BAB 2
LANDASAN TEORI

Pada bab ini akan dijelaskan mengenai landasan teori dan metode-metode yang digunakan pada penelitian mengenai segmentasi dokumen ini. Pembahasan akan dimulai dengan penjelasan mengenai segmentasi dokumen itu sendiri, penjelasan mengenai *topic detection and tracking* yang merupakan salah satu aplikasi yang menggunakan proses segmentasi dokumen dan penjelasan mengenai metode segmentasi Texttiling. Selanjutnya akan dijelaskan mengenai metode *genetic algorithm* (beserta hal-hal yang terkait) dan penelitian segmentasi dokumen menggunakan metode *genetic algorithm* yang pernah dilakukan.

2.1 SEGMENTASI DOKUMEN

Segmentasi dokumen merupakan suatu kegiatan membagi-bagi suatu dokumen menjadi bagian-bagian yang homogen atau memiliki keterkaitan yang tinggi (Lamprier et al., 2007). Segmentasi dokumen ini dapat diterapkan pada berbagai macam jenis data dan kebutuhan, misalnya segmentasi dokumen berdasarkan topik-topik penyusunnya jika dokumen tersebut terdiri dari berbagai topik yang belum tersegmentasi dengan jelas baik untuk dokumen yang masih memiliki tanda baca maupun dokumen tanpa tanda baca (dokumen hanya mengandung teks). Contoh aplikasi yang memanfaatkan proses segmentasi dokumen adalah aplikasi TDT yang akan dibahas pada subbab 2.1.1. Contoh dokumen yang terdiri dari beberapa topik namun belum tersegmen secara jelas terdapat pada Gambar 2.1 sebagai berikut:

Paul McMaster yang bermain untuk klub Knox Taiders di liga bola basket Australia Tenggara, kepergok membawa obat-obatan tersebut pada 1 Februari lalu. Ia dihentikan oleh petugas bea cukai Australia saat kembali dari Thailand.Menurut pihak ASADA, McMaster kedapatan membawa sejumlah tablet yang dikategorikan sebagai dua jenis anabolic steroid. Presiden AS George W Bush akan mengumumkan langkah-langkah strategis untuk memperkuat sistem keuangan dan memulihkan kepercayaan pasar, Selasa (14/10) waktu setempat atau Rabu WIB.Berbekal dana talangan (bailout) senilai 700 miliar dollar AS, Departemen Keuangan AS akan menginvestasikan dana 250 miliar dollar ke sejumlah bank. Menurut sumber di Gedung Putih dan dilaporkan CNN, sebagai langkah awal sembilan bank akan mendapat kucuran dana segar dari pemerintah. Bush juga akan mengalokasikan dana lain sebesar 100 miliar dollar AS untuk departemen keuangan. Kota kuno ini begitu indah. Saya sampai lupa mengedipkan mata menyaksikan pagoda Nyatapola yang menjulang megah. Barisan kuil dan istana kuno, gang sempit yang meliuk-liuk, melemparkan saya kembali ke masa silam. Di antara ketiga kota kuno di Lembah Kathmandu, bagi saya Bhaktapur adalah yang paling indah. Di sini kita terbebas dari ingar bingar kendaraan bermotor dan pengaruh buruk turisme yang merambah Kathmandu.

Gambar 2.1 Contoh dokumen yang belum tersegmen

Dari contoh Gambar 2.1, terlihat bahwa tanpa adanya segmentasi yang jelas maka akan sulit untuk menentukan batas akhir dari suatu topik pembahasan. Hasil segmentasi dokumen pada Gambar 2.1 dapat dilihat pada Gambar 2.2. Dengan adanya batas segmen tersebut, pembaca dapat dengan mudah mengetahui akhir pembahasan dari suatu topik, sehingga tidak akan menimbulkan kebingungan.

Paul McMaster yang bermain untuk klub Knox Taiders di liga bola basket Australia Tenggara, kepergok membawa obat-obatan tersebut pada 1 Februari lalu. Ia dihentikan oleh petugas bea cukai Australia saat kembali dari Thailand.Menurut pihak ASADA, McMaster kedapatan membawa sejumlah tablet yang dikategorikan sebagai dua jenis anabolic steroid. **<BATAS SEGMENTASI>** Presiden AS George W Bush akan mengumumkan langkah-langkah strategis untuk memperkuat sistem keuangan dan memulihkan kepercayaan pasar, Selasa (14/10) waktu setempat atau Rabu WIB.Berbekal dana talangan (bailout) senilai 700 miliar dollar AS, Departemen Keuangan AS akan menginvestasikan dana 250 miliar dollar ke sejumlah bank. Menurut sumber di Gedung Putih dan dilaporkan CNN, sebagai langkah awal sembilan bank akan mendapat kucuran dana segar dari pemerintah. Bush juga akan mengalokasikan dana lain sebesar 100 miliar dollar AS untuk departemen keuangan. **<BATAS SEGMENTASI>** Kota kuno ini begitu indah. Saya sampai lupa mengedipkan mata menyaksikan pagoda Nyatapola yang menjulang megah. Barisan kuil dan istana kuno, gang sempit yang meliuk-liuk, melemparkan saya kembali ke masa silam. Di antara ketiga kota kuno di Lembah Kathmandu, bagi saya Bhaktapur adalah yang paling indah. Di sini kita terbebas dari ingar bingar kendaraan bermotor dan pengaruh buruk turisme yang merambah Kathmandu.

Gambar 2.2 Contoh dokumen yang sudah tersegmen

2.1.1 TOPIC DETECTION AND TRACKING (TDT)

Topic detection and tracking (Allan, 2002) merupakan suatu penelitian yang berkaitan dengan pengolahan dokumen suara yang bertujuan untuk memonitor peristiwa-peristiwa yang terjadi di dunia melalui siaran berita. Sistem TDT ini akan menangkap pemberitaan mengenai peristiwa yang disiarkan melalui tv ataupun radio, lalu memberitahukan jika ada peristiwa baru atau menarik yang terjadi. Secara umum sistem TDT ini akan menerima rekaman berita suara (dokumen suara) lalu mengubah berita suara tersebut menjadi dokumen-dokumen teks yang masing-masing berisikan satu berita hasil transformasi.

Ada beberapa task yang perlu dilakukan oleh sistem TDT ini untuk mendukung keseluruhan proses dari TDT ini, diantaranya (Allan, 2002):

1. *Story segmentation*

   Melakukan proses pemisahan berita dari dokumen suara menjadi satuan berita-berita yang terpisah.

2. *First Story Detection*

   Melakukan deteksi pada berita jika berita tersebut merupakan berita baru yang muncul.

3. *Cluster Detection*

   Melakukan pengelompokan berita berdasarkan topiknya masing-masing.

4. *Tracking*

   Melakukan pencarian berita-berita tambahan yang berkaitan dengan suatu topik.

5. *Story link detection*

   Melakukan pemeriksaan terhadap dua dokumen jika kedua dokumen tersebut memiliki keterkaitan.

Dengan mengoptimalisasi semua *task* yang ada pada TDT, maka akan dihasilkan suatu sistem pencarian berita yang optimal.

2.1.2 TEXTTILING

Texttiling merupakan suatu metode untuk membagi suatu dokumen menjadi bagian-bagian tertentu, yang pada awalnya diperkenalkan oleh Marti A. Hearst. Secara umum, Texttiling merupakan salah satu metode untuk melakukan segmentasi dokumen. Metode Texttiling ini terdiri dari 3 tahap, yaitu (Hearst, 1997):

1. *Tokenization*
2. *Lexical Score Determination*
3. *Boundary Identification*

Pada tahap *tokenization* dilakukan proses pemeriksaan pada setiap kata yang terdapat pada dokumen. Proses ini dilakukan agar kata-kata yang akan dipergunakan pada proses selanjutnya merupakan kata-kata yang sudah seragam dan memiliki makna pada dokumen tersebut. Pada tahap ini juga dilakukan proses

pemilihan kata-kata yang tidak termasuk dalam kelompok kata *stopwords* (kata-kata yang hampir selalu muncul disemua jenis dokumen, sehingga kurang menggambarkan karakteristik dari dokumen). Kata-kata yang tidak termasuk kedalam kelompok kata *stopword* tersebut kemudian dikelompokan menjadi suatu kalimat semu dengan ukuran tertentu. Kalimat semu ini kemudian disebut juga dengan *token-sequence*s. Kalimat semu ini bertujuan untuk menciptakan suatu ukuran seragam sebelum dilakukan proses selanjutnya yaitu *score determination*.

Pada tahap *lexical score determination* ini akan dilakukan proses penilaian terhadap setiap *token-sequences* yang ada. Penilaian ini bertujuan untuk menentukan nilai kedekatan antar segmen. Terdapat dua metode yang dapat digunakan pada proses penghitungan nilai kedekatan tersebut, yaitu: metode *block comparison* dan *vocabulary introduction*. Metode b*lock comparison* menilai tingkat kedekatan dengan membandingkan jumlah kata-kata yang sama pada dua segmen/blok yang bersebelahan. Proses penghitungan tersebut dilakukan dengan cara sebagai berikut:

Diberikan 2 blok teks b_1 dan b_2, masing-masing dengan *k token sequence*, dimana $b_1 = \{tokens_{i-k}, \ldots, tokens_i\}$ dan $b_2 = \{tokens_{i+1}, \ldots, tokens_{i+k+1}\}$,

$$score(i) = \frac{\sum_t w_{t,b_1} w_{t,b_2}}{\sqrt{\sum_t w_{t,b_1}^2 \sum_t w_{t,b_2}^2}} \quad (2.1)$$

Dengan $w_{t,b}$ merupakan bobot kata t pada segmen b. Pada metode *vocabulary introduction*, tingkat kedekatan antar segmen bersebelahan dihitung berdasarkan jumlah kosa kata baru yang digunakan pada masing-masing segmen. Proses penghitungan nilai kedekatan antar segmen dengan metode *vocabulary introduction* adalah dengan menjumlahkan jumlah kata-kata baru pada kedua segmen lalu dibagi dengan jumlah keseluruhan kata dikali dua, seperti pada persamaan berikut:

Untuk setiap gap antara *token sequence* i, buat teks dengan interval b dengan panjang $w * 2$ (dimana w adalah panjang *token sequence*) diantara i, dan bagi b

menjadi dua bagian b_1 dan b_2, dimana b_1 = $\{tokens_{i-w},..., tokens_i\}$ dan b_2 = $\{tokens_{i+1},..., tokens_{i+w+1}\}$,

$$score(i) = \frac{NumNewTerms\ (b_1) + NumNewTerms\ (b_2)}{w \times 2} \quad (2.2)$$

Dengan b_1 dan b_2 merupakan dua segmen yang berdekatan, $NumNewTerms(b)$ adalah jumlah kata baru pada interval *b* dan *w* merupakan jumlah keseluruhan kata.

Pada tahap *boundary identification*, dilakukan proses penghitungan *depth score* yang merupakan nilai selisih antara nilai kedekatan diantara dua segmen yang berurutan. Semakin tinggi nilai kedekatan tersebut maka nilai *depth score* akan semakin kecil, dan juga sebaliknya.

Hasil dari segmentasi dengan menggunakan metode Texttiling ini menunjukan hasil yang cukup baik. Metode ini sudah diintergrasikan dengan suatu *user interface* pada suatu sistem *information retrieval*, dan juga telah sukses digunakan untuk melakukan segmentasi pada surat kabar berbahasa Arab yang tidak memiliki batas-batas paragraf.

2.1.3 PENELITIAN SEGMENTASI DOKUMEN TEKS BAHASA INDONESIA MENGGUNAKAN METODE TEXTTILING

Pada penelitian yang dilakukan oleh Edison Pardengganan Siahaan (Siahaan, 2008) sebagai tesis kelulusan program magister ilmu komputer di Universitas Indonesia ini, segmentasi dokumen teks bahasa indonesia dilakukan dengan menggunakan metode Texttiling berbasiskan perbandingan blok yang menggunakan berbagai variasi metode pembobotan kata. Penelitian segmentasi dokumen ini dilakukan untuk dokumen teks bahasa Indonesia hasil transkripsi dari dokumen suara. Proses transkripsi dokumen suara menjadi dokumen teks dilakukan oleh *Automatic Speech Recognition* (ASR) yang umumnya tidak memiliki batas-batas akhir topik yang jelas. Beberapa variasi metode pembobotan kata yang digunakan pada penelitian ini adalah: *TF-IDF-Mutual Information*, *TF-*

IDF(word Frequency), *TF-IDF-Mutual Information Word Similarity*, *TF-IDF*, dan *Latent Semantic Analysis*.

Korpus yang digunakan pada penelitian ini terdiri dari berita teks dan berita suara televisi. Koleksi berita suara terdiri dari beberapa topik, yaitu: politik, hukum, sosial budaya, ekonomi, dan olah raga. Jumlah total dokumen suara adalah 87 dokumen. Jumlah total dokumen berita teks adalah 3967 dokumen.

Penelitian ini menunjukan hasil yang paling baik dengan menggunakan metode pembobotan *TF-IDF* (*Word Frequency*) yaitu dengan ketepatan segmentasi 81,4% untuk dokumen teks hasil transkripsi dan 73,2% untuk dokumen teks. Proses pengenalan kata yang baik saat dilakukannya transkripsi dari dokumen suara merupakan faktor yang sangat penting untuk meningkatkan akurasi proses segmentasi.

2.2 GENETIC ALGORITHM

Salah satu pendekatan yang dapat digunakan untuk melakukan pemecahan suatu masalah adalah dengan menggunakan suatu algoritma pencarian solusi (*searching*). Algoritma pencarian adalah suatu cara untuk menemukan solusi dari suatu permasalahan, biasanya solusi diperoleh dengan mengevaluasi berbagai kemungkinan solusi yang mungkin. Ada beberapa jenis algoritma pencarian yang dapat digunakan dengan kelebihan dan kekurangannya masing-masing, salah satu diantaranya adalah algoritma *local search*. *Local search* merupakan suatu algoritma pencarian dimana proses pencarian solusi tidak dilakukan secara sistematis (menyimpan setiap langkah proses pencarian), melainkan pencarian dengan hanya memperhatikan suatu *state* dan hanya bergerak ke *state* lain yang cukup dekat dengan *state* tersebut. *State* merupakan suatu keadaan/solusi yang mungkin sebagai penyelesaian suatu permasalahan, sedangkan *state space* merupakan himpunan keseluruhan *state* yang mungkin (Russel & Norvig, 2003). Algoritma pencarian seperti ini sangat cocok diimplementasikan untuk masalah-masalah yang solusinya tidak mempermasalahkan langkah-langkah proses pencarian solusi tersebut (hanya mengutamakan solusi akhir). Keunggulan dari penggunaan algoritma *local search* ini adalah:

- Penggunaan *memory* yang cukup sedikit dan biasanya besarnya penggunaan selalu tetap/konstan.
- Algoritma *local search* biasanya dapat menemukan solusi yang cukup baik pada suatu *search space* yang sangat besar.

Salah satu varian dari algoritma *local search* adalah *genetic algorithm*. Setiap algoritma pencarian pasti memiliki suatu metode tersendiri untuk mencapai hasil yang diinginkan. *Genetic algorithm* beroperasi dengan menggunakan analogi seleksi alam (evolusi) pada dunia nyata. Pada *genetic algorithm* proes pencarian solusi dilakukan dengan beberapa tahap yang serupa dengan proses seleksi alam, yaitu proses seleksi, proses persilangan (*crossover*), dan proses mutasi. Ketiga proses tersebut akan membantu menghasilkan solusi yang cukup baik diantara *search space* yang sangat luas. Beberapa hal yang perlu dipahami pada *genetic algorithm* adalah:

- Individu / *Chromosome*

  Merupakan representasi solusi atau representasi *state* yang diinginkan untuk menyelesaikan suatu masalah. Bagaimana representasi solusi sebagai suatu individu ini merupakan suatu proses yang sangat penting karena tanpa representasi solusi yang baik maka *genetic algorithm* tidak akan beroperasi dengan baik. Individu ini biasanya merupakan serangkaian *string* yang setiap elemennya memiliki makna pada solusi yang diharapkan.
- Populasi

  Merupakan sekumpulan individu/*chromosome* yang akan menjalani proses evolusi pada *genetic algorithm*.
- *Fitness function – Fitness score*

  Fitness function merupakan suatu fungsi yang akan menilai baik atau buruknya suatu individu sebagai sebuah solusi bagi permasalahan. *Fitness score* adalah nilai yang diberikan pada suatu individu sebagai perbandingan tingkat baik atau buruknya individu tersebut dibandingkan dengan individu lainnya pada suatu populasi. Representasi baik atau buruknya suatu *fitness score* bergantung pada kebutuhan, biasanya semakin tinggi *fitness score* maka individu tersebut semakin baik.

- Seleksi

  Merupakan suatu proses pemilihan individu yang akan menjalani proses evolusi. Proses seleksi ini dilakukan secara acak/*random* tetapi berdasarkan atas *fitness score* dari masing-masing individu. Biasanya probabilitas terpilihnya suatu individu akan semakin besar jika individu tersebut memiliki *fitness score* yang baik. Dalam proses ini masih dimungkinkan suatu *chromosome* yang buruk akan disertakan dalam proses evolusi (karena pemilihan dengan metode acak/*random*), hal ini sesuai dengan analogi seleksi alam yang terjadi pada kenyataannya.

- Persilangan / *Crossover*

  Merupakan suatu proses menghasilkan individu baru dari 2 individu sebelumnya, sehingga tingkat keragaman individu pada keseluruhan proses evolusi ini akan semakin tinggi. Individu baru yang dihasilkan ini berasal dari pertukaran elemen-elemen *string* secara *random* dari kedua individu sebelumnya. Hal ini akan menjamin keragaman individu pada keseluruhan proses evolusi. Keragaman akan terjadi pada tahap-tahap awal evolusi sebelum individu-individu tersebut konvergen pada suatu solusi yang baik. Contoh proses *crossover* pada *genetic algorithm* dapat dilihat pada Gambar 2.3 sebagai berikut:

  010101010101 → 010101111111
  111111111111 → 111111010101

Gambar 2.3 Contoh proses *crossover*

- Mutasi

  Merupakan proses perubahan secara *random* elemen-elemen pada suatu individu hasil dari proses *crossover*. Contoh proses mutasi individu hasil *crossover* pada *genetic algorithm* dapat dilihat pada Gambar 2.3 sebagai berikut:

  010101101111
  111111010111

Gambar 2.4 Contoh proses mutasi

Pseudocode dari *genetic algorithm* dapat dilihat pada Gambar 2.4 sebagai berikut (Russel & Norvig, 2003):

```
function GENETIC-ALGORITHM(population, FITNESS-FUNCTION)
returns an individual
   inputs:    population, a set of individuals
              FITNESS-FUNCTION, a function that measures the fitness of
              an individual
   repeat
      New_population ← empty set
      Loop for i from 1 to SIZE(population) do
            x ←RANDOM-SELECTION(population, FITNESS-
            FUNCTION)
            y ← RANDOM-SELECTION (population, FITNESS-
            FUNCTION)
            child ←REPRODUCE(x,y)
            if (small random probability) then child ← MUTATE(child)
            add child to new_population
      population ← new_population
   until some individual is fit enough , or enough time has elapsed
   return the best individual in population, according to FITNESS-FUNCTION

function REPRODUCE(x,y) returns an individual
   inputs: x,y, parent individuals
   n ← LENGTH(x)
   c ←random number from 1 to n
   return APPEND(SUBSTRING(x,1,c),SUBSTRING(y,c+1,n))
```

Gambar 2.5 *Pseudocode genetic algorithm*

2.2.1 PENELITIAN SEGGEN

Penelitian SegGen (SegGen: a Genetic Algorithm for Linear Text Segmentation) adalah salah satu penelitian untuk melakukan segmentasi pada dokumen menjadi bagian-bagian yang homogen untuk dokumen secara umum (Lamprier et al., 2007). Pada penelitian SegGen ini segmentasi dokumen dilakukan dengan *genetic algorithm* yang cukup berbeda dibanding metode-metode lain yang digunakan pada penelitian mengenai segmentasi dokumen, letak perbedaannya adalah pada penelitian SegGen dengan *genetic algorithm* ini proses penentuan letak segmen-segmen (*boundaries*) tidak dilakukan secara sekuensial

(misalnya seperti pada metode Texttiling), tetapi penentuan letak segmen dipandang secara global dari suatu dokumen. Pada penelitian SegGen ini digunakan 2 kriteria yang akan digunakan untuk menghasilkan segmentasi dokumen yang baik, yaitu: memaksimalkan *internal cohesion* (tingkat kemiripan antara kalimat-kalimat penyusun suatu segmen) pada suatu segmen dan memaksimalkan ketidakmiripan antar segmen yang berdekatan (*dissimilarity*). Sesuai dengan dua kriteria tersebut, maka segmentasi dokumen dapat dipandang sebagai menemukan bagian-bagian pada dokumen dengan keterkaitan yang kuat antara kalimat-kalimat penyusunnya dan tidak berkaitan dengan bagian lain yang berdekatan. Oleh karena itu segmentasi dokumen merupakan suatu *multiobjective problem* dengan dua fungsi untuk dioptimisasi yaitu *internal cohesion* dan *dissimilarity*. Kedua kriteria ini akan saling bertolak belakang dimana *internal cohesion* akan meningkat seiring dengan banyaknya jumlah segmen yang dibuat, dan sebaliknya *dissimilarity* akan semakin rendah jika terlalu banyak segmen. Penjelasan lebih lanjut mengenai *multiobjective problem* akan dibahas pada subbab 2.2.2.

Penelitian SegGen ini mengggunakan pendekatan *strength pareto evolutionary algorithm* (SPEA) (Zitzler, 1999) sebagai metode untuk melakukan optimisasi pada kedua kriteria tersebut, yang merupakan salah satu pendekatan penyelesaian *multiobjective problem*. Penjelasan lebih lanjut mengenai *strength pareto evolutionary algorithm* (SPEA) akan dibahas pada subbab 2.2.3. Beberapa perumusan yang dilakukan penelitian SegGen ini untuk menemukan solusi letak segmen dengan *genetic algorithm* adalah sebagai berikut:

- Perumusan masalah

  Untuk setiap individu $\vec{x}$ akan diterapkan dua kriteria evaluasi, yaitu: $C(\vec{x}) \in [0,1]$ yang merupakan nilai *internal cohesion* (kemiripan antar kalimat pada satu segmen) pada segmen tersebut dan $D(\vec{x}) \in [0,1]$ yang merupakan nilai *dissimilarity* (ketidakmiripan antara segmen yang berdekatan). Optimisasi permasalahan juga merupakan aproksimasi dari himpunan kedua kriteria tersebut, yaitu:

$$\mathcal{O} = \{\vec{x} \in \{0,1\}^{ns-1} \mid \nexists \vec{x}' \in \{0,1\}^{ns-1}, \left(C(\vec{x}) < C(\vec{x}')\right) \wedge \left(D(\vec{x}) < D(\vec{x}')\right)\} \quad (2.3)$$

- Individu / c*hromosome* dan populasi awal

  Dokumen dengan ns kalimat akan direpresentasikan dengan vektor biner $\vec{x}$ dengan $(ns-1)$ elemen. $x_i = 1$ menunjukan bahwa terdapat segmen diantara kalimat ke i dan ke $i+1$. Populasi awal dipilih secara *random* dan dibuat secara bertahap. Untuk menjamin keragaman populasi maka individu dibuat berdasarkan pada letak segmen-segmen pada individu lain, serta untuk mempermudah *genetic algorithm* dalam mencapai konvergensi maka pada populasi awal disertakan hasil segmentasi yang baik dari metode segmentasi dokumen lain.

- *Fitness function*

  Seperti telah dijelaskan sebelumnya, penelitian SegGen menggunakan dua kriteria untuk dioptimisasi yaitu *internal cohesion* dan *dissimilarity*. Untuk menghitung kedua kriteria tersebut diperlukan penghitungan tingkat kemiripan antara dua buah kalimat. Penghitungan kemiripan tersebut dihitung berdasarkan *cosine similarity measure*, yaitu:

$$Sim(s_1, s_2) = \frac{\sum_{i=1}^{t} w_{i,s_1} \times w_{i,s_2}}{\sqrt{\sum_{i=1}^{t} w_{i,s_1}^2 \times \sum_{i=1}^{t} w_{i,s_2}^2}} \quad (2.4)$$

  dimana t merupakan jumlah kata bermakna, dan w_{i,s_j} adalah bobot kata bermakna i pada kalimat s_j. *Internal cohesion* dihitung berdasarkan:

$$C(\vec{x}) = \frac{\sum_{i=1}^{nbseg} SumSim\ (seg_i)}{\sum_{i=1}^{nbseg} NbCouples\ (seg_i)} \quad (2.5)$$

  dimana $nbseg$ merupakan jumlah segmen pada individu, $SumSim(seg_i)$ merupakan jumlah dari kermiripan antar kalimat pada seg_i dan $NbCouples(seg_i)$ merupakan jumlah kemungkinan pasangan kalimat pada seg_i. Kemiripan antar segmen dihitung berdasarkan:

$$SimSeg(seg_1, seg_2) = \frac{\sum_{s_j \in S(seg\,1)} \sum_{s_k \in S(seg\,2)} Sim(s_j, s_k)}{|S(seg_1)| \times |S(seg_2)|} \quad (2.6)$$

dengan seg_i merupakan segmen, s_j merupakan kalimat, $S(seg_i)$ merupakan himpunan kalimat penyusun segmen seg_i dan $|S(seg_1)|$ merupakan kardinalitas himpunan tersebut. Sedangkan ketidakmiripan antar segmen dapat dihitung berdasarkan:

$$D(\vec{x}) = 1 - \left(\frac{\sum_{i=1}^{nbseg\ -1} SimSeg\ (seg_i, seg_{i+1})}{nbseg\ -1} \right) \quad (2.7)$$

dengan $nbeseg$ adalah jumlah total segmen pada $\vec{x}$ dan seg_i adalah segmen. Pada *strength pareto evolutionary algorithm*, *fitness score* suatu individu dihitung berdasarkan dominasi antar individu pada populasi. Suatu individu terdominasi jika terdapat suatu individu lain pada populasi yang memiliki nilai yang lebih baik atau sama untuk semua kriteria. Individu yang tidak didominasi akan disimpan pada suatu *archive* sehingga informasi individu yang tidak terdominasi tidak akan hilang selama proses evolusi. Untuk setiap individu yang terdapat pada *archive* ini akan dihitung suatu *hardness value H* yang dihitung berdasarkan jumlah individu pada populasi yang didominasinya:

$$H(\vec{x}) = \frac{|\{\vec{y} / \vec{y} \in P_t \wedge C(\vec{x}) > C(\vec{y}) \wedge\ D(\vec{x}) > D(\vec{y})\}|}{|P_t| + 1} \quad (2.8)$$

dengan $\bar{P}$ adalah himpunan elemen pada *archive*. *Fitness score* pada individu yang terdapat pada *archive* merupakan *inverse* dari nilai *hardness value*: $F(\vec{x}) = \frac{1}{H(\vec{x})}$. *Fitness score* pada individu yang terdapat pada populasi dapat dihitung dengan menjumlahkan *hardness value* individu-individu yang mendominasinya:

$$F(\vec{x}) = \frac{1}{1+\sum_{\vec{x} \in \bar{P} \wedge C(\vec{x}) \geq C(\vec{y}) \wedge D(\vec{x}) \geq D(\vec{y})} H(\vec{x})} \quad (2.9)$$

dengan P_t adalah himpunan populasi.

- Operator evolusi

  Evolusi pada *genetic algorithm* terdiri dari 3 tahap, yaitu seleksi, *crossover*, dan mutasi. Pada penelitian SegGen ini proses seleksi dilakukan dengan menggunakan kan *fitness score* untuk menentukan probabilitas pemilihan (individu dengan *fitness score* tinggi akan mendapat probabilitas lebih tinggi). Proses *crossover* dilakukan dengan menyilangkan secara *random* bagian dari 2 individu. Proses mutasi dilakukan dengan menggunakan 2 jenis mutasi yaitu mutasi dengan menggati individu dengan individu lain yang dibuat secara *random*, dan mutasi dengan menggati letak segmen ke kalimat terdekat pada suatu individu dengan probabilitas tertentu.

- Ekstraksi solusi

  Ketika proses evolusi selesai, maka *archive* akan mengandung himpunan individu yang tidak terdominasi, maka untuk menentukan solusi yang terbaik dilakukan proses agregasi kedua kriteria pada elemen dari *archive* :

  $$Agg(\vec{x}) = C(\vec{x}) + \alpha \times D(\vec{x}) \quad (2.10)$$

  Pada tahap agregasi ini, kriteria tingkat ketidakmiripan antara segmen berdekatan diberi bobot lebih tinggi dibandingkan kriteria *internal cohesion*. Individu yang baik merupakan individu dengan nilai agregasi terbaik.

Penelitian SegGen menggunakan data dokumen berupa artikel. Dokumen-dokumen yang digunakan terdiri dari 4 jenis yaitu: dokumen dengan 50 kalimat dan 2 segmen, dokumen dengan 50 kalimat dan 4 segmen, dokumen dengan 100 kalimat dan 4 segmen, dan dokumen dengan 100 kalimat dan 8 segmen. Hasil segmentasi dievaluasi dengan menghitung *precision, recall,* dan *WindowDiff* (*WindowDiff* tidak akan dibahas lebih lanjut dalam laporan tugas akhir ini).

Hasil yang diperoleh pada penelitian SegGen ini cukup baik dibandingkan dengan metode-metode segmentasi sebelumnya yang menggunakan metode sekuensial dalam menentukan batasan. Hasil penelitian SegGen inipun menunjukan bahwa metode ini cukup adaptif terhadap jumlah batas yang harus ditemukan.

2.2.2 MULTIOBJECTIVE PROBLEM AND OPTIMIZATION

Beberapa jenis permasalahan yang lazim ditemui pada kehidupan nyata adalah *single-objective problem* (SOP) dan *multiobjective problem* (MOP). Pada *single-objective problem* permasalahan yang dihadapi adalah bagaimana mencari suatu solusi yang optimal terhadap satu kriteria permasalahan, misalnya bagaimana cara meminimalkan biaya produksi suatu produk. Untuk solusi permasalahan *single-objective problem* tersebut, secara mudah solusi dapat diperoleh dengan cara meminimalkan biaya pembelian bahan baku produk teresebut, atau dengan kata lain untuk menemukan solusi untuk permasalahan *single-objective problem* adalah dengan mencari nilai optimal untuk permasalahan tersebut dan biasanya proses menemukan solusi optimal tersebut terdefinisikan secara jelas. Pada *multiobjective problem,* permasalahan yang dihadapi adalah bagaimana mencari suatu solusi yang optimal terhadap beberapa (lebih dari satu) kriteria permasalahan, misalnya bagaimana cara meminimalkan biaya produksi suatu produk dan meningkatkan kualitasnya. Pada permasalahan *multiobjective problem* tersebut, solusi optimal tidak dapat diperoleh hanya dengan cara mengoptimalkan kedua kriteria secara terpisah karena kedua kriteria tersebut saling bertolak belakang (jika biaya minimal maka kualitas akan menurun, dan sebaliknya). *Multiobjective problem* dapat direpresentasikan sebagai berikut (Zitzler, 1999):

Memaksimalkan: $y = f(x) = (f_1(x), f_2(x), f_3(x), \ldots, f_k(x))$

Terhadap: $e(x) = \big(e_1(x), e_2(x), .., e_m(x)\big) \leq 0$

Dimana: $x = \big(x_1, x_2, \ldots, x_{n,}\big) \in X$

$y = \big(y_1, y_2, \ldots, y_{n,}\big) \in Y$

dengan x adalah vektor solusi, y adalah vektor nilai fungsi objektif, X adalah domain solusi, dan Y adalah domain *objective*. *Constraint* $e(x) \leq 0$ menunjukan himpunan solusi yang mungkin.

Pada *single-objective problem*, pencarian solusi dilakukan dengan mencari nilai yang membuat fungsi objektif maksimal. Pada *multiobjective problem* proses pencarian solusi tidak dapat dilakukan semudah pada *single-objective problem*, mengingat fungsi objektif yang perlu dioptimisasi merupakan fungsi-fungsi yang saling bertolak belakang satu sama lain. Untuk suatu vektor objektif u dan v berlaku (Zitzler, 1999):

$u = v$ jika dan hanya jika $\forall\, i \in \{1,2,\ldots,k\} : u_i = v_i$

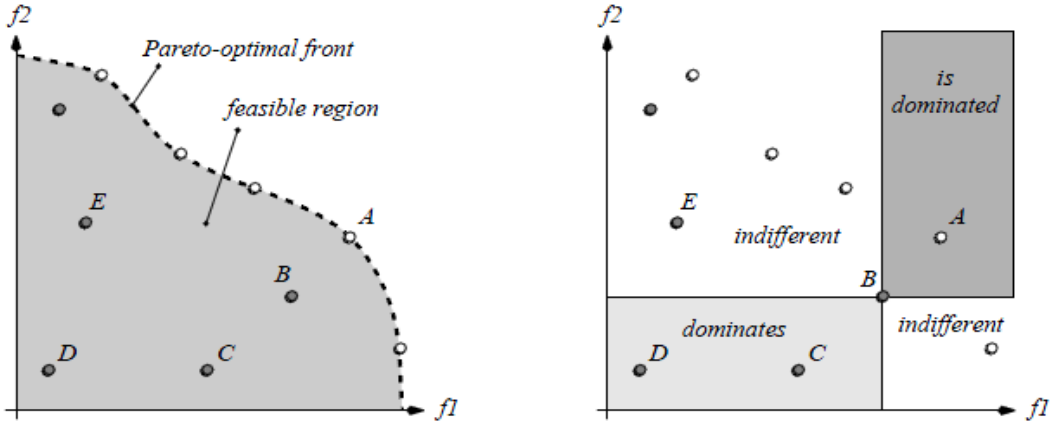
$u \geq v$ jika dan hanya jika $\forall\, i \in \{1,2,\ldots,k\} : u_i \geq v_i$

$u > v$ jika dan hanya jika $u \geq v \wedge u \neq v$

$u \leq v$ jika dan hanya jika $\forall\, i \in \{1,2,\ldots,k\} : u_i \leq v_i$

$u < v$ jika dan hanya jika $u \leq v \wedge u \neq v$

Contoh ilustrasi *multiobjective problem* dengan 2 fungsi objektif dapat dilihat pada Gambar 2.5 sebagai berikut (Zitzler, 1999):



Gambar 2.6 Ilustrasi multiobjective problem dengan pareto optimality pada domain fungsi objektif

Pada gambar sebelah kiri dapat dilihat bahwa solusi B lebih baik dari daripada solusi C, karena solusi B memiliki nilai yang lebih baik dari C pada kedua fungsi. Demikian pulan dengan Solusi C yang lebih baik daripada solusi D karena untuk nilai fungsi f_1, solusi C memiliki nilai yang lebih baik meskipun untuk fungsi $f_2$kedua solusi memiliki nilai yang sama. Namun hal yang berbeda terjadi pada solusi B dan E, dimana solusi B memiliki nilai yang lebih baik untuk fungsi pertama tetapi solusi E memiliki nilai yang lebih baik untuk fungsi kedua .Hal tersebut dapat direpresentasikan dengan B > C, C > D, B > D, B $\not\succ$ E, dan E $\not\succ$ B oleh karena itu pada *multiobjective problem,* hubungan antar vektor solusi pada domain fungsi objektif dapat dinyatakan sebagai berikut untuk dua buah vektor solusi a dan b:

$a \succ b$ (a *dominates* b) jika dan hanya jika $f(a) > f(b)$

$a \succcurlyeq b$ (a *weakly dominates* b) jika dan hanya jika $f(a) \geq f(b)$

$a \sim b$ (a *indiferrent* b) jika dan hanya jika $f(a) \not\geq f(b) \wedge f(b) \not\geq f(a)$

(representasi ini berlaku pula untuk $\prec, \preccurlyeq$, dan $\sim$ untuk permasalahan mencari minimal)

Sehingga dapat peroleh kesimpulan bahwa pada gambar diatas B $\succ$ C, B $\succ$ D, C $\succ$ D, dan B $\sim$ E. Pada contoh gambar diatas solusi A merupakan *pareto optimal*, karena A tidak didominasi oleh solusi lain, sehingga solusi tersebut tidak dapat ditingkatkan lagi tanpa menurunkan nilai salah satu fungsi objektifnya. Pada gambar diatas kumpulan titik-titik berwarna lebih terang merupakan kumpulan solusi yang merupakan *pareto optimal* dan solusi-solusi tersebut bersifat *indifferent* satu sama lain. Himpunan dari seluruh solusi-solusi yang merupakan *pareto optimal* disebut *pareto optimal set*.

2.2.3 STRENGTH PARETO EVOLUTIONARY ALGORITHM (SPEA)

Salah satu pendekatan dalam menyelesaikan permasalahan *multiobjective* problem dengan *genetic algorithm* adalah dengan pendekatan *strength pareto evolutionary algorithm* (SPEA). Pada SPEA, proses untuk mencari himpunan solusi dari

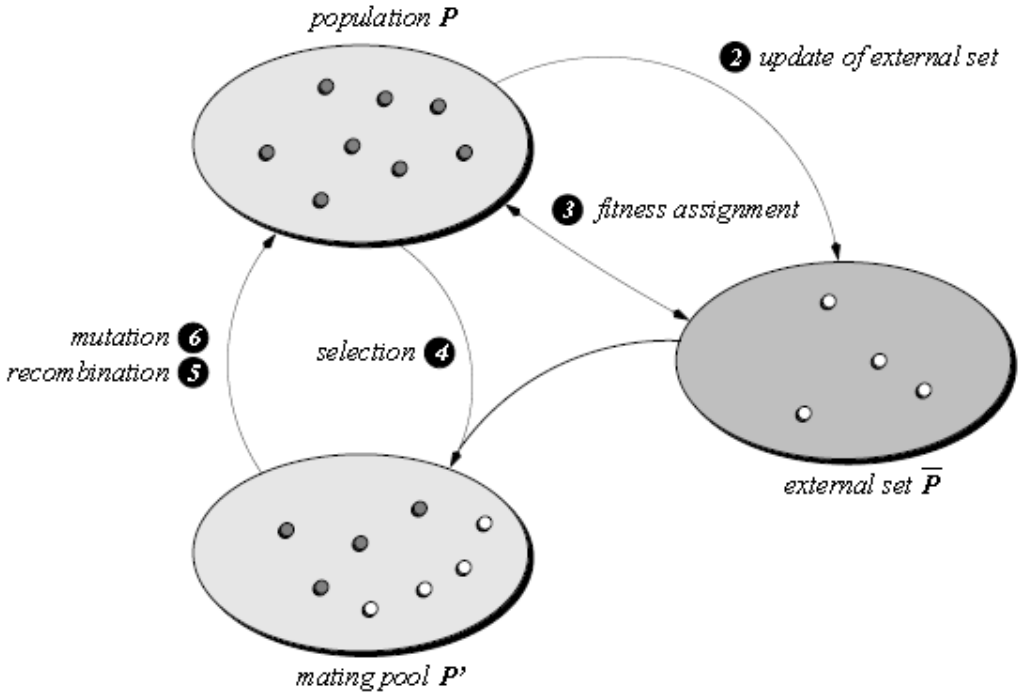
multiobjective problem dilakukan dengan pendekatan *pareto optimal*, seperti telah dijelaskan pada subbab sebelumnya. SPEA menggunakan suatu *archive* untuk menyimpan himpunan solusi yang tidak didominasi oleh solusi-solusi lainnya dalam populasi, hal ini bertujuan agar himpunan solusi-solusi yang baik (tidak terdominasi) tidak akan hilang selama proses evolusi. Himpunan solusi yang tidak didominasi ini akan dilibatkan dalam proses evolusi dan menentukan *fitness score* bagi setiap individu lain pada populasi.

SPEA merupakan algoritma yang cukup populer untuk permasalahan-permasalahan *multiobjective optimization*. *Pesudocode* dari SPEA dapat dilihat pada Gambar 2.6 sebagai berikut (Zitzler, 1999):

Input:	N	(*population size*)
	$\overline{N}$	(maximum size of external set)
	T	(maximum number of generation)
	p_c	(crossover probability)
	p_m	(mutation rate)
Output:	A	(nondominated set)
Step 1:	Initialization: Generate an initial population P_0 and create the empty external set $\overline{P_0} = \emptyset$. Set t = 0.	
Step 2:	Update of external set: Set the temporary external set $\bar{P}' = \overline{P_t}$.	
	a)	Copy individuals whose decision vectors are nondominated regarding $m(P_t)$ to $\overline{P}'$: $\bar{P}' = \bar{P}' + \{i \mid i \in P_t \wedge m(i) \in p(m(P_t))\}$.
	b)	Remove individuals from $\bar{P}'$ whose corresponding decision vectors are weakly dominated regarding $m(\bar{P}')$, i.e., as long as there exists a pair (i, j) with $i, j \in \bar{P}'$ and $m(i) \succcurlyeq m(j)$ do $\bar{P}' = \bar{P}' - \{j\}$
	c)	Reduce the number of individuals externally stored by means of clustering, and assign the resulting reduced set to $\bar{P}_{t+1}$.
Step 3:	Fitness assignment: Calculate fitness values of individuals in P_t and $\overline{P_t}$.	
Step 4:	Selection: Set $P' = \emptyset$. For $i = 1, \ldots, N$ do	
	a)	Select two individuals $i, j \in P_t + \bar{P}_t$ at random.
	b)	If $F(i) < F(j)$ then $P' = P' + \{i\}$ else $P' = P' + \{j\}$. Note that fitness is to be minimized here.
Step 5:	Recombination	
Step 6:	Mutation	
Step 7:	Termination: Set $P_{t+1} = P^m$ and $t = t + 1$. If $t \geq T$ or another stopping criterion is satisfied then set $A = p(m(P_t))$ else go to step 2.	

Gambar 2.7 *Pseudocode* SPEA (Zitzler, 1999)

Untuk memperjelas proses kerja algoritma SPEA ini, gambaran alur proses algoritma SPEA dapat dilihat pada Gambar 2.7 sebagai berikut:



Gambar 2.8 Alur proses SPEA (Zitzler, 1999)

Pada algoritma SPEA, *fitness score* suatu inidividu pada populasi dihitung berdasarkan individu lain pada *archive* yang mendominasinya, sedangkan untuk individu yang terdapat pada *archive* diberikan suatu nilai "*strength*" yang menunjukan seberapa banyak individu tersebut mendominasi individu lain pada suatu evolusi. *Fitness score* untuk individu yang merupakan anggota dari *archive* sama dengan nilai "*strength*" individu tersebut. Pada algoritma SPEA ini, semakin baik suatu individu maka nilai *fitness score* nya akan semakin kecil. *Pseudocode* penghitungan *fitness score* untuk semua individu dapat dilihat pada Gambar 2.8 sebagai berikut (Zitzler, 1999):

Input:	P_t	(population)
	$\bar{P}_t$	(external set)
Output:	F	(Fitness values)

Step 1: Each individual $i \in \bar{P}_t$ is assigned a real value $S(i) \in [0,1]$, called strength, $S(i)$ is proportional to the number of population members $j \in P_t$ for which $m(i) \succcurlyeq m(j)$:

$$S(i) = \frac{|\{j \mid j \in P_t \wedge m(i) \succcurlyeq m(j)\}|}{N+1}$$

The fitness of i is equal to its strenghth: $F(i) = S(i)$

Step 2: The fitness of an individual $j \in P_t$ is calculated by summing the strengths of all externally stored individuals $i \in \bar{P}_t$ whose decision vectors weakly dominate $m(j)$. Add one to the total in order to guarantee that members of $\bar{P}_t$ gave better fitness than members of P_t (fitness is to be minimized)

$$F(j) = 1 + \sum_{i \in \bar{P}_t, m(i) \succcurlyeq m(j)} S(i) \; where \; f(j) \in [1, N)$$

Gambar 2.9 *Pseudocode* penghitungan *fitness score* (Zitzler, 1999)

2.2.4 STRENGTH PARETO EVOLUTIONARY ALGORITHM 2 (SPEA 2)

Strength pareto evolutionary algorithm 2 (SPEA 2) merupakan suatu algoritma hasil pengembangan dari algoritma SPEA. Secara umum algoritma SPEA 2 ini mirip dengan pendahulunya yaitu algoritma SPEA, namun beberapa perbedaan yang terdapat pada algoritma SPEA 2 dibandingkan dengan algoritma SPEA adalah:

- Pengembangan cara penghitungan *fitness score* untuk setiap individu.
  Pada SPEA 2 ini, *fitness score* suatu individu tidak hanya ditentukan oleh individu-individu yang terdapat pada *archive*, melainkan penentuan *fitness score* juga memperhitungkan jumlah individu lain yang didominasi dan mendominasi individu tersebut.
- Penghitungan nilai densitas yang berkaitan dengan proses penghitungan *fitness score*.

- Implementasi metode *archive truncation* untuk mengurangi jumlah individu di *archive* jika jumlah individu di *archive* sudah melewati batas maximum.

Algoritma SPEA 2 ini dapat mengatasi beberapa kekurangan algoritma SPEA, diantaranya adalah:

- Pada SPEA , karena *fitness score* setiap individu pada populasi yang tidak termasuk kedalam *archive* hanya bergantung dari nilai "*strength"* individu pada *archive* yang mendominasinya maka seandainya pada *archive* hanya terdapat satu individu, semua individu pada populasi akan mendapat *fitness score* yang sama. Hal ini kurang baik karena *fitness score* menjadi kurang menggambarkan kualitas suatu individu.
- *Density estimation*
  Memberikan suatu nilai tambahan agar *fitness score* tidak hanya dihitung berdasarkan nilai dominasi saja. Nilai *density* ini dihitung berdasarkan perbedaan nilai objektif suatu individu terhadap individu lain.
- *Archive Truncation*
  Penyempurnaan metode pemilihan individu di *archive*, jika jumlah individu di *archive* melebihi batas maksimum.

Pseudocode dari SPEA 2 dapat dilihat pada Gambar 2.9 (Zitzler, 2004). Pada algoritma SPEA 2, penghitungan *fitness score* dilakukan dengan beberapa tahapan yang melibatkan keseluruhan individu, tidak hanya ditentukan oleh individu yang terdapat pada *archive*. Pada SPEA 2, semakin kecil *fitness score* suatu individu maka kualitas individu tersebut semakin baik. Proses penghitungan *fitness score* adalah (Zitzler, 2004):

- Untuk semua individu yang termasuk pada *archive* dan yang tidak, hitung nilai $S(i)$ yang merupakan jumlah individu lain yang didominasi oleh individu tersebut.

$$S(i) = |\{ j \mid j \in P_t + A_t \wedge i \succ j\}| \qquad (2.11)$$

- Setelah menghitung $S(i)$ untuk semua individu, lalu hitung pula $R(i)$ untuk setiap individu yang merupakan nilai *raw fitness*, berdasarkan nilai $S(i)$ individu-individu yang mendominasi suatu individu tersebut.

$$R(i) = \sum_{j \in P_t + A_t,\ j \succ i} S(j) \tag{2.12}$$

- Menghitung nilai densitas $D(i)$, dimana densitas merupakan perbedaan nilai fungsi objektif (σ_i^k) suatu individu terhadap individu lainnya.

$$D(i) = \frac{1}{\sigma_i^k + 2} \tag{2.13}$$

  Penambahan angka dua dimaksudkan agar menghindari pembagian dengan nol.

- Menghitung *fitness score* $R(i)$ untuk setiap individu.

$$F(i) = R(i) + D(i) \tag{2.14}$$

Dengan metode penghitungan *fitness score* seperti diatas, SPEA 2 dapat mengatasi beberapa kelemahan dari algoritma sepelumnya, yaitu SPEA.

Input	M	(offspring population size)
	N	(archive size)
	T	(maximum number of generations)
Output	A^*	(nondominated set)
Step 1:	Initialization: Generate an initial population P_0 and create the empty archive (external set) $A_0 = \emptyset$. Set $t = 0$.	
Step 2:	Fitness assignment: Calculate fitness values of individuals in P_t and A_t	
Step 3:	Environmental selection: Copy all nondominated individuals in P_t and A_t to A_{t+1}. If size of of A_{t+1} exceeds N then reduce A_{t+1} by means of the truncation operator, otherwise it size of A_{t+1} is less then fill A_{t+1} with dominated individuals in P_t and A_t.	
Step 4:	Termination: if $t \geq T$ or another stopping criterion is satisfied then set A^* to the set of decision vectors represented by the non dominated individuals in A_{t+1}. Stop.	
Step 5:	Mating selection: Perform binary tournament selection with replacement on A_{t+1} in order to fill the mating pool.	
Step 6:	Variation: Apply recombination and mutation operators to the mating pool and set P_{t+1} to the resulting population. Increment generation counter ($t = t + 1$) and go to Step 2	

Gambar 2.10 *Pseudocode* SPEA 2 (Zitzler, 2004)