossover*. Pada variabel jumlah iterasi akan diterapkan tiga buah variasi, yaitu 100 iterasi, 500 iterasi, dan 1000 iterasi. Pada variabel ukuran populasi akan diterapkan empat buah variasi, yaitu 20 individu, 30 individu, 40 individu, dan 50 individu. Pemilihan jumlah iterasi dan ukuran populasi ini dirasakan sudah cukup mewakili rentang nilai yang bervariasi (dari kecil sampai besar).

¹ Lihat <http://www.dcs.shef.ac.uk/~sam/stringmetrics.html#dice>

3.4 PERBANDINGAN ANTARA METODE GENETIC ALGORITHM DAN METODE TEXTTILING

Pada percobaan ini akan dibandingkan hasil segmentasi antara metode *genetic algorithm* dan metode Texttiling. Metode Texttiling yang digunakan telah disediakan oleh *website* <http://morphadorner.northwestern.edu>. Metode Texttiling yang disediakan pada *website* tersebut membutuhkan 3 *input* sebagai parameternya, yaitu teks dokumen yang akan disegmen, *window size*, dan *segment size*. *Window size* merupakan banyaknya kata yang akan digunakan sebagai *token sequences* pada metode Texttiling. *Segment size* merupakan jumlah kisaran kalimat yang harus terdapat pada satu segmen. Variabel pada percobaan ini adalah *window size* dan *segment size*, dimana akan diujikan percobaan dengan *window size* sebesar 20 kata (sesuai referensi Hearst untuk ukuran *token sequences* (Manning, 1999) dan *segment size* bervariasi sebesar 10, 20, dan 50. Pada percobaan ini akan digunakan data percobaan yang sama untuk kedua metode, penjelasan lebih detil akan dijelaskan pada subbab 5.9.

3.5 METODE PENGUKURAN TINGKAT AKURASI

Untuk mengukur tingkat akurasi hasil segmentasi dengan *genetic algorithm*, digunakan dua buah metode pengukuran tingkat akurasi, yaitu *precision* dan *recall*. *Precision* dan *recall* merupakan dua metode pengukuran akurasi yang sudah lazim digunakan dibidang *information retrieval*. Kedua metode ini akan menguji seberapa akurat konfigurasi segmen yang dihasilkan dibandingkan dengan konfigurasi segmen yang seharusnya. Berikut adalah cara penghitungan *precision* dan *recall* terhadap suatu konfigurasi segmen (Lamprier et al., 2007):

$$precision = \frac{\text{jumlah segmen benar yang ditemukan}}{\text{jumlah segmen yang ditemukan}} \quad (3.5)$$

$$recall = \frac{\text{jumlah segmen benar yang ditemukan}}{\text{jumlah segmen yang seharusnya}} \quad (3.6)$$

Metode pengukuran akurasi *precision* dan *recall* ini tidak dapat mengukur seberapa jauh perbedaan antara konfigurasi segmen yang dihasilkan dengan konfigurasi segmen yang seharusnya, tetapi dapat mengukur seberapa besar akurasi hasil segmentasi (*precision*) dan menguji apakah segmen yang seharusnya telah ditemukan atau tidak. Pemilihan dua metode tersebut diharapkan dapat menguji ketepatan dan kebenaran hasil segmentasi.

