

**IMPLEMENTASI ALGORITMA DIJSKTRA UNTUK
PERINGKASAN OTOMATIS ARTIKEL ILMIAH
BERBAHASA INGGRIS**

Skripsi



oleh
YANUAR BUDI PRASETYO
22104912

PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
2014

**IMPLEMENTASI ALGORITMA DIJSKTRA UNTUK
PERINGKASAN OTOMATIS ARTIKEL ILMIAH
BERBAHASA INGGRIS**

Skripsi



Diajukan kepada Program Studi Teknik Informatika Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana
Sebagai Salah Satu Syarat dalam Memperoleh Gelar
Sarjana Komputer

Disusun oleh

YANUAR BUDI PRASETYO
22104912

PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
2014

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

IMPLEMENTASI ALGORITMA DIJKSTRA UNTUK PERINGKASAN OTOMATIS ARTIKEL ILMIAH BERBAHASA INGGRIS

yang saya kerjakan untuk melengkapi sebagian persyaratan menjadi Sarjana Komputer pada pendidikan Sarjana Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Kristen Duta Wacana, bukan merupakan tiruan atau duplikasi dari skripsi keserjanaan di lingkungan Universitas Kristen Duta Wacana maupun di Perguruan Tinggi atau instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Jika dikemudian hari didapati bahwa skripsi ini adalah hasil plagiasi atau tiruan dari skripsi lain, saya bersedia dikenai sanksi yakni pencabutan gelar keserjanaan saya.

Yogyakarta, 4 Juli 2014



YANUAR BUDI PRASETYO

22104912

HALAMAN PERSETUJUAN

Judul Skripsi : IMPLEMENTASI ALGORITMA DIJKSTRA
UNTUK PERINGKASAN OTOMATIS ARTIKEL
ILMIAH BERBAHASA INGGRIS

Nama Mahasiswa : YANUAR BUDI PRASETYO

N I M : 22104912

Matakuliah : Skripsi (Tugas Akhir)

Kode : TIW276

Semester : Genap

Tahun Akademik : 2013/2014

Telah diperiksa dan disetujui di
Yogyakarta,
Pada tanggal 4 Juli 2014

Dosen Pembimbing I



Gloria Virginia, S.Kom., MAI, Ph.D.

Dosen Pembimbing II



Antonius Rachmat C., SKom., M.Cs

HALAMAN PENGESAHAN

IMPLEMENTASI ALGORITMA DIJKSTRA UNTUK PERINGKASAN OTOMATIS ARTIKEL ILMIAH BERBAHASA INGGRIS

Oleh: YANUAR BUDI PRASETYO / 22104912

Dipertahankan di depan Dewan Penguji Skripsi
Program Studi Teknik Informatika Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana - Yogyakarta
Dan dinyatakan diterima untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Komputer
pada tanggal 12 Juli 2014

Yogyakarta, 4 Juli 2014
Mengesahkan,

Dewan Penguji:

1. Gloria Virginia S.Kom MAI, Ph.D.
2. Antonius Machmat C, S.Kom., M.Cs
3. Aloysius Airlangga Bajuadji, S.Kom., M.Eng.
4. Hendro Setiadi, M.Eng




Dekan



(Drs. Wimmie Handwidjojo, MIT.)

Ketua Program Studi



(Nugroho Agus Haryono, M.Si)

UCAPAN TERIMA KASIH

Puji dan syukur penulis panjatkan kepada Tuhan kehadiran Tuhan Yang Maha Esa yang telah melimpahkan rahmat dan anugerah sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul Implementasi Algoritma Dijkstra untuk Peringkasan Otomatis Jurnal Ilmiah Berbahasa Inggris.

Penulisan laporan ini merupakan kelengkapan dan pemenuhan dari salah satu syarat dalam memperoleh gelar Sarjana Komputer. Selain itu bertujuan melatih mahasiswa untuk dapat menghasilkan suatu karya yang dapat dipertanggungjawabkan secara ilmiah, sehingga dapat bermanfaat bagi penggunanya.

Dalam menyelesaikan pembuatan program dan laporan Tugas Akhir ini, penulis telah banyak menerima bimbingan, saran, dan masukan dari berbagai pihak, baik secara langsung maupun secara tidak langsung. Untuk itu dengan segala kerendahan hati, pada kesempatan ini penulis menyampaikan ucapan terima-kasih kepada :

1. Ibu Gloria Virginia, S.Kom., MAI, Ph.D selaku dosen pembimbing I yang telah memberikan bimbingan dengan sabar dan baik kepada penulis, juga kepada
2. Bpk. Antonius Rachmat C., S.Kom, M.Cs selaku dosen pembimbing II atas bimbingan, petunjuk, dan saran yang diberikan selama pengerjaan Tugas Akhir ini sejak awal hingga akhir, juga kepada
3. Dosen-dosen Universitas Kristen Duta Wacana yang telah membantu memberikan pengarahan dan masukan kepada penulis.
4. Keluarga tercinta yang setia memberikan dukungan, doa, dan semangat.
5. Lina Kusuma Dewi atas saran, dukungan, dan semangat yang diberikan.
6. Pihak lain yang tidak dapat penulis sebutkan satu per satu, sehingga Tugas Akhir ini dapat terselesaikan dengan baik

Penulis menyadari bahwa program dan laporan Tugas Akhir ini masih jauh dari sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun dari pembaca sekalian. Sehingga suatu saat penulis dapat memberikan karya yang lebih baik lagi.

Akhir kata penulis ingin meminta maaf bila ada kesalahan baik dalam penyusunan laporan maupun yang pernah penulis lakukan sewaktu membuat program Tugas Akhir. Sekali lagi penulis memohon maaf yang sebesar-besarnya. Dan semoga dapat berguna bagi kita semua.

Yogyakarta, 30 Mei 2014

Penulis

@UKDW

INTISARI

Implementasi Algoritma Dijkstra untuk Peringkasan Otomatis Jurnal Ilmiah Berbahasa Inggris

Pada saat ini, sangatlah mudah untuk mendapatkan informasi, baik melalui media cetak maupun media elektronik. Akan tetapi, informasi yang tersedia bisa saja berupa teks yang sangat panjang dan belum tentu bersifat spesifik seperti yang dibutuhkan. Dengan membaca seluruh isi teks, manusia dapat menyimpulkan isi dari teks tersebut. Akan tetapi, keterbatasan waktu menjadi kendala utama karena untuk membaca teks yang panjang membutuhkan waktu yang cukup lama.

Berdasarkan masalah yang telah dipaparkan, aplikasi berbasis algoritma *shortest path* untuk meringkas teks dibangun. Dalam algoritma *shortest path* kalimat akan direpresentasikan sebagai vertex dan hubungan antara dua kalimat sebagai *edge*. Algoritma *shortest path* yang dipilih pada penelitian ini adalah algoritma *Dijkstra*. Hasil dari ringkasan dibangun dari jalur terpendek yang dilalui oleh algoritma *Dijkstra*. Proses peringkasan teks pada penelitian ini juga melibatkan *database wordnet*. *WordNet* merupakan sebuah *database* yang menyimpan sinonim kata berbahasa inggris. Dengan menggunakan *database WordNet* diharapkan dapat meningkatkan kualitas hasil ringkasan teks.

Sistem yang dibangun telah mampu menghasilkan ringkasan 90% relevan. Dari 20 dokumen uji, 18 diantaranya menghasilkan hasil ringkasan yang relevan. Sedangkan dengan menggunakan *database WordNet*, menghasilkan ringkasan 80% relevan. Dari 20 dokumen uji, 16 diantaranya menghasilkan hasil yang relevan.

Kata Kunci: Algoritma *Dijkstra*, *Text Summarization*, Document Teks, *WordNet*, *Shortest Path*.

DAFTAR ISI

PERNYATAAN KEASLIAN SKRIPSI.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PENGESAHAN	v
UCAPAN TERIMA KASIH.....	vi
INTISARI	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
BAB 1 PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	4
1.5. Metode Penelitian.....	4
1.6. Sistematika Penulisan.....	5
BAB 2 TINJAUAN PUSTAKA.....	7
2.1. Tinjauan Pustaka.....	7
2.2. Landasan Teori.....	8
2.2.1. Text Summarization.....	8
2.2.2. TF-IDF	9
2.2.3. TF-ISF.....	10
2.2.4. Graph.....	11
2.2.5. Shortest Path	11
2.2.6. Dijkstra Algorithm	12
2.2.7. F-Measure	14
BAB 3 ANALISIS DAN PERANCANGAN SISTEM.....	16
3.1. Kebutuhan Perangkat Keras dan Perangkat Lunak	16
3.2. Arsitektur Sistem.....	16

3.3.	<i>Use Case Diagram</i>	17
3.4.	Flowchart.....	18
3.4.1.	<i>Flowchart</i> Modul Pre Proses	19
3.4.2.	<i>Flowchart</i> Modul Peringkasan Teks.....	23
3.5.	Perancangan User Interface.....	24
3.6.	Perancangan Basis Data	26
3.6.1.	Schema Diagram.....	26
3.6.2.	Tabel Temp	26
3.6.3.	Tabel Document.....	27
3.6.4.	Tabel Relevan	27
3.7.	Perancangan Struktur Data.....	28
3.7.1.	Struktur Data <i>Union-Find</i>	28
3.7.2.	Struktur Data <i>Priority Queue</i>	30
3.8.	Perancangan Evaluasi Sistem.....	30
3.9.	Contoh Perhitungan Manual.....	31
BAB 4 IMPLEMENTASI DAN ANALISIS SISTEM.....		43
4.1.	Implementasi <i>User Interface</i>	43
4.1.1.	<i>Form</i> Utama.....	43
4.1.2.	<i>Form</i> Evaluasi.....	45
4.1.3.	<i>Form</i> Detail proses.....	46
4.2.	Implementasi Kode.....	48
4.2.1.	Proses Mencari kata yang Berhubungan pada Database WordNet.....	48
4.2.2.	Modul Tokenisasi.....	49
4.2.3.	Modul Pre-Proses.....	50
4.2.4.	Modul Peringkasan Teks.....	53
4.3.	Analisis Performa Sistem.....	55
4.3.1.	Evaluasi Wordnet.....	55
4.3.2.	Evaluasi Non- <i>WordNet</i>	57
4.3.3.	Membandingkan Wordnet dengan Non-Wordnet.....	58
4.4.	Analisis Sistem Secara Manual	61
BAB 5 KESIMPULAN DAN SARAN		64

5.1. Kesimpulan..... 64

5.2. Saran..... 65

@UKDW

DAFTAR TABEL

Tabel 3.1 Tabel <i>Temp</i>	27
Tabel 3.2 Tabel <i>Document</i>	27
Tabel 3.3 Tabel <i>Relevan</i>	28
Tabel 3.4 Tabel Contoh Kata	28
Tabel 3.5 Tabel Keterkaitan Kata	29
Tabel 3.6 Tabel Proses Penggabungan pada Struktur Data <i>Union-Find</i>	29
Tabel 3.7 Tabel Nilai Kemiripan Antar Kalimat	34
Tabel 3.8 Tabel Isi Struktur Data <i>Priority-Queue</i> pada Proses <i>Relax Edge Vertex 1</i>	35
Tabel 3.9 Tabel Isi Struktur Data <i>Priority-Queue</i> pada Proses <i>Relax Edge Vertex 2</i>	37
Tabel 3.10 Tabel Isi Struktur Data <i>Priority-Queue</i> pada Proses <i>Relax Edge Vertex 3</i>	38
Tabel 3.11 Tabel Isi Struktur Data <i>Priority-Queue</i> pada Proses <i>Relax Edge Vertex 4</i>	39
Tabel 3.12 Tabel Isi Struktur Data <i>Priority-Queue</i> pada Proses <i>Relax Edge Vertex 5</i>	40
Tabel 3.13 Tabel Isi Struktur Data <i>Priority-Queue</i> pada Proses <i>Relax Edge Vertex 6</i>	41
Tabel 3.14 Tabel Isi Struktur Data <i>Priority-Queue</i> pada Proses <i>Relax Edge Vertex 7</i>	42
Tabel 4.1 Tabel Daftar Dokumen Uji	55
Tabel 4.2 Tabel Nilai Performa Sistem dengan <i>WordNet</i>	56
Tabel 4.3 Tabel Nilai Performa Sistem Non- <i>WordNet</i>	57
Tabel 4.4 Tabel Perbandingan Nilai <i>F-Measure</i>	59
Tabel 4.5 Tabel Perbandingan Nilai Rata-Rata <i>F-Measure</i> , <i>Recall</i> , dan <i>Precision</i>	60
Tabel 4.6 Tabel Hasil Evaluasi Secara Manual	61

DAFTAR GAMBAR

Gambar 2.1 Contoh <i>Multiple Document Summarization</i>	9
Gambar 2.2 Contoh <i>Single Document Summarization</i>	9
Gambar 2.3 Contoh Representasi <i>Graph</i>	11
Gambar 2.4 Gambar Simulasi Algoritma <i>Dijkstra</i>	13
Gambar 2.5 Tabel Kontingensi	14
Gambar 3.1 Arsitektur Sistem Peringkasan Teks Otomatis.....	16
Gambar 3.2 <i>Use Case Diagram</i>	17
Gambar 3.3 <i>Flowchart</i> Sistem Secara Umum	18
Gambar 3.4 <i>Flowchart</i> Modul Pre-Proses	19
Gambar 3.5 <i>Flowchart</i> Modul Menghitung <i>Tf-Isf</i>	20
Gambar 3.6 <i>Flowchart</i> Modul Menghitung Nilai Kemiripan Antar Kalimat.....	21
Gambar 3.7 <i>Flowchart</i> Modul Pembangunan <i>Graph</i>	22
Gambar 3.8 <i>Flowchart</i> Modul Peringkasan Teks	23
Gambar 3.9 <i>User Interface</i> Saat Sistem Peringkasan Teks Kali Djalankan	24
Gambar 3.10 <i>User Interface</i> Detail Peringkasan Teks	25
Gambar 3.11 <i>User Interface</i> Evaluasi.....	25
Gambar 3.12 <i>Schema Diagram Database</i> Sistem Peringkasan Teks	26
Gambar 3.13 Gambar Proses <i>Relax Edge</i> pada <i>Vertex 1</i>	35
Gambar 3.14 Gambar Proses <i>Relax Edge</i> pada <i>Vertex 2</i>	36
Gambar 3.15 Gambar Proses <i>Relax Edge</i> pada <i>Vertex 3</i>	37
Gambar 3.16 Gambar Proses <i>Relax Edge</i> pada <i>Vertex 4</i>	38
Gambar 3.17 Gambar Proses <i>Relax Edge</i> pada <i>Vertex 5</i>	39
Gambar 3.18 Gambar Proses <i>Relax Edge</i> pada <i>Vertex 6</i>	40
Gambar 3.19 Gambar Proses <i>Relax Edge</i> pada <i>Vertex 7</i>	41
Gambar 4.1 <i>Form</i> Utama	43
Gambar 4.2 <i>Form</i> Utama Saat Memilih Tombol <i>File</i>	44
Gambar 4.3 <i>Form</i> Utama Saat Memilih Berkas Yang Akan Diringkas	44
Gambar 4.4 <i>Form</i> Utama Saat Pengguna Memilih Tombol Proses.....	45
Gambar 4.5 <i>Form</i> Evaluasi.....	46

Gambar 4.6 <i>Form</i> Detil Proses Peringkasan Tanpa <i>WordNet</i>	47
Gambar 4.7 <i>Form</i> Detil Proses Peringkasan Dengan <i>WordNet</i>	47
Gambar 4.8 <i>Procedure</i> Untuk Memasukkan Hasil <i>Query</i> ke Dalam Tabel <i>Temp</i>	48
Gambar 4.9 Implementasi Kode Pembuangan <i>Stopword</i>	49
Gambar 4.10 Implementasi Kode Tokenisasi Kalimat	50
Gambar 4.11 Implementasi Kode Modul Membandingkan Dengan <i>WordNet</i>	50
Gambar 4.12 Implementasi Kode Perhitungan <i>sf</i> dan <i>isf</i>	51
Gambar 4.13 Implementasi Kode Perhitungan <i>tf</i> dan <i>tf-isf</i>	51
Gambar 4.14 Implementasi Kode Perhitungan Nilai Kemiripan.....	52
Gambar 4.15 Implementasi Kode Pembangunan Graf	53
Gambar 4.16 Implementasi Kode Algoritma <i>Dijkstra</i>	54
Gambar 4.17 Implementasi Kode Mendapatkan Jalur Terpendek <i>Dijkstra</i>	54
Gambar 4.18 Implementasi Kode Menggabungkan Jalur Sesuai Jalur Terpendek	54
Gambar 4.19 Grafik Nilai Performa Sistem Dengan <i>WordNet</i>	57
Gambar 4.20 Grafik Nilai Performa Sistem <i>Non-WordNet</i>	58
Gambar 4.21 Grafik Perbandingan Nilai <i>Performance</i>	60

@UKDM

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Pada saat ini, sangatlah mudah untuk mendapatkan informasi, baik melalui media cetak maupun media elektronik. Akan tetapi, banyaknya informasi yang ada belum tentu bersifat spesifik seperti yang dibutuhkan. Keterbatasan waktu untuk membaca semua informasi yang ada juga menjadi salah satu kendala untuk memperoleh informasi utama yang diperlukan. Dengan perkembangan teknologi yang semakin maju, tidaklah mustahil untuk membuat sebuah sistem yang mampu mengatasi berbagai macam kendala yang dihadapi dalam memilah informasi. *Classification, categorization, clustering, dan summarization* dapat menjadi *alternative* untuk membuat sistem yang berguna untuk mendapatkan informasi yang lebih spesifik.

Text summarization adalah suatu metode peringkasan dokumen teks yang dapat menghasilkan teks baru yang berisi informasi penting dari dokumen teks asli. Peringkasan teks ini dapat dilakukan dengan *TF-IDF Method, Cluster Based Method, Graph Theoretic Approach, Latent Semantic Analysis Method, Machine Learning Approach*, dan beberapa metode lainnya.

Graph Theoretic Approach adalah salah satu metode untuk *text summarization* dengan cara memodelkan suatu dokumen teks menjadi sebuah *graph*. *Graph* memiliki *vertex* yang mewakili kalimat dan *edge* yang mewakili relasi antar kalimat. Setiap *vertex* (kalimat) akan diberi bobot yang berbeda-beda sesuai dengan bobot kalimatnya. *Edge* yang mewakili relasi antar kalimat juga diberi bobot sesuai dengan keterkaitan antar kalimat, bobot akan semakin kecil jika kedua kalimat memiliki kesamaan dan posisi kalimat berdekatan.

Setelah terbentuk *graph*, metode *shortest path* akan digunakan untuk menghasilkan teks baru. Beberapa metode *shortest path* yang dapat digunakan adalah *Dijkstra*, *A**, *Floyd-Warshall*, dan *Bellman-Ford*. Dalam riset ini metode *shortest path* yang digunakan adalah algoritma *Dijkstra*. Algoritma *Dijkstra* adalah algoritma untuk menemukan jalur terpendek dari suatu posisi awal ke posisi tujuan. Jika diimplementasikan pada *text summarization*, algoritma ini akan mencari jalur terpendek dimulai dari kalimat pertama (mula-mula) hingga kalimat terakhir (tujuan).

Sjöbergh dan Araki (2006) menguraikan tentang metode *text summarization* menggunakan algoritma *shortest path*, namun mereka belum mempertimbangkan hubungan semantik antara dua kalimat. Untuk menemukan hubungan semantik antara dua kalimat dapat digunakan *WordNet*. *WordNet* disini digunakan untuk mencari keterkaitan antara kalimat sehingga kata yang memiliki arti sama akan disatukan.

1.2. Rumusan Masalah

Dalam rangka membangun sistem peringkasan teks dengan mengimplementasikan algoritma *dijkstra*, beberapa hal berikut ini perlu dipertimbangkan:

1. Bagaimana cara merepresentasikan teks dalam bentuk *graph* ?
2. Fungsi heuristik apa yang harus digunakan untuk menentukan bobot setiap *edge* ?
3. Apakah ada pengaruh terhadap kualitas semantik pada hasil ringkasan teks ketika menggunakan *WordNet*?

1.3. Batasan Masalah

Penelitian yang akan dilakukan memiliki beberapa batasan sebagai berikut ini :

1. Sistem akan dibuat dalam bentuk aplikasi *desktop*, karena penelitian ini belum mempertimbangkan faktor aksesibilitas terhadap hasil peringkasan teks sistem.
2. Dokumen yang digunakan berupa *plain text (ASCII)*. Menggunakan *plain text* karena tidak memiliki metadata seperti tipe teks lainnya, sehingga proses peringkasan teks dapat langsung dilakukan tanpa harus melakukan pre-proses untuk membuat metadata yang tidak digunakan dalam sistem yang dibangun.
3. Dokumen yang digunakan merupakan artikel berbahasa Inggris.
4. Pre-proses terhadap teks untuk membersihkan dokumen dari karakter-karakter selain huruf dan tanda baca.
5. Dokumen uji yang digunakan merupakan sekumpulan artikel yang didapat dari http://gulfuniversity.edu.bh/en/library/download_journals3.html. Dokumen uji yang digunakan berjumlah 20 artikel. Artikel yang akan digunakan adalah artikel yang memiliki abstrak sesuai dengan kriteria yaitu mengandung penjelasan permasalahan, cara menyelesaikan masalah, dan hasil penelitian.
6. Sistem yang dibuat melakukan peringkasan pada dokumen teks tunggal

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengetahui efektivitas dari implementasi algoritma *Dijkstra* dalam menyusun ringkasan teks secara otomatis. Penggunaan *WordNet* disini bertujuan meningkatkan kualitas hasil peringkasan teks.

1.5. Metode Penelitian

Metode yang digunakan dalam penelitian ini antara lain sebagai berikut :

1. Pengumpulan Data

Artikel berbahasa inggris yang akan digunakan diambil dari http://gulfuniversity.edu.bh/wp/library/download_journals3.html.

Artikel yang diambil adalah artikel berekstensi pdf lalu dikonversi secara manual menjadi bentuk *plain text*.

2. Pembuatan Sistem

Sistem akan dikembangkan menggunakan algoritma *Dijkstra* untuk mencari *shortest path* dari *graph* yang dibangun. Dalam *graph* yang dibangun *vertex* dianggap kalimat dan *edge* dianggap sebagai keterkaitan antar kalimat.

3. Evaluasi

Evaluasi akan dilakukan dengan beberapa cara sebagai berikut :

- **Evaluasi Secara Manual**

Evaluasi secara manual dilakukan dengan cara membandingkan hasil peringkasan teks dari sistem dengan abstrak artikel dokumen uji. Hasil peringkasan teks dari sistem dikatakan

akurat jika ringkasan teks mengandung tiga komponen berikut ini:

1. Penjelasan masalah yang diteliti
2. Cara atau metode yang digunakan untuk menyelesaikan masalah
3. Hasil dari penelitian

Dan isinya sesuai dengan isi dari tiga komponen tersebut dalam dokumen uji.

- **Menghitung Performa Sistem**

Untuk mengevaluasi performa system, ada 3 hal yang akan dihitung yaitu *precision*, *recall*, dan *f-measure*. *Relevant document* yang digunakan untuk menghitung performa sistem adalah abstrak yang terdapat pada setiap artikel dalam dokumen uji.

1.6. Sistematika Penulisan

Untuk memudahkan dalam mendapatkan gambaran yang lengkap dan jelas mengenai penelitian yang akan dilakukan, penulis membagi laporan ini menjadi 5 (lima) bab yaitu Bab I Pendahuluan, Bab II Tinjauan Pustaka, Bab III Analisis dan Perancangan Sistem, Bab IV Implementasi dan Analisis Sistem, dan Bab V Kesimpulan dan Saran.

Dalam bab 1 menguraikan hal-hal seperti latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, metode/pendekatan yang digunakan serta sistematika penulisan laporan Tugas Akhir.

Bab 2 berisi tentang tinjauan pustaka serta landasan teori yang diperlukan untuk memecahkan masalah dalam riset yang dilakukan.

Bab 3 berisi tentang analisis teori yang digunakan dalam penelitian, uraian tentang variabel dan data yang akan dikumpulkan, flowchart dan arsitektur sistem, cara perancangan dan simulasi yang dilakukan .

Bab 4 berisi tentang hasil penelitian/implementasi serta pembahasan/analisis dari penelitian yang telah dilakukan dan dijelaskan secara terpadu.

Bab 5 berisi kesimpulan dari sistem yang telah dibuat dan saran yang akan berguna untuk pengembangan system selanjutnya. Dengan adanya saran, diharapkan riset yang dilakukan selanjutnya akan menghasilkan hasil yang lebih baik.

@UKDWN

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil implementasi dan analisis sistem, maka diperoleh kesimpulan sebagai berikut:

- Dalam analisis secara manual, dapat disimpulkan bahwa sistem peringkasan teks tanpa menggunakan *database WordNet* menghasilkan ringkasan yang lebih baik. Pada peringkasan tanpa menggunakan *database WordNet*, dari 20 dokumen uji, 18 diantaranya menghasilkan hasil yang relevan dengan abstrak. Sedangkan dengan menggunakan *database WordNet*, hanya 16 dokumen yang relevan dengan abstrak.
- Dalam analisis performa sistem, peringkasan dengan menggunakan *WordNet* memiliki nilai rata-rata *f-measure*, *precision*, dan *recall* yaitu 0.310953, 0.228385, 0.561919. Sedangkan peringkasan teks tanpa menggunakan *database WordNet* memiliki nilai rata-rata *f-measure*, *precision*, dan *recall* sebesar 0.331726, 0.263364, 0.487523. Dari data tersebut, dapat disimpulkan performa sistem lebih baik saat tidak menggunakan *database WordNet*.

5.2. Saran

Saran untuk pengembangan dan perbaikan sistem adalah:

- Mengoptimalkan cara mengambil kata-kata yang berhubungan dalam *database WordNet*. Pada penelitian ini, dilakukan pre-proses untuk mengambil kata yang berhubungan dan memakan waktu kurang lebih 36 jam.
- Dalam penelitian ini, untuk mendapatkan hubungan antar kalimat hanya memanfaatkan 3 dari 18 tabel yang terdapat pada *database WordNet* yaitu tabel *words*, *synsets*, dan *senses*. Pada penelitian selanjutnya, dapat mencoba untuk memanfaatkan semua tabel *WordNet*.
- Menggunakan formula yang lebih optimal untuk memanfaatkan *database wordnet*.
- Dapat mencoba melakukan *stemming* dan *lemmatization* terlebih dahulu sebelum dilakukan peringkasan teks.

DAFTAR PUSTAKA

- Budhi, G. S., Intan, R., R, S., & R, S. R. (2007). Indonesian Automated Text Summarization. Proceeding ICSIT, (pp. 26-27).*
- Douglas, W. B. (2001). Introduction to Graph Theory Second Edition. United States of America: Prentice-Hall.*
- Halim, S., & Halim, F. (2013). Competitive Programming 3. Singapore.*
- Sjöbergh, J., & Araki, K. (2006). Extraction based summarization using a shortest path algorithm. In Proceedings of 12th Annual Language Processing Conference (NLP 2006).*
- Manning, C. D., Raghavan, P., & Schitze, H. (2009). An Introduction to Information Retrieval. England: Cambridge University Press.*
- Ruohonen, K. (2013). Graph Theory. Tampere University of Technology.*
- Sankar K, Sobha L. (2009). An Approach to Text Summarization. Association for Computational Linguistics , 53-60.*
- Weiss, S. M., Indurkha, N., & Zhang, T. (2009). Text Mining Predictive Methods for Analyzing Unstructured Information. Springer.*