TESIS - KI2502

POSITION TEXT GRAPH DAN PERAN SEMANTIK KATA DALAM PEMILIHAN KALIMAT REPRESENTATIF *CLUSTER* PADA PERINGKASAN MULTI-DOKUMEN

Gus Nanang Syaifuddiin
5113201040

PEMBIMBING I
Dr. Agus Zainal Arifin, S.Kom, M.Kom.

PEMBIMBING II
Diana Purwitasari, S.Kom, M.Sc.

PROGRAM MAGISTER
BIDANG KEAHLIAN KOMPUTASI CERDAS & VISUALISASI
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2015

TESIS - KI2502

POSITION TEXT GRAPH AND ROLE OF SEMANTICS WORD ON ELECTING REPRESENTATIVE SENTENCE OF CLUSTER IN RESUMING MULTI DOCUMENT

Gus Nanang Syaifuddiin
5113201040

SUPERVISOR I
Dr. Agus Zainal Arifin, S.Kom, M.Kom.

SUPERVISOR II
Diana Purwitasari, S.Kom, M.Sc.

MAGISTER PROGRAM
THE EXPERTISE FIELD OF INTELLIGENT COMPUTING AND VISUALISATION
DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2015

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas berkat limpahan rahmat dan hidayat-Nya sehingga Tesis yang berjudul:

***Position Text Graph* Dan Peran Semantik Kata Dalam Pemilihan Kalimat Representatif Cluster Pada Peringkasan Multi-Dokumen**

dapat diselesaikan dengan baik. Semoga tesis ini dapat memberikan manfaat pada perkembangan ilmu pengetahuan khususnya dalam bidang peringkasan multi-dokumen serta dapat memberikan kontribusi bagi peneliti selanjutnya. Dengan selesai dan tersusunnya laporan tesis ini, maka penulis mengucapkan terima kasih atas bantuan dan dukungan dari berbagai pihak baik moril maupun materiil dalam pembuatan tesis ini, antara lain:

1. Bapak Waskitho Wibisono, S.Kom., M.Eng., Ph.D. selaku Ketua Program Magister Teknik Informatika yang telah memberi dukungan dan arahan dalam menyelesaikan permasalahan akademik.
2. Bapak Dr. Agus Zainal Arifin, S.Kom, M.Kom selaku dosen pembimbing I yang telah banyak memotivasi dan membuka cakrawala dalam memandang persoalan dari sudut riset. Dan dengan kesabarannya banyak mendorong dan membimbing proses terselesaikannya tesis ini.
3. Ibu Diana Purwitasari, S.Kom, M.Sc. selaku dosen pembimbing II yang memotivasi dan dengan kesabarannya membimbing dan mendorong penulis dalam menyelesaikan tesis ini.
4. Ibu Dr. Chastine Fatichah, S.Kom., M.Kom., Ibu Isye Arieshanti, S.Kom., M.Phil., dan Ibu Wijayanti Nurul Khotimah, S.Kom., M.sc. selaku dosen penguji yang telah banyak memberikan motivasi dan saran yang mendukung terselesaikannya tesis ini.

5. Bapak Pardi dan Ibu Wiji selaku orang tua yang telah mendidik, membimbing dan selalu memberikan motivasi sehingga penulis dapat menyelesaikan tesis ini.
6. Teman-teman di lingkungan ITS dan semua pihak yang tidak dapat disebutkan satu per satu yang telah memberikan motivasi dan saran kepada penulis dalam menyelesaikan studi S2 ini. Semoga Tuhan Yang Maha Esa membalas semua kebaikan tersebut dengan pahala yang berlimpah. Sebagai akhir kata, penulis menyadari bahwa laporan tesis ini masih jauh dari kesempurnaan. Untuk itu kritik dan saran dari pembaca akan dapat digunakan untuk mengembangkan penelitian ini selanjutnya.

Surabaya, Januari 2016

Penulis

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom.)
di
Institut Teknologi Sepuluh Nopember Surabaya

oleh:
Gus Nanang Syaifuddiin
Nrp. 5113201040

Dengan judul :
POSITION TEXT GRAPH DAN PERAN SEMANTIK KATA DALAM PEMILIHAN KALIMAT REPRESENTATIF CLUSTER PADA PERINGKASAN MULTI-DOKUMEN

Tanggal Ujian : 18-1-2016
Periode Wisuda : 2015 Gasal

isetujui oleh:

r. Agus Zainal Arifin, S.Kom, M.Kom
IP. 197208091995121001 (Pembimbing 1)

fana Purwitasari, S.Kom, M.Sc
IP. 197804102003122001 (Pembimbing 2)

r. Eng. Chastine Fatichah, S.Kom, M.Kom
IP. 197512202001122002 (Penguji 1)

ye Arieshanti, S.Kom, M.Phil
IP. 197804122006042001 (Penguji 2)

Wijayanti Nurul Khotimah, S.Kom, M.Sc
IP. 198603122012122004 (Penguji 3)

Direktur Program Pasca Sarjana,

Prof.Ir.Djauhar Manfaat, M.Sc., Ph.D.
NIP. 196012021987011001

POSITION TEXT GRAPH DAN PERAN SEMANTIK KATA DALAM PEMILIHAN KALIMAT REPRESENTATIF CLUSTER PADA PERINGKASAN MULTI-DOKUMEN

Nama mahasiswa : Gus Nanang Syaifuddiin
NRP : 5113201040
Pembimbing I : Dr. Agus Zainal Arifin, S.Kom, M.Kom.
Pembimbing II : Diana Purwitasari, S.Kom, M.Sc.

ABSTRAK

Coverage dan *salient* merupakan masalah utama yang menjadi perhatian para peneliti dalam peringkasan dokumen. Pendekatan *clustering* mampu memberikan coverage yang baik terhadap semua topik namun tidak memiliki informasi-informasi yang bisa mewakili kalimat-kalimat lain (*salience sentence*).

Salience dapat digali dengan melihat hubungan dari satu kalimat dengan kalimat lain yang dibangun dengan pendekatan *position text graph*, namun *position text graph* hanya mampu menggali hubungan antar kalimat tanpa memperhatikan peran semantik kata *("who" did "what" to "whom", "where", "when", and "how")* dalam kalimat yang dibandingkan.

Pada tesis ini kami mengusulkan sebuah metode baru strategi pemilihan kalimat representatif *cluster* yang diberi nama SSID (*Semantic Sentence Information Density*) dengan pendekatan *position text graph* dan peran semantik kata pada peringkasan multi-dokumen. Beberapa tahapan dalam penelitian ini adalah *text preprocessing*, *clustering* kalimat, pengurutan *cluster*, pemilihan kalimat representatif *cluster* dan penyusunan hasil ringkasan.

Uji coba dilakukan terhadap dataset *Document Understanding Conference* (DUC) 2004 *Task 2*. Hasil uji coba menunjukkan SSID berhasil mengatasi kelemahan *position text graph* dan meningkatkan nilai korelasi ROUGE-1 dan ROUGE-2. Nilai analisa ROUGE-1 pada proses SSID meningkat 0.85% jika dibandingkan dengan LIGI dan 2.42% dibandingkan dengan SIDeKiCK. Pada analisa ROUGE-2 SSID meningkat 10.33% jika dibandingkan dengan LIGI dan meningkat 9.73% dibandingkan dengan SIDeKiCK.

Kata kunci: peringkasan multi-dokumen, *position text graph, semantic role labeling*, *salience* dan *coverage*

POSITION TEXT GRAPH AND ROLE OF SEMANTICS WORD ON ELECTING REPRESENTATIVE SENTENCE OF CLUSTER IN RESUMING MULTI DOCUMENT

Name	: Gus Nanang Syaifuddiin
Student Identity Number	: 5113201040
Supervisor I	: Dr. Agus Zainal Arifin, S.Kom, M.Kom.
Supervisor II	: Diana Purwitasari, S.Kom, M.Sc.

ABSTRACT

Coverage and salient is the main problem to the attention of researchers in document summarisation. Sentence clustering approach gives good coverage of all the topics and has information that can represent other sentences (salience sentence).

Salience can be explored by looking at the relationship from one sentence to another sentence that was built with the approach position text graph, but the position of text graph only explore the relationship between a sentence without considering the role of semantic word (Who"did What"to Whom," Where ,"'when and" how)' in the sentence being compared.

In this thesis, we propose a new method of election strategy sentence cluster representative named SSID (Semantic Sentence Information Density) to approach the text position and role of the semantic graph word in multi-documents summarization. Several stages in this study: text processing, clustering sentences with histogram-based similarity clustering, sorting cluster, selection of a representative sentence cluster and preparation of a summary.

The test is done with the dataset Document Understanding Conference (DUC) 2004. The results showed SSID have the higest value of the correlation in ROUGE-1 and ROUGE-2. The value ROUGE-1 on the SSID increased 0.85% compared with LIGI and increased 2.42% compared with the sidekick. In ROUGE-2 SSID 10.33% when compared with LIGI and increased 9.73% compared with the SIDeKiCK.

.

Keywords:*multi-document summarization, position text graph, semantic role labeling,salience dan coverage*

DAFTAR ISI

halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

DAFTAR LAMPIRAN

Halaman

BAB 1
PENDAHULUAN

Pada era teknologi saat ini pertukaran informasi khususnya dokumen menjadi suatu hal yang umum sehingga jumlah dokumen meningkat secara signifkan. Ini menyulitkan seseorang dalam melakukan pencarian dokumen yang sesuai dengan dokumen yang mereka inginkan. Sehingga dibutuhkan peringkasan multi dokumen yang mampu melakukan peringkasan dokumen secara otomatis. Peringkasan multi-dokumen secara otomatis menjadi salah satu topik penting dalam *Natural Langguange Processing* (NLP) beberapa tahun terakhir ini (Barzilay dkk, 2005).

Terdapat dua metode yang digunakan dalam melakukan peringkasan nulti-dokumen secara otomatis: *abstractive* dan *extractive.* Peringkasan secara *abstractive* dilakukan dengan mendapatkan informasi yang disampaikan oleh dokumen sumber dan membentuk peringkasan dengan teknik menyatukan atau menurunkan informasi (Barzilay dkk, 2005). Peringkasan *extractive* dilakukan dengan melakukan extraksi terhadap kalimat dan mengurutkan berdasarkan nilai paling tinggi dan dijadikan sebagai kandidat ringkasan. *S*ebagian besar peneliti terfokus pada peringkasan multi-dokumen secara *extractive* seperti yang dilakukan pada penelitian ini.

Kalimat yang dipilih pada hasil ringkasan harus mempunyai *good coverage* dan *salient* terhadap topik dari dokumen sumber. *Coverage* dan *salient* menjadi masalah utama dalam peringkasan secara *abstractive* atau *extractive*. *Clustering* kalimat merupakan salah satu metode yang dapat memberikan *good coverage.* Beberapa penelitian diantaranya (Schlesinger dkk, 2008) dengan CLASSY (*clustering, Linguistics, And Statistics for Summarization Yield*) digunakan untuk melakukan pemangkasan kalimat secara bahasa dan menggunakan metode statistik untuk mendapatkan ringkasan topik dari dokumen sumber.

Ma dkk (2009) membangun *cluster* berdasarkan gabungan *query sentence* dan mendapatkan ringkasan kalimat berdasarkan modifikasi MMR (*Maximal Marginal Relevance*) (Carbonell dkk, 1998). Gupta dkk (2012) menyajikan sebuah metode untuk peringkasan multi-dokumen dengan menggabungkan ringkasan dokumen tunggal dan membentuk *cluster* kalimat dari beberapa fitur diantaranya bobot, kalimat, lokasi fitur dan konsep kesamaan fitur. *Good coverage* dapat dicapai dengan menjaga tingginya hubungan antar *cluster* (Sarkar, 2009), sehingga dapat mengidentifikasi topik dan sub-topik kedalam *cluster-cluster* bentukan. Koherensi dari *cluster* secara dinamis dipantau dengan menggunakan konsep yang disebut *cluster similarity histogram (*Hammouda dkk, 2004). *Similarity based Histogram clustering* (SHC) terbukti lebih baik jika dibandingkan dengan *Hierarchical Agglomerative clustering* (HAC), *Single-Pass clustering* dan *K-Nearest Neighbor clustering* (Sarkar, 2009).

Banyak peneliti mengidentifikasi *salient sentence* dengan meningkatkan metode *sentence rangking.* Hal ini juga diakui bahwa strategi pemilihan kalimat sangat penting dan bertujuan untuk mengurangi redundansi antara kalimat yang dipilih sehingga hasil ringkasan memungkinkan dapat mencakup lebih banyak konsep dari dokumen sumber. Beberapa metode telah dikembangkan antara lain (Ge dkk, 2011) mengurangi bobot dari kalimat yang mengandung *discourse connectors* (DC) seperti "*because*", "*as a result*", "*after*" dan "*before*".

Kalimat penting (*salient sentence)* penyusun sebuah ringkasan harus memiliki kepadatan informasi. *Salient sentence* harus mengandung informasi sebanyak mungkin dari dokumen sumber (He dkk, 2008). Menurut He dkk (2008) fitur kepadatan informasi kalimat *sentence information density* (SID) dapat digali dengan pendekatan *positional text graph.*

Penelitian Kruengkrai (2003) menunjukkan kombinasi relasi kalimat dan kata-kata penting pada penyusunan peringkat meningkatkan kemunkinan kesesuaian hasil ringkasan, tapi disisi lain metode ini juga memasukkan kalimat yang tidak relevan pada hasil ringkasan. Kalimat dapat menjadi penting jika kata-kata yang menyusun kalimat tersebut juga penting (Wan dkk, 2007).

Salah satu algoritma penyusunan peringkat yang berbasis *graph* yang popular diantaranya *LexRank* (Erkan dkk, 2004) atau *TextRank* (Mihalcea dkk,

2005) yang memanfaatkan kesamaan hubungan antara kalimat untuk membangun sebuah *graph*, dan menggunakan algoritma peringkat berbasis *graph* untuk memperoleh bobot peringkat dari suatu kalimat.

Dibidang *Natural Langguage Processing* (NLP) telah dikembangkan sebuah metode semantik parsing yang diberi nama *Semantic Role Labeling* (SRL) (Gildea dkk, 2001) yang dapat digunakan untuk mengidentifikasi argumen dari predikat dalam suatu kalimat, dan menentukan semantic role atau peran semantik.

Sebagian besar algoritma penyusunan peringkat berbasis *graph* menggunakan kalimat sebagai *bag of word* atau informasi sintaksis dalam dokumen teks, tapi mengabaikan informasi semantik. Manusia memahami kalimat berdasarkan peran semantik kata *("who" did "what" to "whom", "where", "when", and "how").* Informasi semantik lebih sesuai untuk menggambarkan persepsi manusia terhadap kalimat dibandingkan dengan kalimat sebagai *bag of word.*

Oleh karena itu pada tesis ini kami mengusulkan metode baru, strategi pemilihan kalimat representatif *cluster* yang diberi nama SSID (*Semantic Sentence Information Density*) dengan pendekatan position text graph dan peran semantik kata pada peringkasan multi-dokumen. Sehingga dapat meningkatkan kemungkinan kesesuaian hasil ringkasan dan menghilangkan kalimat yang tidak relevan pada hasil ringkasan.

1.1 Perumusan Masalah

Perumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana mendapatkan peran kata dari suatu kalimat?
2. Bagaimana mengkombinasikan peran kata dalam suatu kalimat dengan metode *position text graph*?

1.2 Batasan Masalah

1. Jenis peringkasan otomatis yang dibangun berbasiskan metode *extractive*.

2. Hasil ringkasan tidak mempertimbangkan urutan atau kesesuaian kalimat untuk kemudahan pembacaan.
3. Data yang dijadikan data uji adalah dataset dari *Document Understanding Conference* (DUC) 2004.
4. Pengujian kualitas hasil ringkasan dibatasi pada *Recall-Oriented Understudy for Gisting Evaluation* (ROUGE) yaitu ROUGE-1 dan ROUGE-2.

1.3 Tujuan dan Manfaat Penelitian

Tujuan dari penelitian ini adalah mengusulkan metode baru, strategi pemilihan kalimat representatif *cluster* yang diberi nama SSID (*Semantic Sentence Information Density*) dengan pendekatan *position text graph* dan peran semantik kata pada peringkasan multi-dokumen.

Manfaat dari penelitian ini adalah agar meningkatkan kualitas *salient* pada pemilihan kalimat representatif *cluster* dalam peringkasan multi-dokumen secara otomatis.

1.4 Kontribusi

Kontribusi pada penelitian ini adalah mengajukan metode baru dengan pendekatan semantik berupa peran semantik kata dan *position text graph* pada pemilihan kalimat representatif *cluster* dalam peringkasan multi-dokumen secara otomatis

BAB 2
DASAR TEORI DAN KAJIAN PUSTAKA

2.1 Dasar Teori

Dasar teori merupakan rangkuman semua teori yang digunakan sebagai pedoman dalam melakukan penelitian. Dasar teori yang digunakan meliputi peringkasan dokumen otomatis, *clustering, similarity sistogram cluster* (SHC), p*osition text graph*, peran semantik kata, pemilihan kalimat representatif.

2.2 Peringkasan Dokumen Otomatis

Peringkasan dokumen otomatis adalah suatu proses mereduksi ukuran dokumen dengan tetap menjaga isi semantik dari dokumen sumber (Cai dkk, 2011). Peringkasan dokumen otomatis adalah suatu proses menciptakan versi singkat dari suatu dokumen yang mampu menberikan informasi yang berguna bagi pengguna (Erkan dkk, 2004).

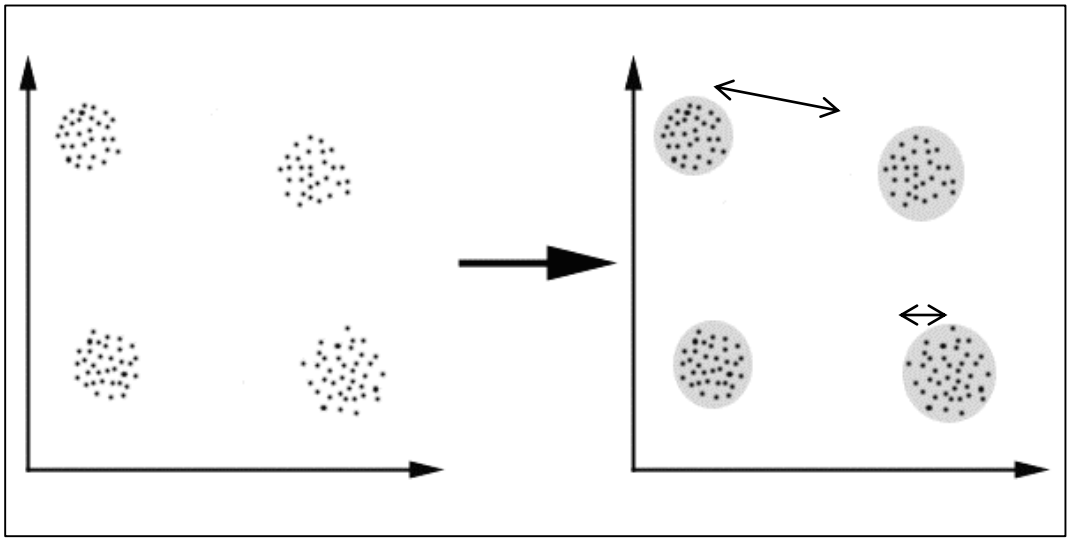
Terdapat dua metode dalam peringkasan dokumen *extractive* dan *abstractive*. Metode *extractive* adalah dengan memilih kalimat yang memiliki skor tertinggi dari dokumen asli dan menempatkannya bersama-sama untuk membentuk versi singkat dari dokumen tanpa memodifikasi teks asli (Meena dkk, 2014), sedangkan metode *abstractive* memungkinkan mengandung kata atau kalimat yang tidak terdapat dalam dokumen sumber (Kogilavani dkk, 2010).

Suatu ringkasan dokumen dapat berupa peringkasan umum (*generic*) atau berfokus pada query (*query-focused*) / *Topic-oriented*. *Topic-oriented* berorientasi pada topik yang diinginkan pengguna, dan mendapatkan informasi dari teks yang sesuai dengan topik yang diinginkan (Erkan dkk, 2004).

2.3 Clustering

Clustering merupakan algoritma pengelompokan sejumlah data menjadi kelompok-kelompok data tertentu yang serupa. C*lustering* dilakukan dengan mencari kesamana data atau karakteristik yang terdapat dalam data dan mengelompokkanya menjadi *cluster*. Metode *clustering* yang baik akan

menghasilkan *cluster* dengan kesamaan *inter-class* yang tinggi dan rendah terhadap kesamaan *intra-class* (Mann dkk, 2013).



Gambar 2.1 Ilustrasi clustering Data Intra-class dan Inter-class

Algoritma *clustering* dibagi menjadi beberapa diantaranya (Mann dkk, 2013):

1. Algoritma *Partitioning* membagi data kedalam *k* partisi, di mana setiap partisi merupakan *cluster*. Partisi dilakukan berdasarkan tujuan dan fungsi tertentu.
2. Algoritma *Hierarchical* adalah teknik pengelompokan data yang membagi dataset serupa dengan hirarki *cluster*. *Hierarchical clustering* dibagi menjadi *Agglomerative Nesting* dan *Divisive Analysis*.
3. *Agglomerative Nesting* dikenal sebagai AGNES. Metode ini membangun pohon *cluster node* dengan pendekatan *bottom-up*.
4. *Devise Analysis* dikenal sebagai DIANA. Metode ini membangun pohon *cluster* node dengan pendekatan top-down.
5. Algoritma *Density Based* sebuah algoritma *clustering* yang dikembangkan berdasarkan tingkat kerapatan data (*density-based*). Dimana algoritma ini menumbuhkan daerah yang memiliki kerapatan tinggi menjadi *cluster*,

6. Algoritma *Grid Density Based* sebuah algoritma *clustering* yang merupakan pengembangan dari algoritma *Density Based* dengan grid model.

2.4 Similarity Histogram Cluster (SHC)

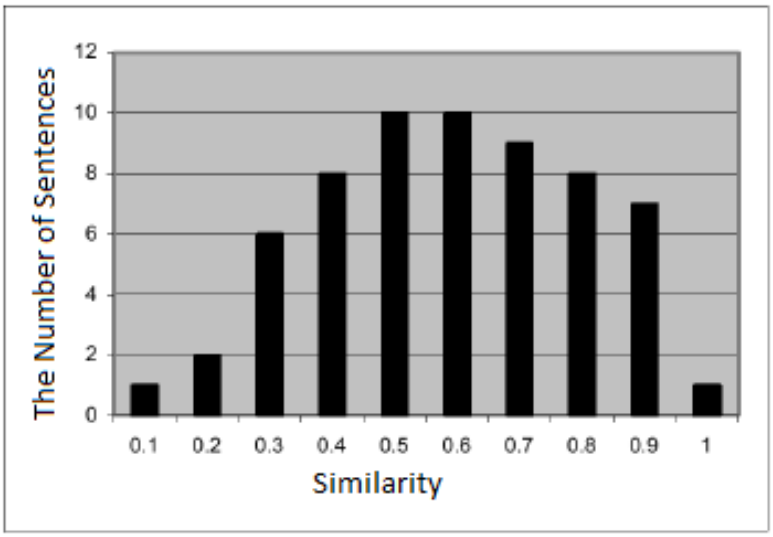
SHC merupakan suatu metode pembuatan *cluster* secara dinamis dari hasil adopsi model *cluster* yang tumpang tindih. Konsep utama dalam SHC adalah untuk menjaga *coherency* yang tinggi pada setiap *cluster* (Hammouda dkk, 2003). *Similarity Histogram Cluster* merupakan representasi statistik dari suatu distribusi similaritas pasangan antar anggota yang ada pada suatu *cluster*. Jumlah dari *bin* dalam histogram menunjukkan interval nilai similaritas tertentu.Namun *vector* yang menunjukkan similaritas dari kalimat-kalimat menjadi sangat jarang (*too sparse*) karena kalimat terlalu pendek ketika dibandingkan satu dengan yang lain. Sehingga Sarkar (2009) menggunakan *uni-gram matching-based similarity measure* ditunjukkan pada Persamaan 2.1:

$$\text{sim}(s_i, s_j) = \frac{(2*|s_i| \cap |s_j|)}{|s_i|+|s_j|} \quad (2.1)$$

dimana s_i dan s_j adalah kalimat *s* ke-i dan ke-j. Selanjutnya $|s_i| \cap |s_j|$ merepresentasikan jumlah dari kata-kata yang sesuai antara kalimat *s* ke-*i* dan kalimat *s* ke-*j*. $|s_i|$ adalah panjang kalimat *s* ke-*i* yaitu jumlah kata yang menyusun kalimat tersebut. Metode *unigram matching based similarity measure* adalah metode yang digunakan untuk mengukur similarity untuk setiap pasangan kalimat pada cluster dan kandidat anggota cluster baru dalam SHC.

Data yang akan ditambahkan ke dalam *cluster* dibandingkan terhadap seluruh *histogram cluster*, dan jika menurunkan distribusi *coherency*, maka data itu tidak ditambahkan, jika tidak maka akan ditambahkan. SHC berarti menjaga

distribusi similaritas agar cenderung ke kanan (ke arah nilai 1 yaitu nilai similaritas terbesar) untuk menjaga *coherency* yang tinggi pada setiap *cluster*.



Gambar 2.2 Histogram Rasio pada Cluster (Sarkar, 2009)

Kualitas hubungan antar *cluster* dinilai dengan menghitung rasio jumlah kesamaan diatas *similarity threshold* S_T terhadap total kesamaan. Jika n_c adalah jumlah dari kalimat pada suatu *cluster*, maka jumlah dari pasangan kalimat yang ada pada *cluster* tersebut adalah $m_c=n_c(n_c+1)/2$, dimana $S=\{s_i:i=1,...,m_c\}$ adalah himpunan kesamaan pada *cluster*. Similarity histogram dari *cluster* dinotasikan dengan $H= \{h_1, h_2, h_3, ..., h_{nb}\}$. Jumlah dari *bin* yang ada pada suatu histogram dinotasikan dengan n_b sedangkan jumlah similaritas kalimat yang ada pada *bin* ke-*i* dinotasikan dengan h_i. Fungsi untuk menghitung h_i ditunjukkan pada Persamaan (2.2).

$$h_i = \text{count}(\text{sim}_j) \text{Untuk } \text{sim}_{\text{li}} \leq \text{sim}_j \leq \text{sim}_{\text{ui}}, \quad (2.2)$$

Sim_{li} adalah batas bawah similarity pada bin ke-i sedangkan sim_{ui} adalah batas atas similaritas pada *bin* ke-*i*. Histogram Ratio (HR) dari suatu *cluster* dapat dihitung dengan Persamaan (2.3).

$$\text{HR} = \frac{\sum_{i=T}^{n_b} h_i}{\sum_{j=1}^{n_b} h_j} \quad (2.3)$$

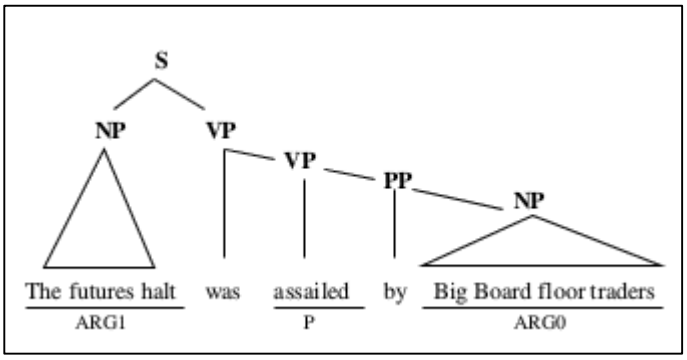
$$T = \lfloor S_T * n_b \rfloor \quad (2.4)$$

S_T adalah similarity threshold. Persamaan (2.4) menunjukkan jumlah bin yang sesuai dengan *similarity threshold* yang dinotasikan dengan *T*.

2.5 Peran Semantik Kata

Dalam kebanyakan metode yang ada, cara yang digunakan untuk mendapatkan *salient sentence* adalah dengan *sentence ordering*. Kalimat akan dianggap penting jika mengandung banyak kata yang penting atau terletak di *top position*. Namun, fitur seperti ini tidak mencakup informasi semantik apapun. Oleh karena itu pada penelitian ini digunakan *Semantic Role Labeling* (SRL) untuk melakukan extraksi terhadap peran semantik kata dalam kalimat. *Semantic Role Labeling* merupakan proses pengidentifikasian argumen dari predikat dalam suatu kalimat, dan menentukan *semantic role* atau peran semantik.

SRL adalah suatu proses yang digunakan untuk menentukan hubungan peran semantik antar kata dalam suatu kalimat. Pada (Baker dkk, 1998) mengemukakan gagasan *Framenet* untuk menggambarkan fungsi tata bahasa, Jenis frase, dan sifat-sifat sintaksis lainnya. *Framenet* difokuskan pada *frame semantic* yang merepresentasikan skematis dari situasi yang digambarkan suatu kalimat.

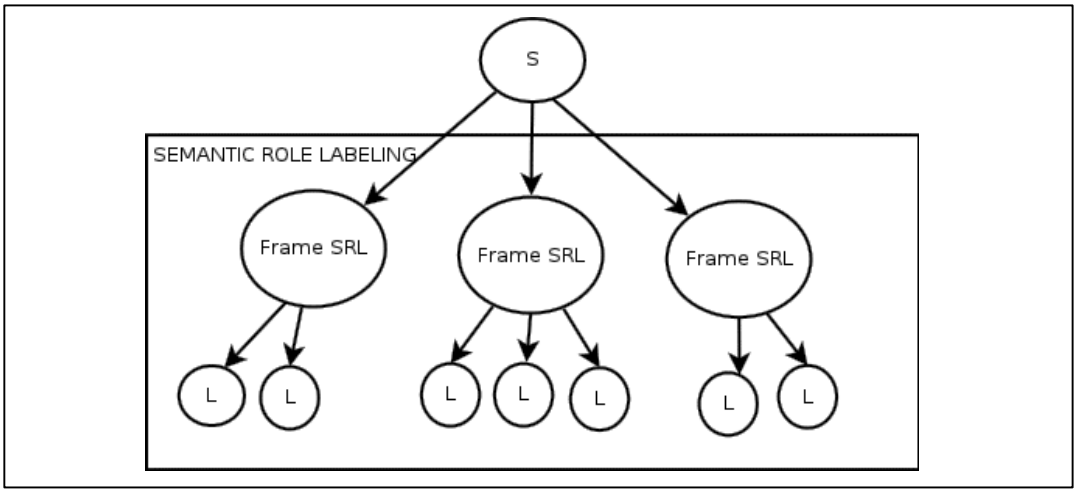


Gambar 2.3 Proses *Sematic Role Labeling* (Guildea dkk, 2009)

Framenet dikembangkan dengan menggunakan training dan mengambil fitur leksikal, kata kerja, kata benda dan fitur-fitur lain serta hubungan tata bahasa (Gildea dkk, 2002), Surdeanu dkk (2003) menambahkan fitur pendeteksi struktur predikat-argumen, Pradhan dkk (2004) menambahkan *support vector machine* pada bagian klasifikasinya untuk meningkatkan kinerja.

Bjorkelund (2009) menambahkan kontribusi dengan menambahkan klasifikasi untuk mengidentifikasi arti dari predikat, arti dari argument dan label dari argument yang kemudian didemontrasikan dalam penelitian Bjorkelund dkk (2010).

Pada penelitian lain Palmer dkk (2005) mendiskusikan peran semantik yang digunakan untuk proses anotasi yang disebut *proposition bank* (*Probank*) dengan pendekatan penyediaan data statistik untuk memberikan penjelasan pada setiap klausa *treeBank*.



Gambar 2.4 Extraksi Kalimat dengan *Semantic Role Labeling*

Semantic Role Labeling dapat dibagi menjadi dua sub tasks utama, yaitu mengidentifikasi *Frame SRL* dalam kalimat dan melabeli kata pada *Frame SRL* dengan label yang semantik yang telah disediakan SRL.

Tabel 2.1 Contoh Hasil Extraksi Kalimat dengan SRL

	SRL	**POS**
John	giver [A0]	S1 (S (NP (NNP John))
gave	V : give	(VP (VBD Gave)
Marry	entity given to [A2]	(NP (NNP Mary))
the	thing given [A1]	(NP (DT the)
book		(NN book)))

Pada Table 2.1 dimana terdapat kalimat "*John gave Marry the book*" pada proses *semantic rule labeling* tiap kata pada kalimat tersebut diberi label berdasarkan P*art of Speech* (POS). Kalimat "*John gave Marry the book*" menjadi *S1 (S (NP (NNP John))(VP (VBD Gave)(NP (NNP Mary))(NN book)))*. Dari struktur *pos part of speech* (POS) kemudian digunakan sebagai acuan untuk mencari struktur sintaksis kalimat atau label semantik pada *corpus* Treebank. John adalah [A0] subjek dari *give* , *gave* adalah *verb* (V), Marry adalah [A2] objek kedua, sedangkan *the book* adalah [A1] subjek pertama (benda yang diberikan).

Tabel 2.2 Label yang Digunakan pada SRL

Label	**Keterangan**	**Peran**
V	*Verb*	Predicate
AM-PRD	*secondary predicate*	
A0	*Subject*	who
A1	*Object*	Whom
A2	*Indirect Object*	
AM-ADV	*adverbial modification*	how
AM-MNR	*Manner*	
AM-DIR	*direction*	Where
AM-LOC	*location*	Where
AM-DIS	*discourse marker*	What
AM-EXT	*extent*	
AM-MOD	*general modification*	
AM-PNC	*proper noun component*	
AM-NEG	*negation*	why
AM-PRC	*purpose*	
AM-REC	*reciprocal*	

Label	Keterangan	Peran
AM-TMP	*temporal*	when

Beberapa label yang ada pada *semantic rule labeling* (SRL) seperti terdapat pada Tabel 2.2 diantaranya *V* yang berarti *Verb*(kata kerja) yang kemudian dipahami dengan *how(bagaimana),* AM-DIR adalah *direction* dan AM-LOC adalah *location* yang kemudian dikelompokkan sebagai *where*(kemana). Pengelompokan label ini penting karena beberapa label dapat mempunyai kata yang sama contohnya adalah AM-DIR dan AM-LOC.

Sebuah predikat dalam sebuah kalimat biasanya merupakan peristiwa atau tindakan. Peran semantik lebih terkait dalam memberikan informasi yang berguna seperti "siapa", "apa", "kapan", "di mana", "mengapa", dan bagaimana (Yan dkk, 20014). Contoh kalimat "*Bayern Munich's Robert Lewandowski has entered the record books with the quickest hat-trick in Bundesliga which he set the record for the fastest four goal and win (S11)*".

Pada penelitian ini hasil extraksi dari SRL akan digunakan sebagai penentu dari peran kata yang telah di lakukan berdasarkan Tabel 2.1, yang selanjutnya peran kata akan dinotasikan *Who (Args$_1$), Predicate (Args$_2$), Whom (Args$_3$), What (Args$_4$), When (Args$_5$), Where (Args$_6$), Why (Args$_7$).*

Tabel 2.3 Contoh Hasil Extraksi Peran Kata dalam Kalimat

		Who (Args$_1$)	***Predicate (Args$_2$)***	***Whom (Args$_3$)***	***What(Args$_4$)***
S	*FRAME*$_1$	Bayern Munich Robert Lewandowski	entered	the record books	with the quickest hat-trick in Bundesliga which he set the record for the fastest four goal and win
	FRAME$_2$	Which he	set	record for the fastest	

		Who (Args$_1$)	***Predicate (Args$_2$)***	***Whom (Args$_3$)***	***What(Args$_4$)***
	FRAME$_3$	he	win		

2.6 Pemilihan Kalimat Representatif *Cluster* dengan *Position Text graph*

Pada pemilihan kalimat representatif hasil extraksi *Sematic Rule Labeling* digunakan sebagai penghitungan skor kalimat. Metode *extractive* melibatkan menugaskan skor *salienct sentence* dari teks (misalnya kalimat, paragraf) dalam dokumen dan penggalian informasi teks yang mempunyai skor tertinggi. Skor biasanya didapatkan dengan *fusion information* (Barzillay dkk, 1999), kompresi kalimat (Knight dkk, 2002), Kogilavani (2010) menggunakan fitur *sentence profile feature* yang merupakan kombinasi dari bebrapa fitur dalam kalimat.

Hasil extraksi dan segmentasi kalimat pada Tabel 2.3 digambarkan sebagai *N* adalah jumlah *Frame SRL* hasil segmentasi dari kalimat dengan SRL dan S_i=*{FRAME$_1$,FRAME$_2$,.....,FRAME$_N$}* dimana S_i adalah kalimat ke-i dalam *cluster* dengan *FRAME$_n$ ={Args$_1$,Args$_2$,Args$_3$,Args$_4$,Args$_5$,Args$_6$,Args$_7$}*, *Args* adalah argumen yang didapat dari extraksi kalimat. Pada pengukuran jarak similaritas antara kalimat digunakan *Jaccard Coefficient* (2.5).

$$SimFSRL(FSRL_i, SFRL_k) = \frac{C(Arg_i) \cap C(Arg_k)}{C(Arg_i) \cup C(Arg_k)} \quad (2.5)$$

$$SimS(S_g, S_h) = \sum_n \quad SimFSRL(FSRL_{nj}, FSRL_{nk}) \quad (2.6)$$

Position Text Graph dikemukakan oleh He dkk (2011) digunakan untuk mendapatkan kepadatan informasi dalam kalimat. Pada penelitian He dkk (2011) kesamaan setiap kalimat dalam dokumen dihitung dengan menggunakan *cosine simmilarity* sehingga membentuk matrik kesamaan yang digunakan untuk membangun p*osition text graph. Graph* digambarkan sebagai *P= (V,E)*, dimana P merepresentasikan *graph*, *V={S$_1$,S$_2$, ... S$_n$}* adalah *vertex* pada *graph* yang

merepresentasikan kalimat-kalimat dalam suatu *cluster*. *Graph* dibangun dengan cara sebagai berikut: *graph* P dibangun berdasarkan kalimat-kalimat dalam *cluster*. Saat pertama *graph* P adalah kosong setelah itu semua kalimat dalam suatu *cluster* dimasukan sebagai vertex. Langkah kedua hitung nilai similarity untuk setiap pasangan kalimat dalam P, jika nilai similarity suatu pasangan kalimat memenuhi *threshold* α maka *edge* dibentuk dan bobot pasangan kalimat tersebut adalah nilai *similarity* yang dimilikinya. Ketika *graph* telah dibangun, fitur *sentence information density* dihitung dengan Persamaan (2.7) berikut:

$$F_{\text{sid}}(s_{\text{kj}}) = \frac{W_{s_{\text{kj}}}}{\max\limits_{l \in \{1,2,..n\}} W_{s_{\text{lj}}}} \tag{2.7}$$

Jumlah kalimat *s* pada cluster ke-*j* ditunjukkan dengan n, $W_{s_{\text{kj}}}$ adalah penjumlahan bobot dari semua edge yang datang dari kalimat *s* ke-*k* pada *cluster* ke-*j*, sedangkan Persamaan 2.8 adalah bobot *edge* maksimum diantara semua kalimat yang ada pada *cluster* ke-j.

$$\max_{l \in \{1,2,..n\}} W_{s_{\text{lj}}} \tag{2.8}$$

BAB 3
METODE PENELITIAN

Secara umum, penelitian ini diawali dengan studi literatur, analisis data, desain sistem, implementasi, serta diakhiri dengan uji coba. Sedangkan penulisan laporan penelitian dimulai pada awal sampai akhir penelitian. Secara detail, penelitian ini dirancang dengan urutan sebagai berikut.

3.1 Studi Literatur

Pada penelitian ini digunakan berbagai referensi sebagai bahan pendukung untuk menerapkan metode yang diusulkan. Studi literatur dilakukan untuk mendapatkan informasi dari berbagai literatur yang akan digunakan, serta metode yang pernah dipelajari sebelumnya.

Studi literatur yang dilakukan diharapkan dapat memberikan data, informasi, dan fakta mengenai peringkasan multi-dokumen yang akan dikembangkan. Studi literatur yang dilakukan mencakup pencarian dan mempelajari referensi-referensi yang terkait, seperti:

1. *Text preprocessing* yaitu *segmentation* (kata dan kalimat), *stopword removal* dan *stemming* (*English Porter Stemmer*).
2. Metode *uni-gram matching based similarity* dan metode *SHC* untuk *clustering* kalimat.
3. Metode pengurutan *cluster* berdasarkan *cluster importance*.
4. Ekstraksi fitur dari kalimat berdasarkan fitur *sentence information density*.
5. *Semantic Rule labeling* (SRL).
6. Metode evaluasi hasil ringkasan dengan ROUGE-1 dan ROUGE-2.

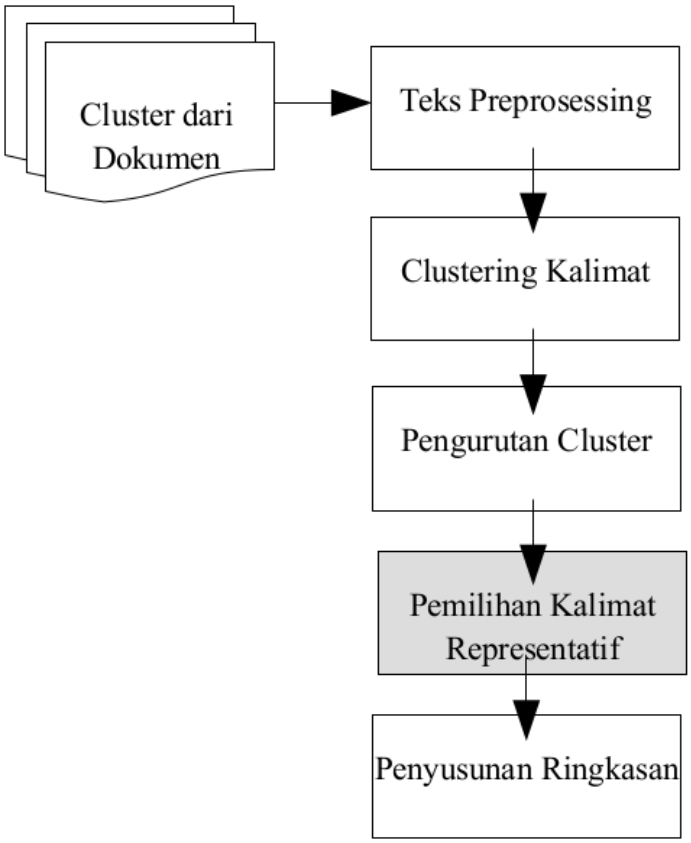
3.2 Analisa Data

Dataset yang digunakan adalah DUC 2004 Task 2. Dataset DUC 2004 Task 2 yang merupakan kumpulan dokumen berita dalam bahasa Inggris dari *Associated Press* dan *New York Times*. Terdiri dari 500 dokumen. Dokumen-dokumen tersebut telah terbagi ke dalam kelompok-kelompok menjadi 50 *cluster* dokumen. Setiap

cluster dokumen terdiri dari rata-rata 10 dokumen berita. Dataset DUC 2004 Task 2 dapat diunduh pada alamat http://duc.nist.gov/duc2004/tasks.htm.

3.3 Desain Model Sistem

Desain model sistem yang digunakan diadopsi dari *framework* Sarkar (2009) yang ditunjukkan pada Gambar 3.1. Pertama data yang telah berupa *cluster* dari dokumen dilakukan *text preprosessing* untuk memudahkan dalam pengolahan data selain itu juga mengekstraksi kata dan kalimat yang berada pada tiap dokumen, tahapan selanjutnya adalah data dari hasil preprosessing dibentuk menjadi *cluster-cluster* kalimat dengan menggunakan *Similarity Histogram Cluster* (SHC).



Gambar 3.1 Desain Model Sistem

Fase clustering digunakan untuk mengambil topik-topik yang berada pada dokumen. Fase selanjutnya adalah pengurutan *cluster* berdasarkan *cluster important* yang dibahas pada bab sub-bab 3.2.2 sehingga *cluster* dengan yang

paling tinggi nilainya adalah *cluster* yang mengandung banyak kalimat dalam satu *cluster* dan mengandung kata penting.

Proses pada fase selanjutnya adalah memilih kalimat yang menjadi perwakilan dari tiap *cluster* atau bisa disebut juga pemilihan kalimat representatif. Pada fase inilah penulis mengajukan metode baru yang lebih detailnya bisa di lihat di sub sub-bab 3.2.4. Fase terakhir dalah penyusunan kalimat ringkasan dimana kalimat perwakilan dari tiap-tiap *cluster* disusun berdasankan dari bobot *cluster*.

3.3.1. Fase Teks Preprosessing

Sebelum data diolah lebih lanjut menjadi *cluster-cluster* kalimat, diperlukan pengolahan awal. Dataset yang digunakan untuk uji coba sistem adalah *Document Understanding Conference* (DUC). DUC merupakan dataset standar untuk menguji sistem peringkasan otomatis khususnya peringkasan multi-dokumen. Pada penelitian ini edisi DUC yang dipilih adalah 2004. Dokumen-dokumen berita yang terdapat pada DUC 2004 adalah dokumen dengan format XML sederhana, sehingga pada proses ekstraksi konten dokumen digunakan teknik XML parsing. Proses-proses yang dilalui dalam *teks preprosessing* adalah *segmentation*, *stopword removal*, dan *stemming*.

Pada penelitian ini segmentasi dilakukan terhadap kata dan kalimat. Segmentasi kata digunakan untuk membedakan kata, spasi dan tanda baca sehingga tanda baca bisa dihilangkan sedangkan segmentasi kalimat digunakan untuk mendapatkan kalimat-kalimat penyusun dari dokumen sumber. Peneliti menggunaka parsing XML untuk mengambil data konten pada dataset DUC 2004 dan menggunakan library S*tanford Natural Language Processing* untuk medapatkan kata dan kalimat dari konten berita.

Kalimat dari hasil segmentasi selanjutnya diproses dengan s*topword removal* untuk menghilangkan kata-kata yang kurang penting. Beberapa contoh kata diantaranya *'is', 'are', 'and'*, kamus *stopword* yang digunakan dari Stanfordnlp bisa diunduh di https://github.com/stanfordnlp/CoreNLP/blob/master-/data/edu/stanford/nlp/patterns/surface/stopwords.txt.

Stemming dilakukan untuk mendapatkan kata dasar dari semua kata-kata penyusun dokumen. Algoritma *stemming* yang digunakan dalam penelitian ini

adalah algoritma *Porter Stemmer*. Pada penelitian ini digunakan *Library Porter Stemmer* untuk melakukan stemming bahasa Inggris. *Librar*y tersebut dapat diunduh secara langsung di http://snowball.tartarus.org/dist/libstemmer_java.tgz. Selanjutnya kalimat hasil preprosessing disimpan dalam file.

Pada preprosessing ini dilakukan extraksi *Semantic Rule labeling* dari setiap kalimat yang kemudian disimpan dan akan digunakan pada pemilihan kalimat representative.

3.3.2. Fase C*lustering* Kalimat

Data dari hasil fase preprosessing selanjutnya dilakukan *clustering* berdasarkan SHC, dimana SHC diadopsi dari penelitian Sarkar (2009). Kalimat dari fase preprosessing dilakukan proses perhitungan kesamaan antar kalimat. *Cosine Similarity* merupakan metode pengukuran yang sering digunakan pada kasus *clustering* dan peringkasan (Erkan dkk, 2004). Kalimat direpresentasikan ke dalam bobot *vector* ketika menghitung *cosine similarity*. Namun fitur *vector* yang menunjukkan similaritas dari kalimat-kalimat menjadi sangat jarang (*too sparse*) karena kalimat terlalu pendek ketika dibandingkan satu dengan yang lain. Sarkar (2009) menggunakan *uni-gram matching-based similarity measure* dalam pengukuran similaritas antar kalimat ditunjukkan pada Persamaan (2.1).

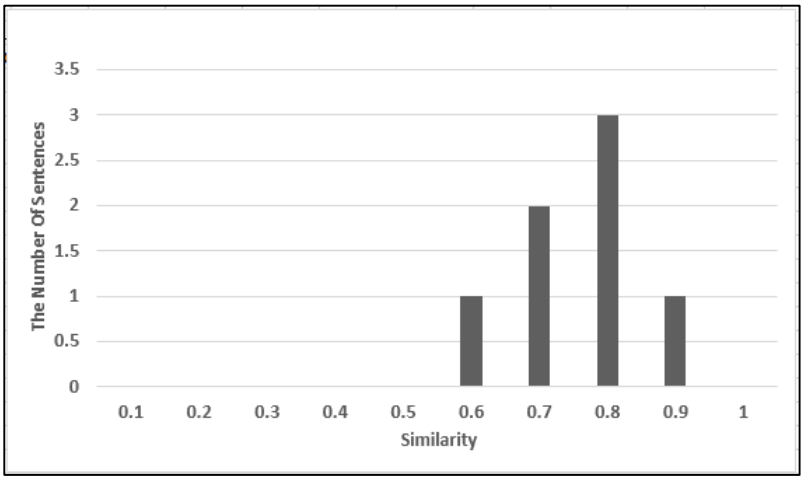
Konsep utama dari SHC adalah menjaga setiap *cluster* sedapat mungkin berada dalam kondisi *koherent* pada tingkat yang baik. Pendekatan yang terdapat dalam algoritma SHC adalah pendekatan *incremental dynamic method* untuk membangun *cluster-cluster* kalimat. Kalimat-kalimat diproses sekali dalam satu waktu dan secara bertahap dimasukkan ke dalam masing-masing *cluster* yang sesuai ketika proses *clustering*.

Tingkat *koherent cluster* dimonitor dengan *Similarity Histogram Cluster.* Kualiatas hubungan antar *cluster* dinilai dengan menghitung rasio jumlah kesamaan diatas *similarity threshold* S_T terhadap total kesamaan. Jika n_c adalah jumlah dari kalimat pada suatu *cluster*, maka jumlah dari pasangan kalimat yang ada pada *cluster* tersebut adalah $m_c=n_c(n_c+1)/2$,dimana $S=\{s_i:i=1,...,m_c\}$ adalah himpunan

kesamaan pada *cluster*. *Similarity histogram* dari *cluster* dinotasikan dengan $H=\{h_1, h_2, h_3, ..., h_{nb}\}$.

```
1: N ← Empty List {Cluster List}
2: for each sentence s do
3:    for each cluster c in N do
4:          HR_old = HR_C
5:          Simulate adding s to c
6:          HR_new = HR_C
7:          if (HR_new ≥ HR_old) OR ((HR_new ≥ HR_min)
            AND (HR_old-HR_new < ε)) then
8:                Add s to c
9:                exit
10:         end if
11:   end for
12:   if s was not added to any cluster then
13:         Create a new cluster c
14:         ADD s to c
15:         ADD c to L
16:   end if
17: end for
```

Gambar 3.2 Algoritma SHC



Gambar 3.3 Histogram pada Cluster SHC

Jumlah dari *bin* yang ada pada suatu *histogram* dinotasikan dengan n_b sedangkan jumlah *similarity* kalimat yang ada pada *bin* ke-*i* dinotasikan dengan h_i. Fungsi untuk menghitung h_i ditunjukkan pada Persamaan (2.2) dimana sim_{li} adalah

batas bawah *similarity* pada *bin* ke-*i* sedangkan sim_{ui} adalah batas atas *similarity* pada *bin* ke-*i*.

Histogram Ratio (*HR*) dari suatu *cluster* dapat dihitung dengan Persamaan (2.3). Setelah fase *clustering* kalimat dengan SHC berhasil maka selanjutnya dilakukan fase pengurutan *cluster* berdasarkan *cluster importance* untuk menentukan tingkat pentingnya suatu *cluster* sebagai kandidat penyusun ringkasan.

3.3.3. Fase Pengurutan *Cluster*

Salah satu metode sederhana dalam pengurutan *cluster* adalah dengan menghitung kalimat yang terdapat dalam *cluster* dengan asumsi bahwa *cluster* yang mempunyai paling banyak kalimat adalah *cluster* yang lebih penting dari *cluster* lain, tapi metode ini tidak berjalan baik ketika:

1. Beberapa *top cluster* memiliki ukuran yang sama
2. *Cluster* terdiri dari kalimat-kalimat pendek yang kurang informatif sehingga hanya meningkatkan ukurannya bukan isinya.

Untuk menatasi masalah ini maka diperlukan pengurutan *cluster* berdasarkan *cluster important* (Sarkar, 2009). Jika frekuensi suatu kata w(*count(w)*) memenuhi *threshold* θ maka kata tersebut adalah kata *frequent*. Dengan cara mengukur kata bobot *cluster* berdasarkan kata-kata penting yang terdapat didalamnya. Pentingnya *cluster* dihitung dengan Persamaan 3.1.

$$\mathrm{W}(c_j) = \sum_{w \in c_j} \log(1 + \mathrm{count}(w)) \qquad (3.1)$$

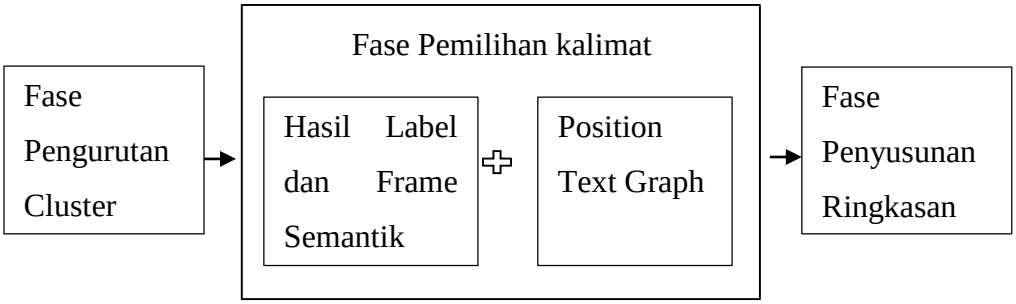
Dimana bobot *cluster c* ke-j dinotasikan dengan $W(c_j)$ dan *count(w)* adalah jumlah kata pada koleksi *input* dan *count (w)* lebih dari *threshold* θ. Pentingnya kata diukur dengan nilai *log* yang menormalisasikan jumlah total kata pada kumpulan seluruh dokumen *input* setelah dilakukan proses *stopwords removal.*

Setelah melakukan pengurutan *cluster* secara *descending*, *top cluster* dipilih sebagai kandidat topik yang mewakili topik-topik dari dokumen *input* dan selanjutnya dilakukan pemilihan kalimat representati yang bisa mewakili *top cluster*.

3.3.4. Fase Pemilihan Kalimat Representatif

Pada fase ini dilakukan kombinasi antara metode *postion text graph* dan peran semantik yang terdapat pada kalimat yang dibandingkan Pemilihan kalimat representatif tersebut ditentukan berdasarkan skor kalimat. Skor kalimat dihitung berdasarkan kombinasi metode yang ada pada penelitian ini yaitu *postion text graph* dan peran semantik kata.

Setiap kalimat pada diubah berdasarkan hasil extraksi *Semantic Rule Labeling*. Setiap kalimat ditransformasikan dan diberi label berdasarkan hasil pengolahan *Semantic Rule Labeling* (SRL) pada fase teks preprosessing. Sebagai contoh dimana C_1 adalah hasil *cluster* dari fase 1.2.4 dimana $C_1 = \{S_1,.....S_n\}$, kalimat digambarkan dengan *S* dan *n* adalah jumlah anggota dalam *cluster* C_1. Dilakukan extraksi terhadap setiap kalimat *S* dalam *cluster* kemudian hasil ektraksi peran semantik dijadikan fitur tambahan dalam melakukan perhitungan *position text graph*.

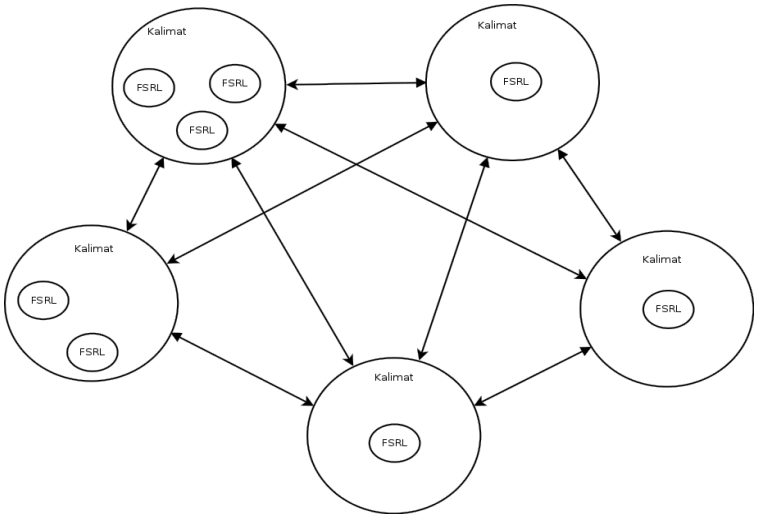


Gambar 3.4 Model Kontribusi yang Diajukan

Hasil ektraksi fitur peran semantik ini didapat dua fitur. Pertama kalimat diekstrak berdasarkan topik bahasan (*Frame SRL*) dalam suatu kalimat dengan menggunakan jumlah predikat atau kata kerja dari kalimat tersebut. Kedua, label *Frame SRL* dari hasil proses dipetakan terhadap *("who" did "what" to "whom", "where", "when", and "how")* yang selanjutnya disebut sebagai peran semantik kata.

Ektraksi SRL dilakukan untuk mendapatkan bahasan dari kalimat berdasarkan jumlah predikat yang ditunjukkan oleh proses pelabelan SRL dimana hasilnya dapat digambarkan $S = \{FRAME_1, FRAME_2, ..., L_m\}$ dimana *m* adalah jumlah predikat dan *FRAME* adalah *Frame SRL*.

Langkah selanjutnya adalah menkombinasikan hasil extraksi dari SRL dan metode *Position text graph* yang dibangun berdasarkan konsep similaritas antar setiap kalimat yang berada di dalam *cluster*. Setiap kalimat yang ada dianalogikan sebagai sebuah *vertex* dan hubungan similarity antar kalimat dinyatakan dengan *edge* yang menghubungkan kalimat-kalimat tersebut. *Edge* antar kalimat terbentuk jika similaritas antara dua kalimat lebih besar dari *threshold* (α) yang ditentukan. Pada pengukuran jarak similaritas antara kalimat digunakan pengukuran *Jaccard coefficient* (2.5) dan dilanjutkan dengan Persamaan (2.6). Kemudian bobot dari tiap *edge* dengan Persamaan (2.7).



Gambar 3.5 Ilustrasi G*raph* Hasil Peran Semantik dan *Position Text graph*.

Sebagai contoh terdapat *cluster* C_1 sebagai hasil dari tahap fase SHC *clustering*:

Cluster* kalimat C1:** *Bayern Munich's Robert Lewandowski has entered the record books with the quickest hat-trick in Bundesliga which he set the record for the fastest four goal and win* ***(d11)*. Bayern Munich has entered the record books hat-trick in Bundesliga* ***(d29)****. Bayern Munich win last match versus Wolfsburg in Bundesliga history* ***(d34)****. Robert Lewandowski has quickest hat-trick in Bundesliga history* ***(d43)****.*

Cluster kalimat C_1 mempunya empat anggota kalimat ***(d11), (d29), (d34)*** dan ***(d43)*** dinotasikan dengan $C_1 = \{S_1, S_2, S_3, S_4\}$. Pertama tiap kalimat pada *cluster* C_1 dilakukan extraksi *Frame SRL* dengan *SRL* untuk mendapatkan topik bahasan

dari tiap kalimat. Anggota dari *cluster* digambarkan dengan S_n= { $FRAME_1$, $FRAME_2$, ..., $FRAME_m$}, dimana $_m$ adalah jumlah predikat hasil proses pelabelan SRL. Selanjutnya dari *Frame SRL* dilakukan pelabelan peranan semantik dan dipetakan ke *Who* ($Args_1$), *Predicate* ($Args_2$), *Whom* ($Args_3$), *What* ($Args_4$), *When* ($Args_5$), *Where* ($Args_6$), *Why* ($Args_7$) berdasarkan Tabel 2.1.

Pada Tabel 3.1 kalimat d11 mempunya tiga buah frameSRL diantaranya $Frame_1$, $Frame_2$ dan $Frame_2$. Pada kalimat d29 mempunyai satu frame. Sedangkan tiap frame mempunyai anggota peran kata. Contoh pada Tabel 3.1 $Frame_1$ pada kalimat **d11** mempunyai tiga peran kata yaitu {*Robert Lewandowski*} sebagai *who*, { *has* } sebagai *predicate* dan { *quickest hat-trick in Bundesliga history* } dengan label *whom*

Tabel 3.1 Contoh Hasil Extraksi Peran Semantik Kata dari *Cluster* kalimat C1

Kalimat	**Frame**	**Peran Kata**				
		Who	**Predicate**	**Whom**	**What**	**Where**
d11	**$Frame_1$**	*Bayern Munich Robert Lewandowski*	*entered*	*the record books*	*with the quickest hat-trick in Bundesliga which he set the record for the fastest four goal and win*	
	$Frame_2$	*Which he*	*set*	*record for the fastest*		
	$Frame_3$	*he*	*win*			

Kalimat	Frame	Peran Kata				
		Who	Predicate	Whom	What	Where
d29	**Frame$_1$**	*Bayern Munich*	*entered*	*he record books hat-trick in Bundesliga*		
d34	***Frame$_1$***	*Bayern Munich*	*win*	*last match versus Wolfsburg in Bundesliga history*		
d43	***Frame$_1$***	*Robert Lewandowski*	*has*	*quickest hat-trick in Bundesliga history*		

Hasil extraksi SRL yang berupa *Frame SRL* dan peran semantik dijadikan fitur yang akan digunakan dalam penghitungan *position text graph.* Pada perhitungan awal perbandingan peran semantik dimana tiap argumen dibangdingkan dengan kata atau kalimat yang mempunyai argumen yang sama. Pada perbandingan *Args$_1$* sampai *Args$_7$* digunakan *Jaccard coefficient* dengan sedikit midifikasi dan *threshold.*

$$SimArgs(Args_{si}, Args_{di}) = \frac{C(Args_{si}) \cap C(Args_{di})}{C(Args_{s1}) \cup C(Args_{d1})} \quad (3.2)$$

Dimana Args$_{si}$ adalah argumen dari hasil extraksi *srl* ke-s pada argumen ke-i jika *SimArgs(Args$_{si}$,Args$_{di}$)* lebih besar dari *threshold T* maka *SimArgs(Args$_{si}$,Args$_{di}$)=1* jika tidak *SimArgs(Args$_{si}$,Args$_{di}$)=0.* Kemudian dilakukan penghitungan skor kalimat dengan Persamaan *2.5* dengan jumlah argumen yang mempunyai kesamaan 1 dan 0, dilanjutkan dengan menggunakan Persamaan *2,6* untuk menukur similaritas dari *FrameSRL* dan Persamaan *2,7* yang

digunakan untuk mengetahui kepadatan informasi dari kalimat yang dihitung. Sehingga didapat hasil skor akhir tiap kalimat dalam C_1.

Pada Table 3.2 jarak kalimat dihitung berdasarkan jarak antar frameSRL dari kalimat tersebut sehingga jarak **d29** dan **d11** merupakan penambahan jarak keseluruhan frame berdasarkan extraksi Tabel 3.1. Pada Tabel 3.2 tanda (--) merupakan jarak yang tidak dihitung karena merupakan jarak terhadap kalimat itu sendiri. Pada Table 3.2 disimpulkan bahwa kalimat d11 yang cocok untuk mewakili ***Cluster* kalimat C1**.

Tabel 3.2 Contoh Hasil Perhitungan jarak antar kalimat berdasarkan Semantic Sentence Information Density (SSID)

Kalimat		**d11**			**d29**	**d34**	**d43**	**SSID**
	Frame	**Frame$_1$**	**Frame$_2$**	**Frame$_3$**	**Frame$_1$**	**Frame$_1$**	**Frame$_1$**	
d11	**Frame$_1$**	--	--	--	0.60	0.30	0.43	**1.67**
	Frame$_2$	--	--	–--	0.16	0.00	0.00	
	Frame$_3$	--	--	--	0.00	0.18	0.43	
d29	**Frame$_1$**	0.60	0.16	0.00	–--	0.22	0.66	1.55
d34	**Frame$_1$**	0.30	0.00	0.18	0.22	–--	0.35	1.05
d43	**Frame$_1$**	0.43	0.00	0.00	0.66	0.35	–----	1.44

3.3.5. Fase penyusunan Ringkasan

Sebuah kalimat representatif dipilih dari setiap *cluster* berdasarkan hasil dari proses pemilihan kalimat representatif. Pemilihan kalimat dimulai dari *cluster* yang memiliki bobot *cluster importance* paling tinggi. Kemudian pemilihan dilanjutkan pada *cluster* berikutnya sesuai dengan daftar urutan *cluster* berdasarkan bobot *cluster importance* secara *descending*.

3.3.6. Pembuatan Perangkat Lunak

Pada tahap ini ide dari hasil kajian pustaka dan usulan dari metode akan dituangkan kedalam aplikasi yang selanjutnya akan digunakan sebagai sarana uji

coba untuk membuktikan kemampuan metode yang diusulkan oleh penulis. Aplikasi yang dibangun berupa aplikasi desktop dengan bahasa pemrograman Java dan Mysql sebagai penyimpanan datanya.

3.4 Sekenario Uji coba

Uji coba sistem dilakukan untuk menguji atau menjalankan sistem dengan beberapa parameter yang ada pada metode. Pada tahap uji coba enam buah parameter *(Hr_{min}, ε, ST, θ, α dan T)* sistem terlebih dahulu diestimasi melalui proses estimasi parameter. Tujuan dari proses estimasi parameter parameter tersebut adalah mendapatkan nilai parameter-parameter yang paling optimal sehingga dapat memberikan hasil testing yang terbaik. Parameter-parameter yang terdapat pada sistem peringkasan ditunjukkan pada Tabel 3.3.

Tabel 3.3 Parameter Threshold yang Diestimasi

Notasi	Keterangan	Implementasi
HR_{min}	Batas nilai minimum dari *Histogram Ratio*	Fase Penbentukan Cluster SHC
ε	Batas selisih maksimum antara HR_{old} dengan HR_{new}	
S_T	Batas *similarity bin* pada perhitungan *histogram ratio*	
θ	Batas frekuensi minimal kata *w* dalam proses *cluster ordering*	Fase Pembobotan Cluser Important
T	Nilai *threshold* untuk menentukan kesamaan dari argumen	Fase Pemilihan kalimat representatif cluster
α	Nilai *threshold* untuk menentukan pembentukkan *edge* antar kalimat pada fitur semantic *sentence information density*	

Alur pengujian sistem mulai dari estimasi parameter hingga testing sistem. Sebelum melakukan uji coba, dataset DUC 2004 yang digunakan dipisahkan

terlebih dahulu ke dalam dua kategori yaitu data training dan data testing. Pada penelitian ini proporsi data training yang digunakan adalah 50% dan proporsi data testing juga 50%. Pada penelitian ini kinerja dari metode yang diajukan dievaluasi berdasarkan nilai ROUGE-N (Lin, 2004).

$Hr_{min,}$, ε, dan S_T adalah parameter optimasi yang digunakan pada proses *clustering SHC*, θ frekuensi minimal kata *w* dalam proses pengurutan *cluster*, *T* adalah *threshold* untuk menentukan kesamaan dari argumen pada proses penghitungan kesamaan kalimat, α threshold untuk menentukan pembentukkan edge antar kalimat. Pada proses training uji coba akan dilakukan pada data training untuk melakukan optimasi parameter seluruh parameter Tabel 3.3. Setelah didapat parameter optimal dari *(Hr_{min}, ε, S_T, θ, T dan α)* dilakukan uji coba terhadap data testing. Beberapa tahapan rencana uji coba adalah sebagai berikut:

3.4.1. Estimasi Parameter

Pada proses estimasi parameter bertujuan untuk mencari nilai optimal dari parameter HR_{min}, ε, *ST*, θ, *dan* α pada metode yang diusulkan. Pada proses estimasi parameter nilai parameter akan dilakukan inisialisasi dengan beberapa parameter. Parameter yang telah diinsialisasi dikombinasikan untuk mendapatkan kombinasi nilai parameter terbaik yang akan digunakan untuk proses testing. Nilai inisialisasi parameter HR_{min}, ε, *ST*, θ, *dan* α pada metode yang diusulkan juga akan digunakan pada pada metode lain yang akan dibandingkan.

3.4.2. Testing

Berdasarkan kombinasi nilai parameter HR_{min}, ε, *ST*, θ, *dan* α yang telah dioptimalkan pada proses estimasi parameter, maka pada proses testing kombinasi nilai parameter tersebut digunakan secara langsung untuk menguji data testing. ROUGE-*N* mengukur perbandingan *N-gram* dari dua ringkasan, dan menghitung berapa jumlah yang cocok.

Pada uji coba digunakan 1-gram dan 2-gram (ROUGE 1 dan ROUGE 2). Hasil rata-rata ROUGE 1 dan ROUGE 2 dari hasil testing metode yang diusulkan akan dibandingkan dengan pengukuran rata-rata ROUGE-1 dan ROUGE-2

terhadap metode peringkasan multi dokumen yang dikembangkan oleh (Suputra dkk 2013) SIDeKiCK (*Sentence Information Density Kata Kunci Cluster Kalimat*), metode *Local Importance Global Importance* (LIGI) (Sarkar, 2009),

3.5 Evaluasi Hasil

Kualitas hasil ringkasan pada penelitian ini dievaluasi dengan ROUGE. Metode ROUGE telah diadopsi dari DUC untuk mengevaluasi peringkasan teks otomatis. ROUGE sangat efektif digunakan untuk mengevaluasi peringkasan dokumen (Lin, 2004). ROUGE mengukur kualitas hasil ringkasan dengan menghitung unit-unit yang *overlap* seperti *N-gram*, urutan kata dan pasangan-pasangan kata antara ringkasan kandidat dan ringkasan sebagai referensi. ROUGE-*N* mengukur perbandingan *N-gram* dari ringkasan, dan menghitung berapa jumlah yang cocok. Perhitungan ROUGE-*N* yang diadopsi dari perhitungan Lin (2004) ditunjukkan pada Persamaan (3.2):

$$\text{ROUGE} - N = \frac{\sum_{S \in \text{Summ}_{\text{ref}}} \sum_{N-\text{gram} \in S} \text{Count}_{\text{match}}(N-\text{gram})}{\sum_{S \in \text{Summ}_{\text{ref}}} \sum_{N-\text{gram} \in S} \text{Count}(N-\text{gram})} \qquad (3.2)$$

Dimana *N* menunjukkan panjang dari *N-gram*, $Count_{match}$*(N-gram)* adalah jumlah maksimum dari *N-gram* yang muncul pada ringkasan kandidat dan ringkasan sebagai referensi. *Count(N-gram)* adalah jumlah dari *N-gram* pada ringkasan sebagai referensi. Pada penelitian ini fungsi ROUGE yang digunakan adalah ROUGE dengan nilai *N-gram* =1 dan *N-gram* =2. *N-gram* =1 dipilih karena ROUGE-1 lebih berkorelasi dengan ringkasan secara manual (Lin dan Hovy, 2003) sedangkan *N-gram* =2 digunakan sebagai pembanding *N-gram* =1. Hasil dari perhitungan ROUGE mempunyai nilai kesamaan 0 sampai 1.

Perbandingan kualitas hasil ringkasan berdasarkan nilai ROUGE-*N* dilakukan untuk mengetahui apakah metode yang diajukan dapat berjalan efektif atau tidak. Perbandingan dilakukan terhadap metode-metode yang ada pada fase pemilihan kalimat representatif dengan *framework* peringkasan multi-dokumen yang sama seperti pada Gambar 3.1. Metode-metode yang dimaksud adalah pemilihan kalimat representatif berdasarkan pendekatan *local importance* dan

global importance, *sentence information density* dan metode yang diajukan. Ketiga metode tersebut dievaluasi berdasarkan nilai ROUGE-*N* yang dihasilkan. Nilai ROUGE-*N* yang terbesar dari setiap metode menunjukkan kualitas hasil ringkasan yang terbaik.

BAB 4
IMPLEMENTASI DAN UJI COBA

Pada sub bab ini di dijelaskan implementasi dari metode yang diusulkan berdasarkan desain model yang digunakan. Pada implementasi bahasa pemrograman yang digunakan adalah Java dan DUC 2004 Task 2 sebagai dataset yang akan diolah. Pada sub bab ini juga akan dipaparkan uraian hasil dari uji coba yang telah dilakukan.

4.1 Implementasi Metode

Pada implementasi metode usulan desain model sistem yang digunakan diadopsi dari (Sarkar, 2009) Gambar 3.1. Pada sub bab ini model desain sistem akan diimplementasikan kedalam bentuk program baik metode yang diusulkan atau metode yang akan dijadikan tolak ukur. Setiap fase model akan dipaparkan pada sub-bab selanjutnya.

4.1.1. Implementasi Teks Preprosessing

Dalam teks preprosessing terdapat beberapa tahapan utama diantaranya adalah *xmlparsing, segmentation, stopword removal, stemming dan Semantic Role Labeling*. Beberapa tahapan ini dibutuhkan karena dataset yang digunakan adalah berupa *file xml*. Dari *file xml* dilakukan preprosessing untuk mendapatkan data berita yang terdapat dalam tiap file dan kemudian akan disimpan didalam file objek yang bisa digunakan pada penelitian ini.

Data awal dataset DUC 2004 task 2 berupa file dengan format *xml* dengan jumlah 500 file berita dan telah dikelompokkan berdasarkan folder (*Cluster*). Tiap file berisi berita berformat *xml* yang dimulai dengan *header* <*DOC*>, Kemudian dilanjutkan dengan <*DOCNO*> xxxxxxx <*/DOCNO*> yang merupakan keterangan nomor dari dokumen, <*DOCTYPE*> <*/DOCTYPE*> tipe dari jenis dokumen pada dataset ini menggunakan tipe berita (*news*), <*TXTTYPE*><*/TXTTYPE*> merupakan keterangan tipe dari teks yang terdapat dalam file dan merupakan penanda isi berita pada file tersebut.

```
<DOC>
<DOCNO> .......... </DOCNO>
<DOCTYPE> NEWS </DOCTYPE>
<TXTTYPE> NEWSWIRE </TXTTYPE>
<TEXT>
<P>
.......................
</P>
</TEXT>
</DOC>
```

Gambar 4.1 Format dataset DUC 2004 Task 2

Pada preprosessing awal, dataset yang berupa file xml dibaca dan dilakukan *xmlparsing* untuk mendapatkan data informasi dari nomor dokumen dan berita yang terdapat dalam file dataset. Pada *xmlparsing* data yang diambil adalah lokasi folder dari file yang digunakan sebagai nama dari kumpulan berita, *<DOCNO></DOCNO>* nomor dokumen dan *<TEXT> <P></P></TEXT>* yang merupakan berita yang terdapat pada file tersebut.

Tabel 4.1 Parsing XML pada Dataset

Metadata	**Keterangan**
<DOC></DOC>	Awal dan akhir dari file berita
<DOCNO> </DOCNO>	Nomor dokumen
<DOCTYPE> </DOCTYPE>	Tipe dari dokumen
<TXTTYPE></TXTTYPE>	Tipe dari teks
<TEXT> <P></P></TEXT>	Isi dari dokumen

Data hasil *xmlparsing* diproses untuk mendapatkan kalimat-kalimat penyusun dari konten berita dengan memggunakan *library Stanford Natural Language Processing* (Manning dkk, 2014). Kalimat hasil segmentasi disimpan dalam *array* bertipe *string* yang kemudian dilakukan segmentasi lagi untuk mendapatkan kata penyusun dari kalimat tersebut.

```
            Logger.getLogger(SentenceSegmentation.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    public void parsingKalimat(String dokString) {
        corpus = new corpus(dokString);
        this.dokString = dokString;
        Annotation document = new Annotation(dokString);
        Properties props = new Properties();
        props.put("annotators", "tokenize, ssplit, pos");
        StanfordCoreNLP pipeline = new StanfordCoreNLP(props);
        pipeline.annotate(document);
        List<CoreMap> sentences = document.get(CoreAnnotations.SentencesAnnotation.class);
        int i = 0;
        arrKalimat = new String[sentences.size()];
        arrstemKalimat = new String[sentences.size()];
        arrstpKalimat = new String[sentences.size()];
        for (CoreMap sentence : sentences) {
            //mendapatkan segmentasi kalimat
            arrKalimat[i] = sentence.toString().replaceAll("\\n", "");
            kalimat kalimat = new kalimat(arrKalimat[i]);
            try {
                ArrayList<FrameSRL> parse = srl.parse(arrKalimat[i]);
                kalimat.setFrameSRL(parse);
            } catch (Exception ex) {
                Logger.getLogger(SentenceSegmentation.class.getName()).log(Level.SEVERE, null, ex);
            }
            corpus.addKalimat(kalimat);
            String stpKalimat = new String();
            String stemKalimat = new String();
            for (CoreLabel token : sentence.get(CoreAnnotations.TokensAnnotation.class)) {
```

Gambar 4.2 Preprosessing Kalimat

Pada proses segmentasi kalimat ini dilakukan juga ektraksi kalimat berdasarkan *Semantic Rule Labeling* (SRL) menggunakan *library mate tool* (Anders dkk, 2010). Pada proses SRL terdapa beberapa tahapan diantaranya *token*, *lemmatizer*, *postagger dan srlparser*. Dari hasil *parser* kemudian dimasukkan kedalam *array*.

```
public Preprocessor getPreprocessor(FullPipelineOptions options) throws IOException {
        Tokenizer tokenizer=(options.loadPreprocessorWithTokenizer ? getTokenizer(options.tokenizer): null);
        Lemmatizer lemmatizer=getLemmatizer(options.lemmatizer);
        Tagger tagger=options.tagger==null?null:BohnetHelper.getTagger(options.tagger);
        is2.mtag.Tagger mtagger=options.morph==null?null:BohnetHelper.getMTagger(options.morph);
        Parser parser=options.parser==null?null:BohnetHelper.getParser(options.parser);
        Preprocessor pp=new Preprocessor(tokenizer, lemmatizer, tagger, mtagger, parser);
        return pp;
}

public abstract String verifyLanguageSpecificModelFiles(FullPipelineOptions options);

Tokenizer getDefaultTokenizer(){
        return new WhiteSpaceTokenizer();
}
public Tokenizer getTokenizer(File tokenModelFile) throws IOException{
        if(tokenModelFile==null)
                return getDefaultTokenizer();
        else
                return getTokenizerFromModelFile(tokenModelFile);
}

Tokenizer getTokenizerFromModelFile(File tokenModelFile) throws IOException {
        return OpenNLPToolsTokenizerWrapper.loadOpenNLPTokenizer(tokenModelFile);
}
```

Gambar 4.3 Extraksi Peran Semantik pada Preprosessing

Setelah proses segmentasi dilakukan proses *stopward removal* untuk menghilangkan kata yang kurang penting. Selain itu dilakukan pemeriksaan untuk

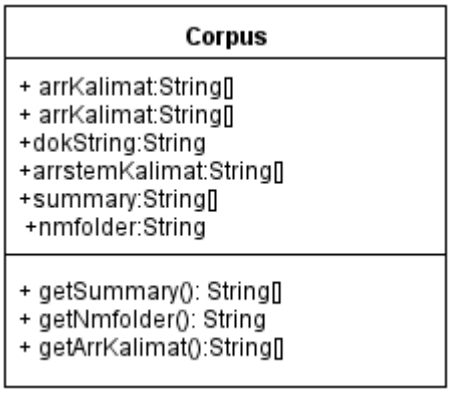
menghilangkan tanda baca. Aturan yang diterapkan dalam *stopword removal* ini diantaranya adalah menghilangkan seluruh kata yang terdapat dalam *wordlist* dan memeriksa kelayakan dari kata tersebut.

```
// fungsi untuk stemming
public String Stemming(String word) throws ClassNotFoundException, InstantiationException, IllegalAccessException {
    Class stemClass = Class.forName("ssid.stemming.englishStemmer");
    SnowballStemmer stemmer = (SnowballStemmer) stemClass.newInstance();
    stemmer.setCurrent(word);
    stemmer.stem();
    String stemmedWord = stemmer.getCurrent();
    return stemmedWord;
}
```

Gambar 4.4 Fungsi Stemming dengan SnowballStemmer

Fungsi *boolean isWord()* merupakan suatu fungsi yang bertugas untuk dalam memeriksa apakah kata yang dihasilkan oleh *token* adalah kata yang baku beberapa filter yang digunakan dalam funsi ini adalah kata yang dihasilkan tidak mengandung koma (,) atau dash (-), dan kata bukan merupakan singkatan selain itu juga seluruh tanda baca juga dihilangkan.

Langkah selanjutnya adalah dialkukan stemming terhadap kata yang telah lolos seleksi dengan menggunakan *snowball stemmer*. Listing program stemmer ditunjukkan Gambar 4.4.



Gambar 4.5 UML Class Objek Corpus dari Hasil Preprosessing

Data hasil preprosessing yang dilakukan disimpan kedalam data file dengan format dts yang merupkan implementasi objek *class corpus*. File hasil preprosessing inilah yang nantinya akan diproses ke tahap selanjutnya.

Tabel 4.2 Preprosessing Kalimat

Kalimat Asli	Kalimat Setelah Preprosessing
In a statement reported by the Anatolia news agency, Ecevit said he would see President Suleyman Demirel Monday morning.	statement report Anatolia news agenc Ecevit Presid Suleyman Demirel Monday morn
Ecevut's alternate efforts to make a minority coalition with outside backing for his Democratic Left Party from Parliament also failed.	Ecevut altern effort make minor coalit back Democrat Left Parti Parliament fail
After failing to bring together political rivals in a coalition, Premier-designate Bulent Ecevit announced Saturday that he was returning his mandate to the Turkish president.	fail bring polit rival coalit Bulent Ecevit announc Saturday return mandat Turkish presid

4.1.2. Implementasi Clustering Kalimat

Data hasil preprosessing kemudian diolah dengan metode SHC. Kalimat diproses dengan metode SHC. Kalimat pertama secara otomatis akan membentuk *cluster* baru. Kalimat selanjutnya diuji terhadap setiap cluster yang telah terbentuk. Tiap kalimat akan dilakukan percobaan untuk ditambahkan kedalam setiap *cluster* yang telah ada.

Metod SHC *clustering* dimulai dengan mencipkan *class SHCclustering* dengan masukan variabel *int numBin, double threshold, double min_histogram, double diff_threshold, ArrayList<String> Listkalimat*.

Setiap cluster dilakukan simulasi perhitungan histogram pada cluster sebelum ditambahkan kalimat dan histogram setelah ditambahkan kalimat baru. Kemudian dilakukan seleksi *if()* jika nilai histogram yang baru lebih besar dari histogram lama dan histogram baru lebih besar dari *threshold* minimal histogram dan perbedaan dari histogram baru dan histogram lama lebih kecil dari batasan

differentThreshold maka kalimat tersebut bisa dimasukkan sebagai anggota *cluster*. Jika syarat tidak terpenuhi maka kalimat akan digunakan untuk membentuk *cluster* baru.

```
public void prosesSHC() {
    double oldClusterSHCHistogramBeforeAddingDoc = minHistogram;
    double newClusterSHCHistogramAfterAddingDoc = minHistogram;
    boolean wasAdded = false;
    Bin[] simulationPackage;/*
     * Bin[] tempSimulationPackage; double temp = 0;
     */

    for (int i = 0; i < arrKalimat.size(); i++) {
      wasAdded = false;
      for (int j = 0; j < clusters.size(); j++) {
        oldClusterSHCHistogramBeforeAddingDoc = (clusters.get(j))
                .getHistogramOfCluster();
        simulationPackage = clusters.get(j)
                .simulationForAddingNewStringToCluster(arrKalimat.get(i));

        /*
         * tempSimulationPackage = simulationPackage; for(int
         * k=0;k<tempSimulationPackage.length;k++) { temp =
         * simulationPackage
         * [k].getCountOfSimilarityinBin()+(clusters.get
         * (j)).getClusterSHCBins()[k].getCountOfSimilarityinBin();
         * tempSimulationPackage[k].setCountOfSimilarityinBin(temp); }
         */
        newClusterSHCHistogramAfterAddingDoc = (clusters.get(j))
                .simulationCalculateHistogramOfCluster(simulationPackage);
        if ((newClusterSHCHistogramAfterAddingDoc > oldClusterSHCHistogramBeforeAddingDoc)
                || ((newClusterSHCHistogramAfterAddingDoc > minHistogram) &&
                ((oldClusterSHCHistogramBeforeAddingDoc - newClusterSHCHistogramAfterAddingDoc)
                < differenceThreshold))) {
          wasAdded = true;
          clusters.get(j).addingNewStringToCluster(arrKalimat.get(i),i,
                   simulationPackage,
                   newClusterSHCHistogramAfterAddingDoc);
          break;

        }
      }
      if (wasAdded == false) {
        ClusterSHC newClusterSHC = new ClusterSHC();
        newClusterSHC.addingNewStringToCluster(arrKalimat.get(i),i, null, -1);
        clusters.add(newClusterSHC);
      }
    }
```

Gambar 4.6 Proses *Clustering* dengan SHC

```
for (Map.Entry<Integer, String> entrySet : arrKalimatofCluster.entrySet()) {
    Integer key = entrySet.getKey();
    String value = entrySet.getValue();
    similarityAddedToRange = false;
    //docSimilarity = arrKalimatofCluster.get(i).distance(doc);
    //System.out.println(doc.getText()+" :::: "+docsOfCluster.get(i).getText()+" ===========> "+docSimilarity);
    //menghitung sim string unigram
    stringSimilarity = UnigramMaching.unigramMatching(value, str);
    for (int j = 0; j < localHistogramBin.length
            && similarityAddedToRange == false; j++) {
      if ((stringSimilarity >= (localHistogramBin[j]
              .getLowerSimilarityOfBin()))
              && (stringSimilarity < (localHistogramBin[j]
              .getUpperSimilarityOfBin()))) {
        tempCountOfHistogram = localHistogramBin[j]
                .getCountOfSimilarityinBin();
        localHistogramBin[j].setCountOfSimilarityinBin(tempCountOfHistogram + 1);
        similarityAddedToRange = true;
      }
    }
  }
  return localHistogramBin;
```

Gambar 4.6 Simulasi Penambahan Kalimat Pada Cluster

Dalam simulasi perhitungan similaritas antar kalimat untuk menentukan *histogram* pada tiap *cluster* digunakan *unigram maching* dengan Persamaan 2.1. Pada proses ini gigunakan tiga parameter yang digunakan dalam pembentukan *cluster* kalimat diantanya HR_{min}, ε *dan* S_T.

Dari hasil *clustering* kalimat ini disimpan kedalam *array* dengan beranggotakan seluruh kalimat yang memenuhi kriteria HR_{min}, ε, *dan* S_T. Kumpulan cluster ini selanjutnya akan diurutkan pada fase selanjutnya berdasarkan *cluster impotant*.

4.1.3. Implementasi Pengurutan Cluster

Pada proses *clustering* kalimat diperoleh *array cluster*. Kemudian *array* tersebut dijadikan input kedalam *class clusterOrder* yang berfungsi menghitung bobot dari *cluster*. Tiap bobot dari *cluster* dihitung dengan menggunakan Persamaan 3.1 berdasarkan batasan frequensi dari jumlah kata. Bobot dari *cluster important* kemudian dimasukkan kembali kedalam objek *cluster* yang akan digunakan sebagai keterangan dari *cluster* tersebut.

```
1: N -> Cluster Array
2: for each cluster N do
3:     valueSentence -> number
4:     for each sentence cluster do
5:         for each word sentence do
6:             if word.getFrequency() > θ then
7:                 valueSentence+=Math.log10((1 + word.getFrequency()));
8:             end if
9:             end for
10:        end for
11:        cluster set weight valueSentence
12: end for
```

Gambar 4.8 Algoritma *Cluster Important*

Tiap *cluster* yang telah mempunyai bobot kemudian diurutkan berdasarkan bobot dengan menggunakan algoritma *quick sort* dengan cara membagi array kedalam partisi-partisi. Sehingga diperoleh *array* beranggotakan *cluster* yang telah terurut berdasarkan bobot *cluster important*.

Pada perhitungan *cluster important* digunakan parameter θ untuk menentukan batasan toleransi pada penentuan *cluster important*. Fungsi *get frequency()* pada Gambar 4.9 Merupakan jumlah suatu kata pada keseluruhan

cluster. Bobot hasil perhitungan *cluster important* kemudian dikembalikan sebagai keterangan pada *cluster* objek dengan fungsi *setClusterorder()*.

```
public void calculate() {
  ArrayList<ClusterSHC> arrayList = new ArrayList<ClusterSHC>();
  for (Iterator<ClusterSHC> iterator = clusterSHCs.iterator(); iterator.hasNext();) {
    ClusterSHC next = iterator.next();
    HashMap<Integer, String> stringsOfCluster = next.getStringsOfCluster();
    double value = 0;
    ArrayList<String> woList = new ArrayList<String>();
    for (String kalimat : stringsOfCluster.values()) {
      String[] kata = kalimat.split(" ");
      for (int i = 0; i < kata.length; i++) {
        if (mapWord.containsKey(kata[i])) {
          if (mapWord.get(kata[i]).getFrequency() >= ThrCountWord) {
            value += Math.log10((1 + mapWord.get(kata[i]).getFrequency()));
          }
          //System.out.println(kata[i]+' '+value);
          woList.add(kata[i]);
        }
      }
    }
    next.setClusterOrder(value);
    next.setWords(woList);
    arrayList.add(next);
  }
  clusterSHCs=arrayList;
}
```

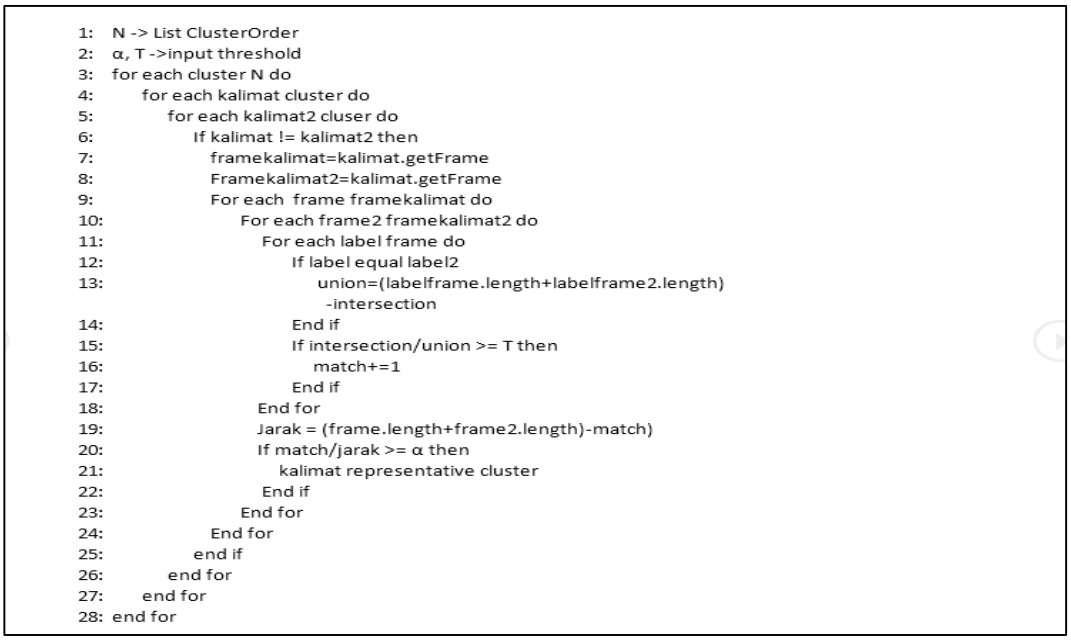
Gambar 4.9 Simulasi Penambahan Kalimat pada *Cluster*

```
public static int partition(ArrayList<ClusterSHC> vector, int left, int right) {
  while (true) {
    while ((left < right) && (vector.get(right).getClusterOrder() < vector.get(right).getClusterOrder())) {
      right--;
    }
    if (left < right) {
      swap(vector, left, right);
    } else {
      return left;
    }
    while ((left < right) && (vector.get(left).getClusterOrder() < vector.get(right).getClusterOrder())) {
      left++;
    }
    if (left < right) {
      swap(vector, left, right--);
    } else {
      return right;
    }
  }
}
```

Gambar 4.10 *Cluster Order* dengan algoritma *quick sort*

4.1.4. Implementasi Pemilihan Kalimat Representatif

Pada implementasi fase ini array *cluster* yang telah terurut kemudian dilakukan perhitungan untuk mendapatkan kandidat yang dapat mewakili tiap *cluster* tersebut. Pada fase ini beberapa parameter *threshold* digunakan diantaranya parameter α dan T.



```
1:  N -> List ClusterOrder
2:  α, T ->input threshold
3:  for each cluster N do
4:      for each kalimat cluster do
5:          for each kalimat2 cluser do
6:              If kalimat != kalimat2 then
7:                  framekalimat=kalimat.getFrame
8:                  Framekalimat2=kalimat.getFrame
9:                  For each  frame framekalimat do
10:                     For each frame2 framekalimat2 do
11:                         For each label frame do
12:                             If label equal label2
13:                                 union=(labelframe.length+labelframe2.length)
                                     -intersection
14:                             End if
15:                             If intersection/union >= T then
16:                                 match+=1
17:                             End if
18:                         End for
19:                         Jarak = (frame.length+frame2.length)-match)
20:                         If match/jarak >= α then
21:                             kalimat representative cluster
22:                         End if
23:                     End for
24:                 End for
25:             end if
26:         end for
27:     end for
28: end for
```

Gambar 4.11 Algorima Pemilihan Kalimat Representatif

Jarak antar kalimat dalam *cluster* dihitung dengan mempertimbangkan kesamaan label peran semantik dari setiap kata berdasarkan Tabel 2.2, selain itu juga memperhitungkan hasil dari *frameSRL*.

Fungsi *calculate* pada *class ssidku* berfungsi untuk mendapatkan perwakilan dari setiap *cluster* berdasarkan label dan *frame* dari kalimat. *Class ClusterSHC* yang merupakan objek dari cluster hasil bentukan fase *cluster ordering*. Variabel order yang merupakan *array* dari *cluster-cluster* yang telah terurut berdasarkan bobot masing-masing cluster.

Dilakukan perulangan *order* berdasarkan jumlah dari besaran *array order*. Selanjutnya satu persatu anggota *array order* diambil dengan perintah *order.get(i)*, *i* merupakan variabel index dari *cluster* yang akan diambil. Tiap cluster kemudian diambil kalimat beserta index dari kalimat tersebut dalam dokumen. Index ini digunakan untuk mengambil *Semantic Rule Labeling* dari hasil preprosessing awal.

Selanjutnya dilakukan perulangan untuk membandingkan *frame* semantik dan label dari tiap kalimat dalam *cluster*. Fungsi *simSentenceSSIDWithLabel(key, key2)* merupakan suatu fungsi yang bertugas untuk mencocokkan tiap *frame* dari kalimat berdasarkan label yang telah diperoleh.

```
public void calculate() {
    // dibalik
    int jumlah_kata = 0;
    for (int i = order.size() - 1; i > 0; i--) {
        //System.out.println("===========CLUSTER KE " + i + " ==================");
        ClusterSHC cluster = order.get(i);
        HashMap<Integer, String> stringsOfCluster = cluster.getStringsOfCluster();
        String tempRingkasan = null;
        double tempMax = 0.0;
        // perbandingan looping kalimat dlama cluster
        for (Map.Entry<Integer, String> entrySet : stringsOfCluster.entrySet()) {
            Integer key = entrySet.getKey();
            String primary = entrySet.getValue();
            //System.out.println("===========JARAK ANTAR KALIMAT==================");
            double unigramMatching = 0.0;
            for (Map.Entry<Integer, String> entrySet2 : stringsOfCluster.entrySet()) {
                Integer key2 = entrySet2.getKey();
                String pembanding = entrySet2.getValue();
                if (!key.equals(key2)) {
                    //double simSentenceSSID = simSentenceSSID(key, key2);
                    double simSentenceSSID = simSentenceSSIDWithLabel(key, key2);
                    //System.out.println(primary + " <==> " + pembanding + " = " + simSentenceSSID);
                    //System.err.println(kalimats.get(key).getKalimat()+" <=======> "+kalimats.get(key2).getKalimat());
                    unigramMatching = simSentenceSSID;
                }
            }

            if (unigramMatching >= tempMax) {
```

Gambar 4.12 Pemilihan Kalimat Representatif

```
ArrayList<kalimat> kalimats = corpus.getKalimat();
ArrayList<FrameSRL> get = kalimats.get(kalimat1).getFrameSRL();
ArrayList<FrameSRL> get1 = kalimats.get(kalimat2).getFrameSRL();
//System.err.println(kalimats.get(kalimat1).getKalimat()+" <=======> "+kalimats.get(kalimat2).getKalimat());
for (Iterator<FrameSRL> iterator = get.iterator(); iterator.hasNext();) {
    FrameSRL next = iterator.next();
    for (Iterator<FrameSRL> iterator1 = get1.iterator(); iterator1.hasNext();) {
        FrameSRL next1 = iterator1.next();
        //System.out.println(framekalimat+"<====>"+framekalimat1);
        //double unigramMatching = labeMach(next, next1);
        double Jcart = labeMach2(next, next1);
        //System.out.println(unigramMatching+" "+this.threshold);
        if (Jcart >= threshold) {
            sim += Jcart;
        }
    }
}
```

Gambar 4.13 Format dataset DUC 2004 Task 2

Pada *labeMach2(next, next1)* digunakan untuk menghitung kesamaan tiap label dengan Persamaan *Jaccard Coefficient* (2.5). Pada perhitungan ini *threshold* digunakan untuk menentukan kesamaan dari kata yang terdapat pada tiap-tiap label. Jika jumlah kata yang sama dibagi dari seluruh gabungan kata antar label lebih besar atau sama dengan *threshold* maka label tersebut dianggap memiliki kesamaan.

Pada fase ini perhitungan *Jaccard Coefficient* dilakukan dua tahap yang pertama sebagai penentu kesamaan atar label dan yang kedua digunakan sebagi bobot kesamaan antar kalimat.

```
public static boolean computeJcart(String a, String b,double t) {
    double intersection = 0;
    String[] arrSplit = a.split(" ");
    String[] arrSplit1 = b.split(" ");
    // Match Count
    for (int i = 0; i < arrSplit.length; i++) {
        for (int j = 0; j < arrSplit1.length; j++) {
            if (arrSplit[i].toLowerCase().equals(arrSplit1[j].toLowerCase())) {
                intersection++;
                break;
            }
        }
    }
    double union=(arrSplit.length+arrSplit1.length)-intersection;
    //return intersection / union;
    if(intersection / union>=t){
        return true;
    }
    return false;
```

Gambar 4.14 Perhitungan Similaritas Peran Semantik

Kemudian didapat bobot kesamaan atar kalimat berdasarkan label peran kata dan frame dari kalimat yang dibandingkan. Kemudian nilai kesamaan ini dikembalikan pada perhitungan similaritas antar kalimat. Kalimat-kalimat tersebut kemudian dijadikan referensi untuk menentukan kalimat yang paling sesuai untuk mewakili *cluster*.

4.1.5. Implementasi Penyusunan Ringkasan

Pada fase ini hasil ringkasan dari *top cluster* disimpan kedalam variabel dengan tipe *string*. Pada penyusunan ringkasan kalimat dibatasi kurang lebih 100 kata. Pada fase ini pemotongan kalimat didasarkan pada satuan kalimat pembentuk. Jika pada saat ditambahkan kalimat akhir jumlah kata kurang dari 100 kata maka kalimat tersebut ditambahkan. Dalam penambahan tidak dilakukan pemotongan

kalimat. Kalimat terakhir ditambahkan secara keseluruhan sehingga hasil ringkasan bisa lebih besar dari 100 kata.

Gambar 4.16 Penyusunan Ringkasan

```
//batasi jumlah +- 100 kata
if (jumlah_kata < 100) {
    ringkasan.add(tempRingkasan);
    if (summary.isEmpty()) {
        summary = tempRingkasan;
    } else {
        summary += tempRingkasan;
    }
    jumlah_kata += tempRingkasan.split(" ").length;
} else {
    break;
}
```

4.2 Uji Coba

Pada bab ini dipaparkan hasil uji coba metode yang diusulkan untuk mengetahui performa dari metode yang diusulkan dan akan dibandingkan dengan dua metode sebelumnya yaitu LIGI dan SIDeKiCK.

Pengukuran performa akan dilakukan dengan membandingkan nilai optimal dilihat dari nilai ROUGE-1 dan ROUGE-2. Nilai ROUGE-1 dan ROUGE-2 yang lebih besar menunjukkan performa metode yang lebih baik dari segi korelasi ringkasan. Hasil ringkasan otomatis dari metode-metode yang diujicoba akan dibandingkan dengan hasil peringkasan manual pada dataset DUC 2004 task 2 sehingga diketahui performa dari tiap-metode. Pada proses ujicoba dataset DUC 2004 Task 2 dibagi menjadi 2 kelompok kelompok pertama digunakan sebagai data *training* sedangkan kelompok kedua digunakan sebagai data *testing*.

Tabel 4.3 Pembagian Dataset DUC 2004 Task 2

Data	Jumlah Cluster	Jumlah Dokumen dalam cluster	Total Dokumen
Trainig	25	10	250
Testing	25	10	250

Sekenario yang digunakan pada proses uji coba ini adalah traning dan testing. Proses training dilakukan untuk menentukan kombinasi parameter yang optimal selanjutnya pada proses testing kombinasi parameter ini akan diuji coba kedalam data *testing* dan akan dilakukan analisa terhadap hasil berdasarkan metode ROUGE-1 dan ROUGE-2. Semakin tinggi nilai ROUGE yang dihasilkan berarti sebakin tinngi korelasi antara hasil peringkasan otomatis dan data peringkasan manual.

4.2.1. Proses Estimasi Parameter

Proses estimasi parameter digunakan untuk menentukan parameter optimal pada sistem peringkasan multi dokumen. Proses estimasi parameter dilakukan untuk mengestimasi kombinasi nilai optimal dari enam buah parameter dalam sistem (HR_{min}, ε, S_T, θ, α dan T) yang telah didefinisikan sebelumnya pada Tabel 3.2.

Pada proses awal nilai-nilai paramater diinisialisasi berdasarkan kemungkinan nilai yang memungkinkan pada tiap parameter. Kemudian dilakukan proses estimasi parameter terhadap kumpulan data training untuk mencari kombinasi parameter yang paling optimal.

Tabel 4.4 Inisialisasi Nilai Parameter yang Digunakan dalam Estimasi parameter

Notasi	Inisialisasi
HR_{min}	0.4,0.5,0.6
ε	0.6,0.5,0.4
S_T	0.3,0.4,0.5,0.6
θ	10,12,15,17,20,22,25,27,30
α	0.4,0.5,0.6,0.7,0.8
T	0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9

Pada inisialisasi ini parameter HR_{min}, ε, S_T merupakan parameter yang berpengaruh terhadap pembentukan cluster sehingga parameter HR_{min}, ε, merupakan parameter yang nilainya selalu berdampingan [HR_{min}, ε]=[0.4,0.6], [0.5,0.5], [0.6,04]. Sehingga dilakukan inisialisasi secara berdampingan, sedangkan S_T merupakan parameter jarak antara *cluster* kemungkinan nilai parameter ini adalah 0-1.

Parameter θ merupakan parameter threshold terhadap jumlah suatu kata yang terdapat dalam dokument Nilai θ dipilih dari 10-30 melihat dari jumlah kata dalam document sehingga parameter tersebut dianggap paling sesuai melihat data dari dataset.

Parameter α merupakan jarak toleransi antar *node* atau kalimat. Rentangan nilai pada parameter ini adalah 0-1. Pada inisialisasi diberikan nilai 0.4-0.8 dengan menaikkan nila 0.1 pada setiap nilainya sehinngga terdapat 7 kemungkinan pada parameter α sendiri. Pada metode LIGI tidak memperhitungkan jarak antar kalimat tetapi lebih kepada kata-kata penting berdasarkan *Local Important* dan *Global Important* sehingga pada LIGI tidak menggunakan parameter α.

Tabel 4.5 Kombinasi Nilai Parameter yang Optimal Berdasarkan Nilai ROUGE-1

Metode	Nilai Parameter						Nilai Rouge-1
	S_T	HR_{min}	ε	θ	α	T	
LIGI	0.4	0.5	0.5	20	-	0.5	0.31113
SIDKCK	0.4	0.4	0.6	10	0.5	0.4	0.31886
SSID	0.3	0.6	0.4	15	0.8	0.3	0.32511

Tabel 4.6 Kombinasi Nilai Parameter yang Optimal Berdasarkan Nilai ROUGE-2

Metode	Nilai Parameter						Nilai Rouge-2
	S_T	HR_{min}	ε	θ	α	T	
LIGI	0.5	0.6	0.4	10	-	0.7	0.11103
SIDKCK	0.4	0.4	0.6	10	0.5	0.6	0.11630
SSID	0.4	0.5	0.5	30	0.7	0.8	0.11600

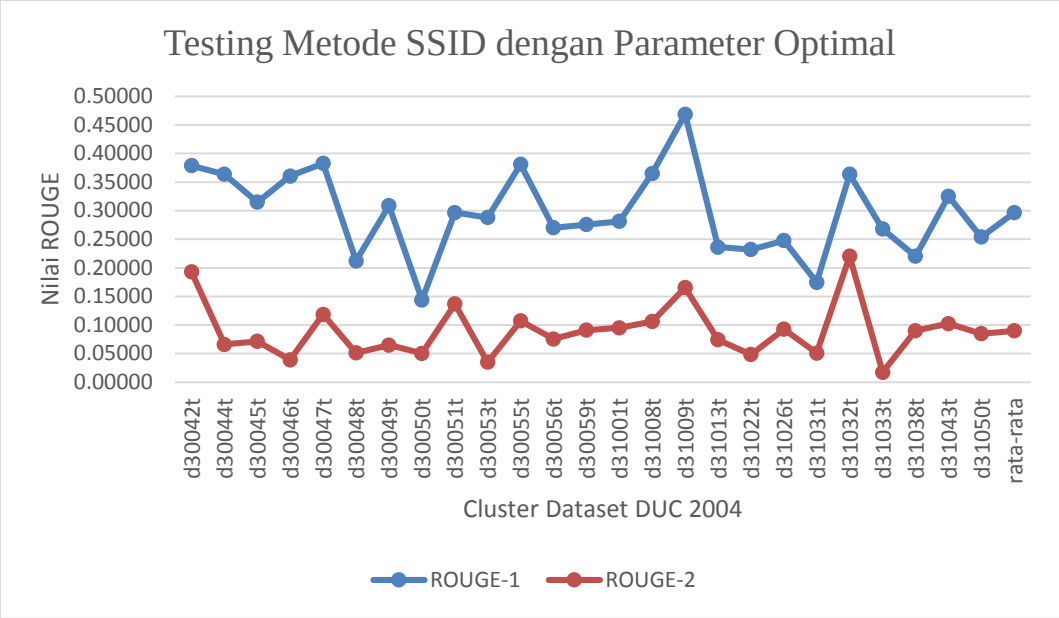
Parameter T digunakan untuk menentukan *threshold* ntuk menentukan kesamaan argumen dalam frame sematic rule labeling. Sedangkan pada LIGI patameter *T* akan digunakan sebagai bobot *Local Important* dan *Global Important* dengan batasan LI+GI=1 sehingga nilai parameter LI dan GI selalu berpasangan. Sedangkan pada SIDeKiCK T digunakan sebagai bobot dari kata kunci cluster itu sendiri. Rentangan nilai T adalah 0-1 dengan interval 0.1.

Hasil Proses traning menunjukkan bahwa kombinasi nilai parameter optimal untuk SSID [S_T=0.3, HR_{min}=0.6, ε=0.4, θ=15, α=0.8, T=0.3] untuk ROUGE-1 dengan nilai rata-rata 0.32.511 dan [S_T=0.4, HR_{min}=0.5, ε=0.5, θ=30, α=0.7, T=0.8] untuk ROUGE-2 dengan rata-rata 0.11600. Dari hasil estimasi parameter juga menunjukkkan bahwa penggunaan peran sematik pada pemilihan kalimat representative dalam SSID dapat meningkatkan nilai rata-rata ROUGE-1. Langkah selanjutnya adalah proses testing metode dengan parameter optimal dari hasil proses estimasi parameter.

4.2.2. Proses Testing Metode yang diusulkan

Pada sub-bab ini dipaparkan hasil testing data terhadap nilai ROUGE-1 dan ROUGE-2. Testing dilakukan dengam menggunakan nilai parameter yang optimal untuk metode yang diusulkan. Dilakukan proses testing data testing yang telah sisipakan pada proses ini digunakan parameter optimal dari proses estimasi parameter yaitu [S_T=0.3, HR_{min}=0.6, ε=0.4, θ=15, α=0.8, T=0.3] dan [S_T=0.4, HR_{min}=0.5, ε=0.5, θ=30, α=0.7, T=0.8] untuk ROUGE-2 sehingga didapat nilai rata-rata ROUGE-1 dan ROUGE-2 terhadap data testing seperti pada Gambar 4.18.

Gambar 4.18 dapat diamati terdapat beberapa data yang berada diatas rata-rata ROUGE baik ROUGE-1 ataupun ROUGE 2 diantaranya d30055t, d30047t, d31008t,d31033t. Hasil ROUGE-1 dan ROUGE-2 mengalami penurunan dari metode lain diantaranya pada dataset d31038t, d30045t, d31022t dan d30050t.



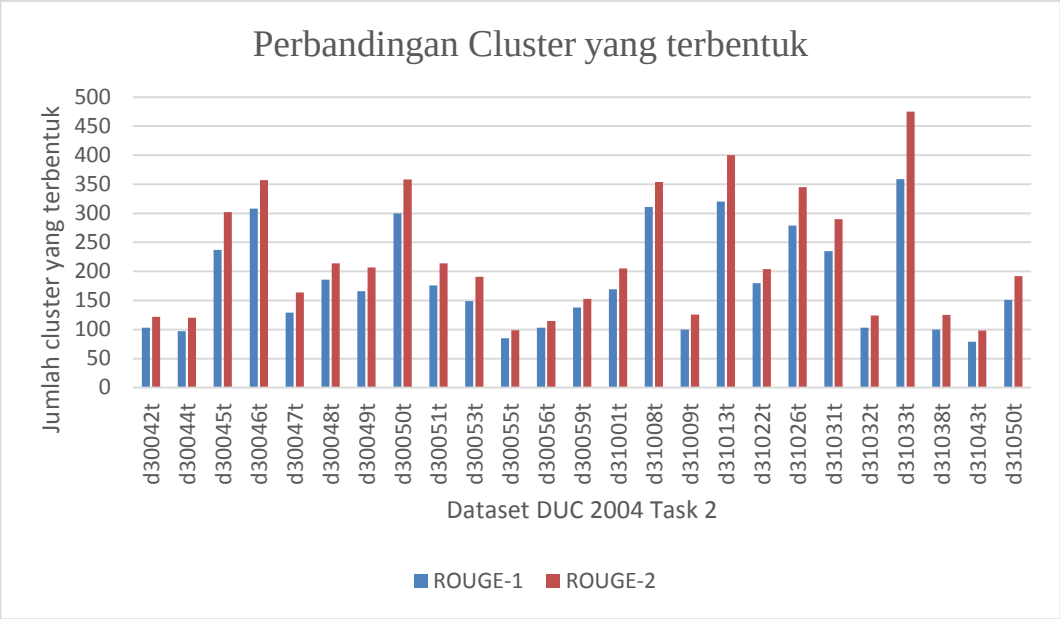
Gambar 4.17 Grafik Nilai ROUGE-1 dan ROUGE-2 Hasil Testing

Beberapa dataset mengalami penurunan dari metode lain diantaranya d31038t, d30045t, d31022t dan d30050t. Berdasarkan analisa dari hasil proses peringkasan otomatis dan cluster yang terbentuk dari proses testing. Proses SSID ini menjadi tidak efektif jika dataset mengandung banyak kalimat yang sama. Analisa ini dicantumkan pada lampiran.

Tabel 4.7 Hasil Proses Parameter Optimal pada Data Training

Metode Evaluasi	Parameter	Nilai rata-rata
ROUGE-1	[S_T=0.3, HR_{min}=0.6, ε=0.4, θ=15, α=0.8, T=0.3]	0.29656
ROUGE-2	[S_T=0.4, HR_{min}=0.5, ε=0.5, θ=30, α=0.7, T=0.8]	0.08992

Dari proses testing didapatkan nilai rata-rata ROUGE-1 = 0.29656 dan ROUGE-2 = 0.08992. Nilai rata-rata ROUGE-1 mempunyai nilai yang lebih besar dari ROUGE-2



Gambar 4.16 Perbandingan Cluster yang Terbentuk Pada Parameter Optimal untuk ROUGE-1 dan ROUGE-2

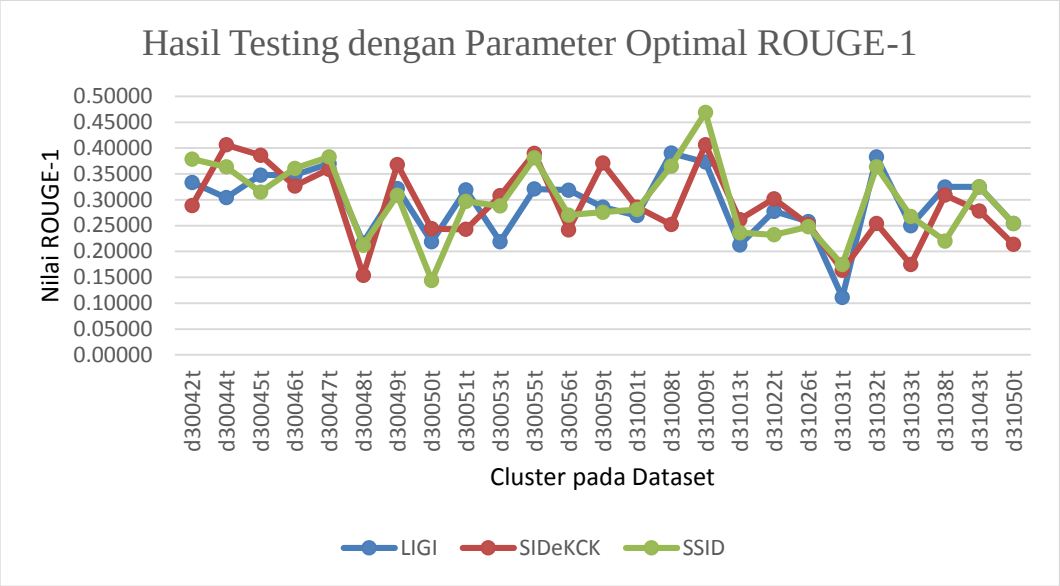
Gambar 4.16 Menunjukkan jumlah cluster yang terbentuk pada metode SSID dimana menunjukkan bahwa ROUGE-2 mempunyai rata-rata cluster yang lebih kecil dibandingkan dengan ROUGE-1 menunjukan bahwa ROUGE-2 menghasilkan jumlah cluster yang lebih banyak dalam prosesnya. Rata-rata jumlah cluster yang terbentuk adalah 182.52 untuk ROUGE-1 dan 222.16 untuk ROUGE-2.

Tabel 4.8 Rata-rata jumlah cluster

Metode Evaluasi	Rata-rata jumlah cluster
ROUGE-1	182.52
ROUGE-2	222.16

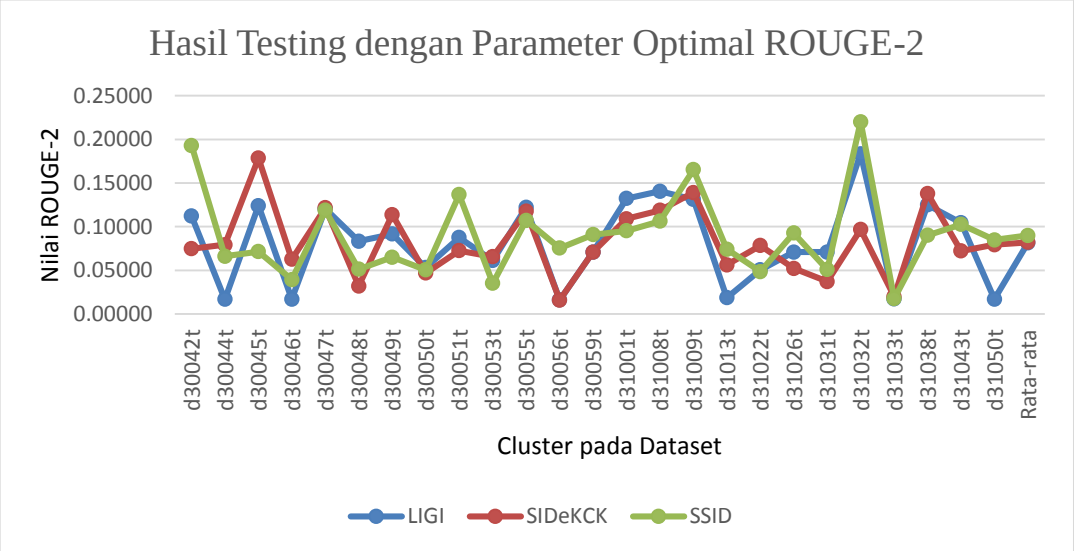
4.2.3. Perbandingan metode SSID, LIGI dan SIDeKiCK

Dari proses testing selanjutnya data hasil testing dari metode SSID akan dibandingkan dengan metode LIGI dan SIDeKiCK.



Gambar 4.18 Hasil Testing LIGI, SIDeKiCK dan SSID ROUGE-1

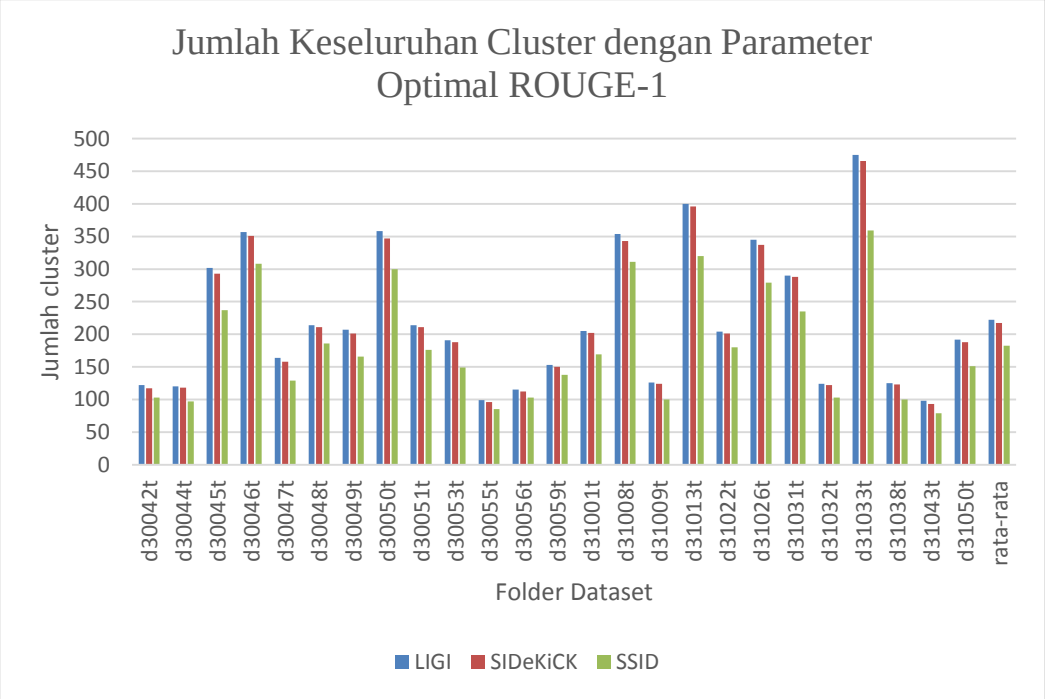
Parameter yang digunakan untuk LIGI dan SIDeKiCK adalah parameter optimal yang diperoleh pada proses estimasi parameter pada metode LIGI dan SIDeKiCK. LIGI [S_T=0.4, HR_{min}=0.5, ε=0.5, θ=20, T=0.5] untuk ROUGE-1 dan [S_T=0.5, HR_{min}=0.6, ε=0.4, θ=10, T=0.7] untuk ROUGE-2. SIDeKiCK [S_T=0.4, HR_{min}=0.4, ε=0.6, θ=10, α=0.5, T=0.4] untuk ROUGE-1 dan [S_T=0.4, HR_{min}=0.4, ε=0.6, θ=10, α=0.5, T=0.5] untuk ROUGE-2.



Gambar 4.19 Hasil Testing LIGI, SIDeKiCK dan SSID ROUGE-2

Hasil testing data dengan kombinasi parameter kombinasi parameter optimal dari setiap metode menghasilkan rata-rata nilai Rouge-1 SSID=0.29656, LIGI =0.294051 dan SIDeKick =0.289541.

Dari hasil uji testing tersebut dapat dilihat bahwa terdapat perbedaan pada distribusi nilai ROUGE-1 dan distribusi nilai ROUGE-2. Rata-rata Nilai ROUGE-1 dan ROUGE-2 yang dihasilkan oleh metode SSID lebih besar dari metode LIGI dan SIDeKCK. Sehingga dapat disimpulkan bahwa untuk kasus Perolehan nilai ROUGE-1 dan nilai ROUGE-2 metode SSID lebih baik dari LIGI dan SIDeKiCK. Langkah selanjutnya adalah melakukan analisa jumlah *cluster* yang dibutuhkan dalam proses peringkasan data jumlah cluster yang terbentuk terdapat pada Gambar 4.20.



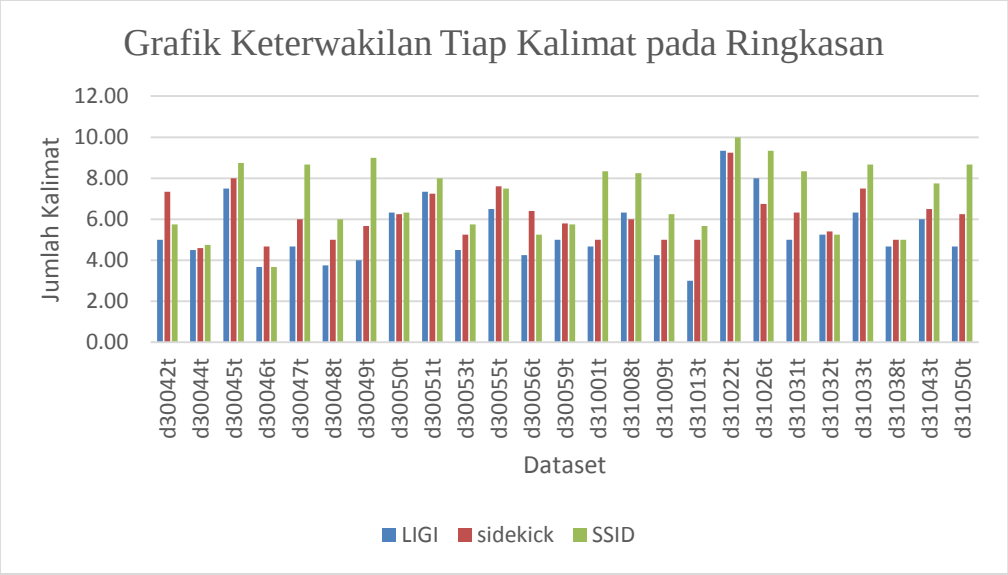
Gambar 4.20 Jumlah Cluster yang terbentuk

Gambar 4.20 menunjukkan bahwa *cluster* yang terbentuk pada proses SSID cenderung lebih sedikit dibandingkan dengan metode lain sehingga dapat disimpulkan *cluster* pada SSID mempunyai anggota yang kalimat lebih banyak. Rata-rata jumlah *cluster* yang terbentuk pada proses *testing* adalah SSID = 182.52, SIDeKick = 217.44 dan LIGI =222.16

Tabel 4.9 Rata-Rata Jumlah *Cluster* yang Dibutuhkan dalam Pembentukan Ringkasan

Metode	Rata-rata jumlah cluster
LIGI	3.36
SIDeKiCK	3.88
SSID	3.48

Tabel 4.9 diperoleh dengan melakukan analisa terhadap proses testing pada SSID atau metode lain dengan memperhatikan jumlah cluster yang diproses untuk menyusun ringkasan dimana hasil dari ringkasan hanya dibatasi sekitar 100 kata, Tabel analisa dapat dilihat pada data lampiran.



Gambar 4.21 Hasil Analisa Keterwakilan Kalimat pada Tiap metode

Jumlah *cluster* yang digunakan dalam sistem peringkasan cenderung sama sekitar 3 sampai 4 *cluster* yang digunakan sebagi pembentuk ringkasan ditunjukkan dalam analisa Tabel 4.8. Pada Gambar 4.20 ditunjukkan bahwa SSID membentuk jumlah cluster yang lebih sedikit dari metode lain dan menggunakan jumlah cluster yang sama dengan metode lain. Tahap selanjutnya adalah mencoba melakukan analisa terhadap jumlah kalimat yang diwakili oleh ringkasan dengan cara menghitung jumlah kalimat yang terdapat *cluster-cluster* yang digunakan dan membaginya dengan jumlah *cluster* dikarenakan setiap *cluster* hanya mempunyai satu kalimat representative yang digunakan sebagai penyusun ringkasan.

Pada Gambar 4.21 menunjukkan bahwa setiap kalimat pada peringksan SSID mewakili lebih banyak kalimat dari metode yang lain.

4.3 Analisa dan Pembahasan

Pada sub-bab ini dipaparkan mengenai hal-hal yang dapat dianalisis dari hasil testing metode SSID berdasarkan nilai ROUGE-1 dan ROUGE-2. Hal-hal yang dinalisis pada penelitian ini dipaparkan secara rinci pada sub sub-bab 4.3.1.

4.3.1. Analisa Performa Metode yang Diusulkan

Metode SSID berhasil mengungguli performa metode LIGI dan SIDeKiCK Berdasarkan nilai ROUGE-1 dan ROUGE-2. Nilai ROUGE-1 dan ROUGE-2 yang lebih besar membuktikan bahwa hasil ringkasan yang dihasilkan oleh metode SSID lebih berkorelasi terhadap hasil ringkasan yang dibuat secara manual oleh manusia. Metode SSID sendiri merupakan kombinasi dari SID + frameSRL + Label berikut table perbandingan dari SSID dengan metode lain menggunakan parameter hasil estimasi parameter ROUGE-1 S_T=0.3, HR_{min}=0.6, ε=0.4, θ=15, α=0.8, T=0.3 dan ROUGE-2 [S_T=0.4, HR_{min}=0.5, ε=0.5, θ=30, α=0.7, T=0.8] dengan menggunakan data testing.

Tabel 4.10 Rata-Rata Jumlah *Cluster* yang Dibutuhkan dalam Pembentukan Ringkasan

Metode	**Metode Evaluasi**	
	ROUGE-1	**ROUGE-2**
SHCClustering + ClusterOrder + Local Importan + Global Important (LIGI)	0.294051	0.08150
SHCClustering+ClusterOrder+SID + KCK (SIDeKiCK)	0.289541	0.08194
SHCClustering + ClusterOrder + SID + FrameSRL + LabelMaching (SSID)	**0.29656**	**0.08992**

Metode SSID menghasilkan nilai rata-rata **0.29656** ROUGE-1 sedangkan ROUGE-2 dengan nilai **0.08992** sehingga menunjukkan bahwa hasil ringkasan

SSID mempunyai korelasi yang lebih baik dibandingkan dengan LIGI dan SIDeKiCK

Tabel 4.11 Rata-Rata Keterwakilan Kalimat pada Hasil Ringkasan

Metode	Keterwakilan kalimat
LIGI	5.38
SIDeKiCK	6.12
SSID	**7.00**

Tabel 4.11 menunjukkan bahwa setiap kalimat dalam ringkasan yang dihasilkan mewakili sejumlah kalimat dalam proses pemilihannya. Hasil Ringkasan dengan SSID (7.00) mempunyai rata-rata nilai keterwakilan lebih besar dari metode lain LIGI(5.38) dan SIDeKiCK(6.12). Setiap kalimat pada ringakasan SSID rata-rata mewakili 7 kalimat berita. Dengan kata lain metode SSID lebih baik dalam pememilihan kalimat-kalimat penting (*salient sentences*) dalam cluster dibandingkan dengan Metode LIGI dan SIDeKiCK berdasarkan jumlah kalimat yang diwakili oleh tiap kalimat pada ringkasan dan Nilai Rouge-1 dan Rouge-2.

Beberapa penyebab kecilnya nilai Rouge-1 dan Rouge-2 adalah diantaranya: Pada beberapa dataset metode yang diusulkan mengalami penurunan kinerja disebabkan oleh banyaknya kutipan dengan kalimat yang sama pada dataset sehingga menambah *noise distance* pada pemilihan kalimat representative, selain itu penguji kebenaran yang ada pada dataset DUC 2004 adalah ringkasan manual sehingga menurunkan kemungkinan nilai Rouge-n hasil analisa.

4.3.2. Pengembangan Lanjutan

Kualitas hasil ringkasan dari sistem peringkasan multi-dokumen yang dibangun berdasarkan framework seperti pada penelitian ini sangat dipengaruhi oleh tiga fase utama yaitu:

1. Fase *clustering* kalimat dengan SHC.
2. Fase pengurutan *cluster* kalimat *cluster importance*.
3. Fase pemilihan kalimat *representative cluster*.

Setiap metode dalam *framework* yang digunakan ini tentunya saling keterkaitan sehingga dibutuhkan kesesuaian antara metode yang digunakan. Berdasarkan pada *framework* yang dibangunt peneliti mengusulkan beberapa pengembangan lebih lanjut diantaranya:

1. Menambahkan penggunaan semantik dalam pembentukan *cluster* SHC.
2. Menggabungkan *Local Important* dan *Global Important* untuk menambahkan corelasi antar *cluster* dalam pemilihan kalimat representative.
3. Mempertimbangkan *synonim*, *antonym* dan beberapa tanda baca lainnya dalam melakukan *clustering*.
4. Menghilangkan kalimat-kalimat yang sama dalam cluster

BAB 5
KESIMPULAN

5.1 Kesimpulan

Pada bab ini dipaparkan kesimpulan yang yang diambil berdasarkan analisan dan hasil percobaan yang dilakukan terhadap metode yang diusulkan. Beberapa kesimpulan yang penulis ambil adalah sebagai berikut:

1. Ringkasan yang dihasilkan oleh metode SSID memberikan hasil yang lebih baik dalam pemilihan kalimat representatif dibuktikan berdasarkan nilai maksimum ROUGE-1 dan ROUGE-2. Dibuktikan berdasarkan korelasi *bigram* dan *unigram* dari hasil ringkasan sistem dan ringkasan manual yang terdapat dalam dataset. Metode SSID menghasilkan rata-rata nilai 0.29656 pada analisa dengan ROUGE-1, meningkat 0.85% jika dibandingkan dengan LIGI dan 2.42% dibandingkan dengan SIDeKiCK. Pada analisa ROUGE-2 SSID menghasilkan rata-rata nilai 0.08992, meningkat 10.33% jika dibandingkan dengan LIGI dan meningkat 9.73% dibandingkan dengan SIDeKiCK.
2. Jumlah Cluster yang terbentuk pada proses SSID lebih padat hal ini dibuktikan dengan sedikitnya jumlah cluster yang terbentuk. Pada proses SSID rata-rata jumlah cluster yang terbentuk 182.52, lebih sedikit jika dibandingkan dengan SIDeKick berjumlah 217.44 *cluster* dan LIGI 222.16 *cluster*.
3. Dengan menambahkan peran semantik pada metode *position text graph* mampu meningkatkan kualitas *salient* pada pemilihan kalimat representatif *cluster*. Dibuktikan dengan jumlah Cluster yang terbentuk pada proses SSID lebih padat dan dan mewakili lebih banyak kalimat dari metode lain. Satu kalimat pada hasil ringkasan rata-rata mewakili 7 kalimat dari dokumen sumber sedangkan LIGI rata-rata mewakili 5.38 kalimat dan SIDeKiCK rata-rata mewakili 6.12 kalimat.

4. Pada beberapa dataset SSID mengalami penurunan kinerja dari metode lain hal ini disebabkan banyaknya kalimat-kalimat kutipan yang sama sehingga kalimat-kalimat ini menjadi *noise distance* pada proses pemilihan kalimat representative *cluster*.

5.2 Saran

Metode peringkasan ini terdiri dari beberapa kombinasi fase diantaranya fase *Clustering dengan SHC*, fase *Cluster Order* dan fase pemilihan kalimat representative. Beberapa saran yang penulis simpulkan dari penelitian ini adalah:

1. Pada fase awal pembentukan *cluster* agar dipilih metode *clustering* yang memperhatikan semantik dari kalimat tapi tetap menjaga korelasi antar kalimat dalam *cluster* tersebut.
2. Mengkombinasikan metode SSID dengan beberapa fitur lainnya seperti *Local Important* dan *Global Important, font based, sentence position,* dan sebagainya.

LAMPIRAN

Lampiran 1. Tabel Hasil Perbandingan Performa

Pada Analisa hasil perfoma LIGI, SIDeKiCK dan SSID dilakukan terhadap data testing dengan parameter optimal.

Tabel 1. Hasil perbandingan performa SSID dan Metode Lain

Dataset	Performa					
	ROUGE-1			ROUGE-2		
	LIGI	SIDeKiCK	SSID	LIGI	SIDeKCK	SSID
d30042t	0.33333	0.28846	0.37879	0.11215	0.07477	0.19298
d30044t	0.30400	0.40625	0.36364	0.01681	0.07937	0.06612
d30045t	0.34783	0.38596	0.31496	0.12403	0.17857	0.07143
d30046t	0.34783	0.32653	0.36066	0.01695	0.06250	0.03922
d30047t	0.36975	0.35897	0.38261	0.12030	0.12174	0.11864
d30048t	0.21705	0.15385	0.21212	0.08333	0.03200	0.05128
d30049t	0.32168	0.36800	0.30882	0.09160	0.11382	0.06504
d30050t	0.21875	0.24390	0.14388	0.05310	0.04688	0.05042
d30051t	0.31933	0.24286	0.29688	0.08759	0.07246	0.13675
d30053t	0.21898	0.30769	0.28800	0.06107	0.06557	0.03509
d30055t	0.32061	0.38983	0.38095	0.12214	0.11765	0.10714
d30056t	0.31858	0.24194	0.27027	0.01600	0.01587	0.07547
d30059t	0.28571	0.37097	0.27586	0.07080	0.07080	0.09091
d31001t	0.26923	0.28571	0.28125	0.13223	0.10909	0.09524
d31008t	0.39024	0.25197	0.36496	0.14063	0.11864	0.10619
d31009t	0.37288	0.40650	0.46875	0.13115	0.13913	0.16529
d31013t	0.21239	0.26154	0.23622	0.01852	0.05607	0.07407
d31022t	0.27778	0.30189	0.23214	0.05042	0.07843	0.04839
d31026t	0.25806	0.25225	0.24762	0.07080	0.05217	0.09259
d31031t	0.11111	0.16364	0.17476	0.07080	0.03704	0.05085
d31032t	0.38261	0.25397	0.36364	0.18349	0.09677	0.22018

d31033t	0.25000	0.17476	0.26786	0.01724	0.01980	0.01739
d31038t	0.32479	0.30909	0.22018	0.12500	0.13793	0.09009
d31043t	0.32479	0.27826	0.32520	0.10435	0.07207	0.10256
d31050t	0.25397	0.21374	0.25397	0.01695	0.07937	0.08475
Rata-Rata	0.29405	0.28954	**0.29656**	0.08150	0.08194	**0.08992**

Hal itu dibuktikan dengan kinerja metode SSID yang mampu menghasilkan rata-rata nilai **0.29656** untuk ROUGE-1 (meningkat 2.42%) dan **0.08992** untuk ROUGE-2 (meningkat 9.73%) dibandingkan dengan metode SIDeKiCK (metode sebelumnya).

Lampiran 2. Perbandingan Rouge-1 dan Rouge-2

Tabel 2. Hasil perbandingan performa Rouge-1 dan Rouge-2 metode SSID

Cluster Dataset	ROUGE-1	ROUGE-2
d30050t	**0.14388**	0.04545
d31031t	**0.17475**	0.01834
d30048t	**0.21212**	0.03448
d31038t	0.22018	0.13793
d31022t	0.23214	0.05405
d31013t	0.23622	0.03539
d31026t	0.24761	0.05405
d31050t	0.25396	0.01739
d31033t	0.26785	0.01724
d30056t	0.27027	0.01801
d30059t	0.27586	0.09259
d31001t	0.28125	0.08547
d30053t	0.288	0.06451
d30051t	0.29687	0.10937
d30049t	0.30882	0.08695
d30045t	0.31496	0.07407
d31043t	0.3252	0.06956
d30046t	0.36065	0.06611
d30044t	0.36363	0.04761
d31032t	0.36363	0.22641
d31008t	0.36496	0.13559
d30042t	0.37878	0.01851
d30055t	0.38095	0.17054
d30047t	0.3826	0.12799
d31009t	0.46875	0.08928
rata-rata	0.29656	0.07588

Data hasil testing diurutkan berdasarkan Rouge-1 terkecil yang kemudian akan dilakukan analisa terhadap data tersebut Dari proses testing menunjukkan cluster d30050t mempunyai nilai rouge-1 yang terkecil.

Lampiran 3. Perbandingan Jumlah Cluster pada Data Testing

Tabel 3. Hasil perbandingan performa SSID dan Metode Lain

Cluster Dataset	**LIGI**	**SIDeKiCK**	**SSID**
d30042t	122	117	103
d30044t	120	118	97
d30045t	302	293	237
d30046t	357	351	308
d30047t	164	158	129
d30048t	214	211	186
d30049t	207	201	166
d30050t	358	347	300
d30051t	214	211	176
d30053t	191	188	149
d30055t	99	96	85
d30056t	115	112	103
d30059t	153	150	138
d31001t	205	202	169
d31008t	354	343	311
d31009t	126	124	100
d31013t	400	396	320
d31022t	204	201	180
d31026t	345	337	279
d31031t	290	288	235
d31032t	124	122	103
d31033t	475	466	359
d31038t	125	123	100
d31043t	98	93	79
d31050t	192	188	151
Rata-rata	**222.16**	**217.44**	**182.52**

Pada proses peringkasan dokumen data berita dikelompokkan kedalam beberapa *cluster*. Dalam proses peringkasan rata-rata jumlah cluster yang terbentuk adalah LIGI= **222.16**, SIDeKiCK =**217.44** dan SSID= **182.52.** Dari rata-rata cluster yang terbentuk memperlihatkan bahwa SSID mempunyai rata-rata cluster yang paling sedikit. Sehingga dapat disimpulkan bahwa tiap cluster mempunyai anggota kalimat yang lebih banyak dari metode lain (LIGI dan SIDeKiCK)

Halaman ini sengaja dikosongkan

Lampiran 4. Analisa Keterwakilan Kalimat

Tabel 4. Jumlah Cluster yang Dibutuhkan dalam Penyusunan Kalimat Ringkasan

	LIGI	**SIDeKiCK**	**SSID**
d30042t	3	3	4
d30044t	4	5	4
d30045t	4	4	4
d30046t	3	3	3
d30047t	3	3	3
d30048t	4	4	3
d30049t	3	3	3
d30050t	3	4	3
d30051t	3	4	4
d30053t	4	4	4
d30055t	4	5	4
d30056t	4	5	4
d30059t	4	5	4
d31001t	3	4	3
d31008t	3	3	4
d31009t	4	5	4
d31013t	2	4	3
d31022t	3	4	3
d31026t	3	4	3
d31031t	3	3	3
d31032t	4	5	4
d31033t	3	2	3
d31038t	3	3	3
d31043t	4	4	4
d31050t	3	4	3
rata-rata	**3.36**	**3.88**	**3.48**

Menunjukkan bahwa rata-rata cluster yang diproses pada setiap metode sekitar 3.50

Tabel 5. Jumlah Kalimat yang dibandingkan dalam Penyusunan Ringkasan

	LIGI	**sidekick**	**SSID**
d30042t	15	22	23
d30044t	18	23	19
d30045t	30	32	35
d30046t	11	14	11
d30047t	14	18	26
d30048t	15	20	18
d30049t	12	17	27
d30050t	19	25	19
d30051t	22	29	32
d30053t	18	21	23
d30055t	26	38	30
d30056t	17	32	21
d30059t	20	29	23
d31001t	14	20	25
d31008t	19	18	33
d31009t	17	25	25
d31013t	6	20	17
d31022t	28	37	30
d31026t	24	27	28
d31031t	15	19	25
d31032t	21	27	21
d31033t	19	15	26
d31038t	14	15	15
d31043t	24	26	31
d31050t	14	25	26
	18.08	**23.76**	**24.36**

Tabel 6. Keterwakilan Kalimat

	LIGI	**sidekick**	**SSID**
d30042t	5.00	7.33	5.75
d30044t	4.50	4.60	4.75
d30045t	7.50	8.00	8.75
d30046t	3.67	4.67	3.67
d30047t	4.67	6.00	8.67
d30048t	3.75	5.00	6.00
d30049t	4.00	5.67	9.00
d30050t	6.33	6.25	6.33
d30051t	7.33	7.25	8.00
d30053t	4.50	5.25	5.75
d30055t	6.50	7.60	7.50
d30056t	4.25	6.40	5.25
d30059t	5.00	5.80	5.75
d31001t	4.67	5.00	8.33
d31008t	6.33	6.00	8.25
d31009t	4.25	5.00	6.25
d31013t	3.00	5.00	5.67
d31022t	9.33	9.25	10.00
d31026t	8.00	6.75	9.33
d31031t	5.00	6.33	8.33
d31032t	5.25	5.40	5.25
d31033t	6.33	7.50	8.67
d31038t	4.67	5.00	5.00
d31043t	6.00	6.50	7.75
d31050t	4.67	6.25	8.67
rata-rata	**5.38**	**6.12**	**7.00**

Tabel keterwakilan Kalimat didasarkan pada jumlah cluster dan jumlah anggota kalimat dalam cluster dimanaa SSID rata-rata mewakili 7.00 kalimat

Halaman ini sengaja dikosongkan

Lampiran 5. Analisa Proses Peringkasan Pada Data Testing

Hasil cluster dan kalimat yang mewakili pada dataset d30050t. Data d30050t merupakan dataset yang mempunyai nilai Rouge-1 paling sedikit sehingga disini penulis merasa perlu untuk melakukan analisa penyebab minimnya nilai Rouge pada beberapa data salah satunya d30050t.

Dari hasil cluster SSID data d30050t empat cluster terpenting yang digunakan dalam pembentukan ringkasan atau pemilihan kalimat representative adalah sebagai berikut:

Tabel 8. Hasil Top Cluster Pada Dataset d30050t

Cluster	**Anggota Cluster**
Cluster 1	Publicly, officials in the organization, the Human Rights Campaign, said they were still deliberating their position in the closely watched race, considered among the tightest in the nation. 6 18 White House officials said they did not know whether the three had made such appeals, and Human Rights Campaign officials declined to comment. 8 341 ``Chuck Schumer has been a strong supporter of issues that are important to gay communities," said a senior White House official who spoke on the condition of anonymity. 6 24 ``Chuck Schumer has been a strong supporter of issues that are important to gay communities," said a senior White House official who spoke on the condition of anonymity. 6 360 For that reason, some White House officials say they think a Human Rights Campaign endorsement of D'Amato will be almost meaningless among gay voters. 4 43 For that reason, some White House officials say they think a Human Rights Campaign endorsement of D'Amato will be almost meaningless among gay voters. 4 348 To gay supporters of D'Amato, an endorsement by the Human Rights Campaign would signify the political maturation

	of the gay electorate and help the organization insulate itself from accusations that it is too close to the Democratic Party. 8 318 Publicly, officials in the organization, the Human Rights Campaign, said they were still deliberating their position in the closely watched race, considered among the tightest in the nation. 6 334 Democrats and advocates in both parties who support rights for gay people said that Vice President Al Gore, Hillary Rodham Clinton and Secretary for Health and Human Services Donna Shalala have made personal appeals to Human Rights Campaign officials urging them not to endorse D'Amato. 9 335 White House officials said they did not know whether the three had made such appeals, and Human Rights Campaign officials declined to comment. 8
Kalimat yang mewakili	Democrats and advocates in both parties who support rights for gay people said that Vice President Al Gore, Hillary Rodham Clinton and Secretary for Health and Human Services Donna Shalala have made personal appeals to Human Rights Campaign officials urging them not to endorse D'Amato.
Cluster 2	1. 32 Those who support a D'Amato endorsement, including top officials within the Human Rights Campaign, contend that in the current political climate, where Congress is almost certain to remain under Republican control after November, gay people must build alliances with moderate Republicans. 10 2. 33 Human Rights Campaign officials also say their standing policy is to support friendly incumbents, even when their challengers have better voting records on gay issues. 8

	3. 349 Those who support a D'Amato endorsement, including top officials within the Human Rights Campaign, contend that in the current political climate, where Congress is almost certain to remain under Republican control after November, gay people must build alliances with moderate Republicans. 10 4. 14 Although the Human Rights Campaign is bipartisan, it has been very close to the Clinton administration, has many Democrats on its board and receives much of its money from Democratic contributors. 5 5. 350 Human Rights Campaign officials also say their standing policy is to support friendly incumbents, even when their challengers have better voting records on gay issues. 8 6. 31 To gay supporters of D'Amato, an endorsement by the Human Rights Campaign would signify the political maturation of the gay electorate and help the organization insulate itself from accusations that it is too close to the Democratic Party. 8
Kalimat yang mewakili	Those who support a D'Amato endorsement, including top officials within the Human Rights Campaign, contend that in the current political climate, where Congress is almost certain to remain under Republican control after November, gay people must build alliances with moderate Republicans.
Cluster 3	1. 17 Democrats and advocates in both parties who support rights for gay people said that Vice President Al Gore, Hillary Rodham Clinton and Secretary for Health and Human Services Donna Shalala have made personal

	appeals to Human Rights Campaign officials urging them not to endorse D'Amato. 9 2. 317 White House officials and gay Democrats, concerned that the nation's largest gay and lesbian political organization is about to endorse Sen. Alfonse D'Amato for re-election, are intensely lobbying the group to try to shift its support to the Democratic challenger, Rep. Charles Schumer. 8 3. 15 Largely because of the group's strong Democratic ties, gay Democrats, New York liberals and White House officials are infuriated that it is even considering endorsing D'Amato, who also runs on the Right to Life and Conservative Party lines and often receives high ratings from the Christian Coalition, which typically opposes legislation on civil rights for gay people. 12
Kalimat yang mewakili	Democrats and advocates in both parties who support rights for gay people said that Vice President Al Gore, Hillary Rodham Clinton and Secretary for Health and Human Services Donna Shalala have made personal appeals to Human Rights Campaign officials urging them not to endorse D'Amato.

Cluster ini mempunyai banyak kalimat yang sama baik berupa pendapat, opini atau kutipan dari sumber lain sehingga kalimat-kalimat ini dapat merusak pembentukan dan pemilihan kalimat representatis karena menyebabkan noise jarak satu kalimat dengan kalimat yang lain. Begitu pula dengan data lain dataset d31038t, d30045t, d31022t dan d30050t.

Lampiran 6. Uji-t Berpasangan Dua Sisi

Pada penelitian ini digunakan nilai α (taraf signifikansi) = 0.2 dan derajat bebas = n-1 = 25. Sehingga nilai tabel berdasarkan nilai yang diambil dari Tabel t yaitu t Tabel(ssid,df)= 1.318. Berikut adalah hasil uji t berpasangan dua sisi terhadap empat buah pasangan nilai distribusi nilai yaitu nilai ROUGE-1 dan nilai ROUGE-2 yang dihasilkan dari metode SSID, SIDeKiCK dan LIGI, dimana:

h_0 : $\mu_{data\ 1} = \mu_{data\ 2}$ atau h_0 : $\mu_D = 0$ (tidak ada perbedaan signifikan)

h_1 : $\mu_{data\ 1} \neq \mu_{data\ 2}$ atau h_1 : $\mu_D \neq 0$ (terdapat perbedaan signifikan)

PERUMUSAN HIPOTESIS :

Secara Matematis

h_0 : $\mu_{data\ 1} = \mu_{data\ 2}$ atau h_0 : $\mu_D = 0$

h_1 : $\mu_{data\ 1} \neq \mu_{data\ 2}$ atau h_1 : $\mu_D \neq 0$

dimana:

$\mu_{data\ 1}$ adalah rata-rata dari distribusi *data1*

$\mu_{data\ 2}$ adalah rata-rata dari distribusi *data2*

$\mu_D = \mu_{data\ 1} - \mu_{data\ 2}$

Secara Umum

h_0 : distribusi nilai *data1* dan *data2* tidak berbeda signifikan

h_1 : distribusi nilai *data1* dan *data2* berbeda signifikan

STATISTIK UJI : $t_{hitung} = \frac{\bar{d}}{s_d / \sqrt{n}}$, dimana:

$\bar{d}$ adalah rata-rata selisih dari setiap data per-kolom, yang dihitung dengan rumus:

$\bar{d} = \frac{\sum_{i=1}^{n} d_i}{n}$, dimana:

d_i = selisih pasangan data yaitu $d_i = data\ 1_i - data\ 2_i$, i = 1,2,3,..., n (jumlah data)

$s_d = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n} (d_i - \bar{d})^2}$, dimana:

s_d = Standar Deviasi selisih pasangan data

Pengambilan Keputusan Hipotesis Uji-t Berpasangan:

1. Jika $-t_{hitung} > -t_{tabel}$ atau $t_{hitung} > t_{tabel}$ atau $|t_{hitung}| > |t_{tabel}|$, maka keputusan:

   ***h_o* ditolak maka ada perbedaan signifikan**.

2. Jika $-t_{hitung} < -t_{tabel}$ atau $t_{hitung} < t_{tabel}$ atau $|t_{hitung}| < |t_{tabel}|$, maka keputusan:

   ***h_o* diterima maka tidak ada perbedaan signifikan**.

Tabel 7. Perbandingan Nilai ROUGE-1 antara Metode SSID dengan Metode SIDeKiCK

Cluster	SSID	SIDeKCK	Selisih
d30042t	0.37879	0.28846	0.0903
d30044t	0.36364	0.40625	-0.0426
d30045t	0.31496	0.38596	-0.0710
d30046t	0.36066	0.32653	0.0341
d30047t	0.38261	0.35897	0.0236
d30048t	0.21212	0.15385	0.0583
d30049t	0.30882	0.36800	-0.0592
d30050t	0.14388	0.24390	-0.1000
d30051t	0.29688	0.24286	0.0540
d30053t	0.28800	0.30769	-0.0197
d30055t	0.38095	0.38983	-0.0089
d30056t	0.27027	0.24194	0.0283
d30059t	0.27586	0.37097	-0.0951
d31001t	0.28125	0.28571	-0.0045
d31008t	0.36496	0.25197	0.1130
d31009t	0.46875	0.40650	0.0622
d31013t	0.23622	0.26154	-0.0253
d31022t	0.23214	0.30189	-0.0697
d31026t	0.24762	0.25225	-0.0046
d31031t	0.17476	0.16364	0.0111
d31032t	0.36364	0.25397	0.1097
d31033t	0.26786	0.17476	0.0931
d31038t	0.22018	0.30909	-0.0889

d31043t	0.32520	0.27826	0.0469
d31050t	0.25397	0.21374	0.0402

Tabel 8. Uji-*t* Berpasangan Dua Sisi Perbandingan Nilai ROUGE-1 Metode SSID dan SIDeKiCK

uji t berpasangan	
jumlah data	25
df (degree freedom)	24
Taraf keyakinan (ssid)	0.2
t Tabel(ssid,df)	1.318
Mean SSID	0.2966
Mean SIDeKiCK	0.2895
selisih Mean	0.007
Standar Deviasi Selisih	0.061
T Hitung	0.578
Jawaban Hipotesis	H0 Diterima
Perbedaan	Tidak Ada Perbedaan Signifikan

Keputusan Hipotesis T Paired:

1. *t hitung* > *-t tabel* atau *t hitung* > *t tabel* atau *Absolut t hitung* > *Absolut t tabel*: Ada Perbedaan Signifikan Atau *H0* Ditolak.
2. *t hitung* < *-t tabel* atau *t hitung* < *t tabel* atau *Absolut t hitun*g < *Absolut t tabel*: Tidak Ada Perbedaan Signifikan Atau *H0* Diterima.

Ho : *Rouge-1 SSID* = Rouge-1 *SIDeKICK*

Ha : *Rouge-1 SSID* != Rouge-1 *SIDeKICK*

Tabel 9. Perbandingan Nilai ROUGE-2 antara Metode SSID dengan Metode SIDeKiCK

Cluster	SSID	SIDeKiCK	Selisih
d30042t	0.19298	0.07477	0.1182
d30044t	0.06612	0.07937	-0.0132
d30045t	0.07143	0.17857	-0.1071
d30046t	0.03922	0.06250	-0.0233
d30047t	0.11864	0.12174	-0.0031
d30048t	0.05128	0.03200	0.0193
d30049t	0.06504	0.11382	-0.0488
d30050t	0.05042	0.04688	0.0035
d30051t	0.13675	0.07246	0.0643
d30053t	0.03509	0.06557	-0.0305
d30055t	0.10714	0.11765	-0.0105
d30056t	0.07547	0.01587	0.0596
d30059t	0.09091	0.07080	0.0201
d31001t	0.09524	0.10909	-0.0139
d31008t	0.10619	0.11864	-0.0124
d31009t	0.16529	0.13913	0.0262
d31013t	0.07407	0.05607	0.0180
d31022t	0.04839	0.07843	-0.0300
d31026t	0.09259	0.05217	0.0404
d31031t	0.05085	0.03704	0.0138
d31032t	0.22018	0.09677	0.1234
d31033t	0.01739	0.01980	-0.0024
d31038t	0.09009	0.13793	-0.0478
d31043t	0.10256	0.07207	0.0305
d31050t	0.08475	0.07937	0.0054

Tabel 10. Uji-*t* Berpasangan Dua Sisi Perbandingan Nilai ROUGE-2 Metode SSID dan SIDeKiCK

Uji t Berpasangan	
jumlah data	25
df (degree freedom)	24
Taraf keyakinan (α)	0.2
t Tabel(α,df)	1.318
Mean SSID	0.0899
Mean SIDeKiCK	0.0819

selisih Mean	0.008
Standar Deviasi Selisih	0.050
T Hitung	0.805
Jawaban Hipotesis	H0 Diterima
Perbedaan	Tidak Ada Perbedaan Signifikan

Keputusan Hipotesis T Paired:

3. *t hitung* > *-t tabel* atau *t hitung* > *t tabel* atau *Absolut t hitung* > *Absolut t tabel*: Ada Perbedaan Signifikan Atau *H0* Ditolak.
4. *t hitung* < *-t tabel* atau *t hitung* < *t tabel* atau *Absolut t hitung* < *Absolut t tabel*: Tidak Ada Perbedaan Signifikan Atau *H0* Diterima.

Ho : *Rouge-2 SSID* = Rouge-2 *SIDeKICK*

Ha : *Rouge-2 SSID* != Rouge-2 *SIDeKICK*

Tabel 11. Perbandingan Nilai ROUGE-1 antara Metode SSID dengan Metode LIGI

Cluster	**SSID**	**SIDeKiCK**	**Selisih**
d30042t	0.19298	0.07477	0.1182
d30044t	0.06612	0.07937	-0.0132
d30045t	0.07143	0.17857	-0.1071
d30046t	0.03922	0.06250	-0.0233
d30047t	0.11864	0.12174	-0.0031
d30048t	0.05128	0.03200	0.0193
d30049t	0.06504	0.11382	-0.0488
d30050t	0.05042	0.04688	0.0035
d30051t	0.13675	0.07246	0.0643
d30053t	0.03509	0.06557	-0.0305
d30055t	0.10714	0.11765	-0.0105
d30056t	0.07547	0.01587	0.0596
d30059t	0.09091	0.07080	0.0201
d31001t	0.09524	0.10909	-0.0139
d31008t	0.10619	0.11864	-0.0124
d31009t	0.16529	0.13913	0.0262
d31013t	0.07407	0.05607	0.0180
d31022t	0.04839	0.07843	-0.0300

d31026t	0.09259	0.05217	0.0404
d31031t	0.05085	0.03704	0.0138
d31032t	0.22018	0.09677	0.1234
d31033t	0.01739	0.01980	-0.0024
d31038t	0.09009	0.13793	-0.0478
d31043t	0.10256	0.07207	0.0305
d31050t	0.08475	0.07937	0.0054

Tabel 12. Uji-*t* Berpasangan Dua Sisi Perbandingan Nilai ROUGE-1 Metode SSID dan LIGI

uji t berpasangan	
jumlah data	25
df (degree freedom)	24
Taraf keyakinan (α)	0.2
t Tabel(α,df)	1.318
Mean SSID	0.0899
Mean SIDeKiCK	0.0819
selisih Mean	0.008
Standar Deviasi Selisih	0.058
T Hitung	0.684
Jawaban Hipotesis	H0 Diterima
Perbedaan	Tidak Ada Perbedaan Signifikan

DAFTAR PUSTAKA

Amandeep Kaur Mann and Navneet Kaur. (2013)"*Review Paper on clustering Techniques*." *Software & Data Engineering*. Global Journal of Computer Science and Technology.

Anders Björkelund, Love Hafdell, and Pierre Nugues. Multilingual semantic role labeling. In Proceedings of The Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009), pages 43--48, Boulder, June 4--5 2009.

Anders Björkelund, Bernd Bohnet, Love Hafdell, and Pierre Nugues. A high-performance syntactic and semantic dependency parser. In Coling 2010: Demonstration Volume, pages 33-36, Beijing, August 23-27 2010.

Barzilay, R. and McKeown, K. R. (2005) "*Sentence Fusion for Multi document News Summarization*." *Computational Linguistics,* 2009. ICSAP 2009. International Conference on IEEE.

Bjorkelund, A., Bohnet, B., Hafdell, L. and Nugues, P. (2009) "*Multilingual Semantic Role Labeling.'Department of Computer Science*, Lund University,

Bjorkelund, A., Bohnet, B., Hafdell, L. and Nugues, P. (2010), "A Hig*h-Performance Syntactic and Semantic Dependency Parser*." Department of Computer Science, Lund University.

Barzilay, R., Kathleen R. M and Elhadad M. (1999). "*Information Fusion in the Context of Multi-Document Summarization*." Dept. of Computer Science, Columbia University.

Cai, X. and Li, W. (2013),"Ranking Through *clustering*: An Integrated Approach to Multi-Document Summarization." *IEEE transactions on audio, speech, and language processing*. IEEE.

Carbonell, Jaime G. dan Goldstein, J,. (1998),*" The Use of MMR and Diversity-Based Reranking for Reodering Documents and Producing Summaries"* Proceedings of the 21st meeting of International ACM SIGIR Conference, 335-336.

Erkan, G. dan Radev, D. R. (2004), "*LexRank: Graph-based Lexical Centrality as Salience in Text Summarization.*" Journal of Artificial Intelligence.

Erkan, G. and Radev, D. R. (2004), "*LexPageRank: Prestige in multi-document text summarization*," in Proc. EMNLP'04.

Ge, S. S., Zhang, Z. and He H. (2011), "*Weighted Graph Model Based Sentence clustering and Ranking for Document Summarization*."

Gildea, D. and Jurafskyy, D. (2001) "*Automatic Labeling of Semantic Role*s." International Computer Science Institiute.Gupta, V. K. and Siddiqui, T. J. (2012), "*Multi-Document Summarization Using Sentence clustering*." IEEE Proceedings of 4th International Conference onIntelligent Human Computer Interaction.

Hammouda, K. M. and Kamel, M. S. (2004) "*Efficient Phrase-Based Document Indexing for Web Document clustering*." IEEE Transactions On Knowledge And Data Engineering.

He, T., Li, F., Shao, W. Chen, J. and Ma, L. (2008), "*A New Feature-Fusion Sentence Selecting Strategy for Query-Focused Multi-Document Summarization*." International Conference on Advanced Language Processing and Web Information Technology. IEEE.

Judith D. Schlesinger, Dianne P. O'Leary, and John M. Conroy. (2008), "*Arabic/English Multi-document Summarization with CLASSY—The Past and the Futur*e." Springer-Verlag Berlin Heidelberg.

Jaime G. Carbonell and Jade Goldstein. (2005),"*The Use of MMR and Diversity-Based Reranking for Reodering Documents and Producing Summaries*." Proceedings of the 21st meeting of International ACM SIGIR Conference.

Kruengkrai, C. and Jaruskulchai, C. (2003), "*Generic Text Summarization Using Local and Global Properties of Sentences*." Proceedings of the IEEE/WIC International Conference on Web Intelligence *(WI'03)*. IEEE.

Knight, K. and Marcu, D. (2002)"Summarization beyond sentence extraction: A probabilistic approach to sentence compression." Artificial Intelligence. Elsevier.

Kogilavani, A.and Balasubramani, Dr.P. (2010), "*clustering and feature Specific Sentenceextraction Based Summarization Of Multiple Documents*."

International journal of computer science & information Technology (IJCSIT).

Sarkar, K. (2009), "*Sentence clustering Based Summarization of Multiple Texs Document.*" Interntional Journal of Computing Science and Communication Technologies.

Schlesinger, Judith D. O'Leary. Dianne P. and Conroy,John M. (2008),*" Arabic/English Multi-document Summarization with CLASSY—The Past and the Future".* IDA/Center for Computing Sciences.

Lin, C. Y. (2004), "*ROUGE: a Package for Automatic Evaluation of Summaries*", In Proceedings of Workshop on Text Summarization Brances Out, Eds: Moens, M. F. dan Szpakowicz, S., Association for Computational Linguistics, Barcelona, hal. 74-81.

Ma, Xiao-Chen., Yu, Gui-Bin., and Ma, Liang. (2009), "*Multi-document Summarization Using clustering Algorithm*." *Computational Linguistics*. IEEE.

Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55-60

Mihalcea, R. and Tarau, P, (2005) "*A language independent algorithm for single and multiple document summarization*," in Proc. IJCNLP-05.

Meena, Y. K., Jain, A., and Gopalani, D. (2014)"*Survey on Graph and cluster Based Approaches in Multi-document Text Summarization*." International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014), IEEE.

Palmer, M., Gildea, D., dan Kingsbury P. (2005) "*The Proposition Bank: An Annotated Corpus of Semantic Roles*." *Association for Computational Linguistics*.

Pradhan, S., Ward, W.., Hacioglu, K., Martin, James H. and Dan, Jurafsky .(2004)."*Shallow Semantic Parsing using Support Vector Machines*."

Suputra, I.P.G.H, Arifin, A.Z, Yuniarti, A. (2013),"*Pendekatan Positional Text Graph Untuk Pemilihan Kalimat Representatif cluster Pada Peringkasan Multi-Dokumen*".Jurnal Ilmu Komputer Universitas Udayana.

Surdeanu, M., Harabagiu S., Williams, J. and Aarseth P. (2003) "*Using Predicate-Argument Structures for Information Extraction*." Language Computer Corp.Richardson, Texas 75080, USA.

Wan, X., Yang, J. and Xiao, J. (2007) "*Sentence Fusion for Multi document News Summarization.*" Association for Computational Linguistics, Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics.

BIOGRAFI PENULIS

Gus Nanang Syaifuddiin. Lahir di Ponorogo, tanggal 14 Agustus 1989. Penulis Tinggal di daerah Ponorogo Jawa Timur. Mengenyam pendidikan Sekolah Dasar di SDN 1 Jalen pada tahun 1995, Sekolah Lanjutan Pondok Modern Arrisalah Slahung Ponorogo dari tahun 2002 sampai 2008. Pada tahun 2008 melanjutkan kuliah sarjana di Jurusan Teknik Informatik di Universitas Muhammadiyah Ponorogo dan kemudian pada tahun 2013 melanjutkan pendidikan pascasarjana di Program Magister Teknik Informatika,ITS Surabaya.