

S2.699 / H / 13



ITS
Institut
Teknologi
Sepuluh Nopember



RTf

006-312

A26

0-1

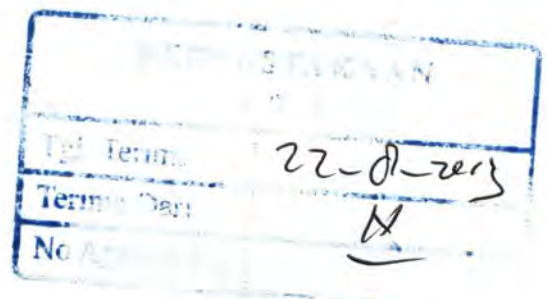
2013

TESIS - KI092361

OTOMATISASI PEMBERIAN REKOMENDASI PRODUK BERDASARKAN BOBOT FITUR DAN BOBOT KALIMAT COMPARATIVE PADA TEKS OPINI

YUFIS AZHAR
5111201705

DOSEN PEMBIMBING:
Dr. Agus Zainal Arifin, S.Kom, M.Kom
Diana Purwitasari, S.Kom, M.Sc



PROGRAM MAGISTER
BIDANG KEAHLIAN KOMPUTASI CERDAS DAN VISUALISASI
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2013

Handwritten notes in a box at the top left, possibly containing a name and a date.

Handwritten notes on the left side, including the number "2005" and a symbol resembling a stylized "M" or "W".



Handwritten notes at the bottom center of the page.

OTOMATISASI PEMBERIAN REKOMENDASI PRODUK BERDASARKAN BOBOT FITUR DAN BOBOT KALIMAT COMPARATIVE PADA TEKS OPINI

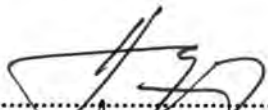
Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom)
di
Institut Teknologi Sepuluh Nopember

oleh :
Yufis Azhar
5111201705

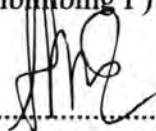
Tanggal Ujian : 24 Juli 2013
Periode Wisuda : September 2013

Disetujui oleh :

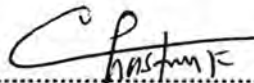
1. Dr. Agus Zainal Arifin, S.Kom, M.Kom
NIP. 19720809 199512 1 001


.....
(Pembimbing I)


2. Diana Purwitasari, S.Kom, M.Sc
NIP. 19780410 200312 2 001


.....
(Pembimbing II)

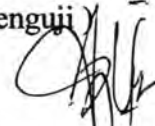
3. Dr. Chastine Fatichah, M.Kom.
NIP. 19751220 200112 2 002


.....
(Penguji)

4. Isye Arieshanti, S.Kom, M.Phil.
NIP. 19780412 200604 2 001


.....
(Penguji)

5. Arya Yudhi Wijaya, S.Kom, M.Kom.
NIP. 19840904 201012 1 002


.....
(Penguji)

Direktur Program Pascasarjana


Prof. Dr. Ir. Adi Soeprijanto, MT
NIP. 19640405 199002 1001

[Halaman ini sengaja dikosongkan]



OTOMATISASI PEMBERIAN REKOMENDASI PRODUK BERDASARKAN BOBOT FITUR DAN BOBOT KALIMAT COMPARATIVE PADA TEKS OPINI

Nama mahasiswa : Yufis Azhar
NRP : 5111201705
Pembimbing : Dr. Agus Zainal Arifin, S.Kom., M.Kom.
Diana Purwitasari, S.Kom., M. Sc.

ABSTRAK

Seorang calon pembeli cenderung lebih suka membandingkan kelebihan dan kekurangan antara sebuah produk dengan produk yang lain sebelum melakukan pembelian. Biasanya, kegiatan ini dilakukan dengan cara mengamati komentar-komentar yang diberikan oleh pembeli lain yang sudah pernah membeli atau menggunakan produk tersebut. Proses ini dapat memakan waktu yang lama mengingat jumlah komentar yang dimiliki oleh suatu produk sangatlah besar.

Proses otomatisasi perbandingan produk berdasarkan teks opini dapat dilakukan dengan cara mengekstrak fitur yang dimiliki oleh produk tersebut. Fitur-fitur inilah yang umumnya menjadi target suatu opini. Akan tetapi, menilai suatu produk lebih baik dari pada produk yang lain hanya dengan melihat fitur-fitur yang dimilikinya tidaklah adil. Karena ini merupakan suatu mekanisme penilaian yang independen. Padahal untuk melakukan suatu perbandingan, dua buah entitas tidak boleh dinilai secara terpisah, melainkan harus dinilai secara bersamaan.

Oleh karena itu, dalam penelitian ini diusulkan suatu metode pembobotan berbasis kalimat *comparative* dan fitur untuk menentukan nilai rekomendasi suatu produk. Dengan penggunaan kalimat *comparative* ini, diharapkan masalah independensi dalam perbandingan produk dapat teratasi. Caranya adalah dengan terlebih dahulu mengekstrak fitur yang dimiliki oleh suatu produk untuk mendapatkan bobotnya. Kemudian cari kalimat *comparative* yang berisi kedua produk yang ingin dibandingkan untuk mendapatkan bobot *comparativenya*. Selanjutnya, dua buah bobot yang telah didapatkan tersebut digabungkan untuk mendapatkan nilai rekomendasi suatu produk. Nilai inilah yang digunakan untuk membandingkan suatu produk dengan produk yang lain. Hasil yang didapatkan menunjukkan bahwa penambahan bobot *comparative* terbukti dapat meningkatkan akurasi dalam pengenalan produk yang lebih unggul hingga 96%.

Kata kunci: Teks Opini, Feature-based Opinion Mining, Kalimat *Comparative*, Perbandingan Produk, Nilai Rekomendasi Produk

OTOMATISASI PERUBAHAN KELOMPOK BAWA
PERUBAHAN ANGGARAN PERUBAHAN BAWA
COMPARATIVE PAPER TYPE OPINE

[Halaman ini sengaja dikosongkan]

ABSTRAK

Penelitian ini bertujuan untuk mengetahui pengaruh dari perubahan struktur organisasi terhadap kinerja organisasi. Penelitian ini menggunakan metode kuantitatif dengan menggunakan data sekunder yang diperoleh dari laporan keuangan perusahaan yang terdaftar di Bursa Efek Indonesia (BEI) selama periode 2010-2015. Hasil penelitian menunjukkan bahwa perubahan struktur organisasi berpengaruh signifikan terhadap kinerja organisasi. Perubahan struktur organisasi yang mengarah ke struktur yang lebih sederhana dan lebih efisien akan meningkatkan kinerja organisasi. Sebaliknya, perubahan struktur organisasi yang mengarah ke struktur yang lebih kompleks dan lebih birokratis akan menurunkan kinerja organisasi.

Penelitian ini bertujuan untuk mengetahui pengaruh dari perubahan struktur organisasi terhadap kinerja organisasi. Penelitian ini menggunakan metode kuantitatif dengan menggunakan data sekunder yang diperoleh dari laporan keuangan perusahaan yang terdaftar di Bursa Efek Indonesia (BEI) selama periode 2010-2015. Hasil penelitian menunjukkan bahwa perubahan struktur organisasi berpengaruh signifikan terhadap kinerja organisasi. Perubahan struktur organisasi yang mengarah ke struktur yang lebih sederhana dan lebih efisien akan meningkatkan kinerja organisasi. Sebaliknya, perubahan struktur organisasi yang mengarah ke struktur yang lebih kompleks dan lebih birokratis akan menurunkan kinerja organisasi.

Penelitian ini bertujuan untuk mengetahui pengaruh dari perubahan struktur organisasi terhadap kinerja organisasi. Penelitian ini menggunakan metode kuantitatif dengan menggunakan data sekunder yang diperoleh dari laporan keuangan perusahaan yang terdaftar di Bursa Efek Indonesia (BEI) selama periode 2010-2015. Hasil penelitian menunjukkan bahwa perubahan struktur organisasi berpengaruh signifikan terhadap kinerja organisasi. Perubahan struktur organisasi yang mengarah ke struktur yang lebih sederhana dan lebih efisien akan meningkatkan kinerja organisasi. Sebaliknya, perubahan struktur organisasi yang mengarah ke struktur yang lebih kompleks dan lebih birokratis akan menurunkan kinerja organisasi.

Kata kunci: Perubahan struktur organisasi, Kinerja organisasi, Bursa Efek Indonesia (BEI), Laporan keuangan.

PRODUCT RECOMENDATIONS BASED ON FEATURES AND COMPARATIVE WEIGHT IN OPINIONATED TEXT

Student Name : Yufis Azhar
Student Identity Number : 5111201705
Supervisors : Dr. Agus Zainal Arifin, S.Kom., M.Kom.
Diana Purwitasari, S.Kom., M. Sc.

ABSTRACT

A customer tends to be more likely to compare the advantages and disadvantages between products before making a purchase. Typically, this activity is done by observing the comments or reviews given by other customers who already have purchased or used the product. This process can take a long time considering the number of reviews can be in hundreds or even thousands for a popular product.

Product comparisons based on subjective text can be done by extracting product features. These features are commonly the target of an opinion. However, assessing a product is better than another product just by looking at its features is not fair. To make a comparison, the two entities should not be assessed independent, but must be assessed together

Therefore, this research proposes a weighting method based on features product and comparative sentence to determine a value of product recommendation. The use of comparative sentences is expected to solve the independence issue in the products comparison. First, we extract the features product to gain its weight. After that, we find comparative sentences that contain both of the products to be compared for determine the preferred product. From that sentence, we will extract the comparative weight. Furthermore, two weights that have been obtained before are combined to get a recommendation value of a product. This value is used to compare one product with another product. The result shows that the use of comparative weight can improve accuracy until 96%.

Keywords: Opinionated Text, Feature-based Opinion Mining, Comparative Sentence, Products Comparison.

[Halaman ini sengaja dikosongkan]

ABSTRACT

The purpose of this paper is to present a new approach to the design of expert systems. The approach is based on the use of fuzzy logic and neural networks. The design process is divided into two main phases: the design of the knowledge base and the design of the inference engine. The knowledge base is designed using fuzzy logic, and the inference engine is designed using neural networks. The resulting expert system is able to handle uncertain and incomplete information, and is able to learn from its experience.

The design process is divided into two main phases: the design of the knowledge base and the design of the inference engine. The knowledge base is designed using fuzzy logic, and the inference engine is designed using neural networks. The resulting expert system is able to handle uncertain and incomplete information, and is able to learn from its experience.

The design process is divided into two main phases: the design of the knowledge base and the design of the inference engine. The knowledge base is designed using fuzzy logic, and the inference engine is designed using neural networks. The resulting expert system is able to handle uncertain and incomplete information, and is able to learn from its experience.

The design process is divided into two main phases: the design of the knowledge base and the design of the inference engine. The knowledge base is designed using fuzzy logic, and the inference engine is designed using neural networks. The resulting expert system is able to handle uncertain and incomplete information, and is able to learn from its experience.

Keywords: Expert systems, Fuzzy logic, Neural networks, Knowledge engineering, Artificial intelligence.

KATA PENGANTAR

Puji syukur kepada Allah SWT atas segala berkah dan anugerah-Nya sehingga penulis dapat menyelesaikan tesis yang berjudul “Otomatisasi Pemberian Rekomendasi Produk Berdasarkan Bobot Fitur dan Bobot Kalimat Comparative Pada Teks Opini”. Tesis ini disusun untuk memenuhi sebagian persyaratan untuk memperoleh gelar Magister Komputer di Institut Teknologi Sepuluh Nopember Surabaya.

Tesis ini dapat terselesaikan tidak lepas dari bantuan dan dorongan yang sangat berharga dari berbagai pihak. Oleh sebab itu penulis mengucapkan terimakasih dan penghargaan yang sebesar-besarnya kepada berbagai pihak antara lain :

1. Bapak Dr. Agus Zainal Arifin, S.Kom, M.Kom sebagai pembimbing I yang telah banyak meluangkan waktu dan pikiran dalam membimbing penulis untuk menyelesaikan tesis ini
2. Ibu Diana Purwitasari, S.Kom, M.Sc selaku pembimbing II yang telah membantu dalam menyempurnakan tesis ini
3. Bapak Waskitho Wibisono, S.Kom, M.Eng, Ph.D selaku Ketua Program Pascasarjana Teknik Informatika beserta staff yang telah memberikan fasilitas akademik
4. Kedua orang tuaku, Abdul Wahab dan Nur Hidayati yang tidak henti-hentinya memberi kasih sayang, dukungan serta doa kepada setiap anaknya
5. Calon istriku, Yeni Suswatiningsih, yang telah memberikan cinta, semangat, perhatian dan hiburan dalam setiap langkahku
6. Saudara-saudaraku, Mbak Fitri, Mbak Khusnah, Mas Alwi, Mas Rizki serta keluarga besarku yang selalu menjadi pelecut semangat dalam setiap aktivitasku
7. Institusiku, Keluarga Besar Jurusan Teknik Informatika Universitas Muhammadiyah Malang atas kesempatan yang diberikan dalam menempuh studi program magister ini

8. Sahabatku, Septiyan Andika Isanta, yang dengan sabar menjawab berbagai pertanyaan bodoh penulis tentang bahasa pemrograman java dan segala keruwetannya
 9. Teman-teman seperjuangan, Hendra, Wahyu, Maskur, Adit, Nasser, Prima, Randy, Ali dan seluruh teman-teman S2 Teknik Informatika ITS yang tidak mungkin dapat penulis sebut satu-persatu, atas keakraban, kekonyolan dan suntikan semangat dalam menjalankan aktivitas selama masa perkuliahan
 10. Semua pihak yang telah membantu penulis dalam menyelesaikan tesis ini
- Penulis menyadari bahwa dalam laporan tesis ini masih banyak kekurangan, oleh karena itu masukan dan saran yang membangun demi perbaikan dan pengembangan tesis ini sangat penulis harapkan. Akhir kata semoga tesis ini bermanfaat bagi para pembaca dalam pengembangan ilmu pengetahuan di negeri ini.

Surabaya, 30 Juli 2013

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	i
ABSTRAK	iii
ABSTRACT	v
KATA PENGANTAR.....	vii
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian dan Manfaat Penelitian	4
1.5 Kontribusi Penelitian.....	4
BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI.....	5
2.1 <i>POS Tagging</i>	5
2.2 <i>Dependencies Parsing</i>	6
2.3 <i>Opinion Mining</i>	7
2.4 <i>Feature-based Opinion Mining</i>	8
2.5 Ekstraksi Fitur Produk Menggunakan Metode Double Propagation	10
2.5.1 Identifikasi Relasi.....	10
2.5.2 Penyusunan <i>Opinion Lexicon</i> dan Ekstraksi Target Opini.....	12
2.5.2.1 Aturan Propagasi Berbasis Relasi	13
2.5.2.2 Algoritma Propagasi.....	15
2.5.2.3 Penentuan Polaritas Kata Opini.....	17
2.6 Analisa Opini dalam Kalimat Comparative	18
2.7 Analisa Opini untuk Perbandingan Produk.....	20
BAB 3 METODA PENELITIAN	23
3.1 Pengumpulan dan Analisis Data	23
3.2 Desain Sistem.....	23
3.2.1 <i>Preprocessing</i>	25
3.2.2 Ekstraksi dan Pembobotan Fitur.....	26
3.2.3 Identifikasi dan Pembobotan Kalimat <i>Comparative</i>	36
3.2.4 Pengenalan Fitur Sejenis	41
3.2.5 Pemberian Nilai Rekomendasi Suatu Produk	41
3.3 Implementasi.....	44
3.4 Skenario Uji coba.....	44
BAB 4 HASIL DAN PEMBAHASAN.....	45

4.1 Pengumpulan Dataset	45
4.2 Implementasi Sistem.....	45
4.2.1 Preprocessing	45
4.2.2 Ekstraksi dan Pembobotan Fitur Produk	46
4.2.2.1 Pendefinisian Rule.....	48
4.2.2.2 Pruning Fitur	53
4.2.3 Pengenalan dan Pembobotan Kalimat Comparative.....	54
4.2.4 Pengenalan Fitur Sejenis.....	56
4.2.5 Pemberian Nilai Rekomendasi Untuk Tiap Produk.....	57
4.3 Hasil Uji Coba	58
4.3.1 Uji Metode MDP	58
4.3.1 Uji Signifikansi Skor CSF	59
4.4 Pembahasan	66
BAB 5 KESIMPULAN DAN SARAN	71
5.1 Kesimpulan.....	71
5.2 Saran	71
DAFTAR PUSTAKA.....	73
LAMPIRAN	75
BIOGRAFI PENULIS	91

DAFTAR GAMBAR

Gambar 2.1 Relasi Ketergantungan Antar Data.....	11
Gambar 2.2 Algoritma Propagasi.....	16
Gambar 3.1 Contoh Review Dalam Situs Amazon.com.....	23
Gambar 3.2 Tahapan Alur Kerja Sistem.....	24
Gambar 3.3 Diagram Proses Perbandingan Produk.....	24
Gambar 3.4 <i>Pseudocode</i> Penyaringan Relasi-relasi Penting.....	30
Gambar 3.5 Ilustrasi Fitur Produk yang Dianggap Beririsan.....	41
Gambar 4.1 Potongan Kode Pendefinisian Kamus Kata.....	47
Gambar 4.2 Potongan Kode Pembacaan Dokumen <i>Tagged</i>	47
Gambar 4.3 Potongan Kode Pembacaan Dokumen <i>Parsed</i>	48
Gambar 4.4 Potongan Kode Implementasi R1.....	49
Gambar 4.5 Potongan Kode Implementasi R2 dan R6.....	50
Gambar 4.6 Potongan Kode Implementasi R3.....	51
Gambar 4.7 Potongan Kode Implementasi R4.....	52
Gambar 4.8 Potongan Kode Implementasi R5.....	53
Gambar 4.9 Potongan Kode Implementasi Kalimat <i>Comparative</i>	55
Gambar 4.10 Potongan Kode Penanganan Kalimat <i>Comparative</i>	55
Gambar 4.11 Potongan Kode Implementasi Kalimat <i>Superlative</i>	56
Gambar 4.12 Potongan Kode Penanganan Kalimat <i>Superlative</i>	56
Gambar 4.13 Potongan Kode Penanganan Fitur Sejenis.....	57

DAFTAR TABEL

Tabel 2.1	Aturan Relasi dalam <i>Double Propagation</i>	15
Tabel 3.1	Tabel Aturan Ekstraksi Kata Opini dan Fitur Produk dalam MDP	27
Tabel 3.2	Tabel Pembobotan Fitur Produk dan Kata Opini dalam MDP.....	28
Tabel 3.3	Kamus Kata Opini (<i>Opinion Lexicon</i>)	30
Tabel 3.4	Daftar Fitur Produk Setelah Kalimat A1 Dievaluasi.....	31
Tabel 3.5	Daftar Kata Opini Setelah Kalimat A2 Dievaluasi	32
Tabel 3.6	Daftar Fitur Produk Setelah Kalimat A2 Dievaluasi.....	32
Tabel 3.7	Daftar Kata Opini Setelah Kalimat A3 Dievaluasi	32
Tabel 3.8	Daftar Fitur Produk Setelah Kalimat A3 Dievaluasi.....	33
Tabel 3.9	Daftar Fitur Produk Setelah Kalimat A4 Dievaluasi.....	33
Tabel 3.10	Daftar Fitur Produk Setelah Review A Dievaluasi	34
Tabel 3.11	Daftar Fitur Produk Setelah Kalimat B1 Dievaluasi.....	34
Tabel 3.12	Daftar Fitur Produk Setelah Review B Dievaluasi	35
Tabel 3.13	Daftar Fitur Produk Setelah Kalimat C1 Dievaluasi.....	35
Tabel 3.14	Daftar Kata Opini Setelah Kalimat C2 Dievaluasi	36
Tabel 3.15	Daftar Fitur Produk Setelah Review C Dievaluasi	36
Tabel 3.16	Contoh Daftar Kata Opini Yang Diekstrak Oleh MDP	38
Tabel 3.17	Contoh Daftar Fitur Produk Yang Diekstrak Oleh MDP.....	38
Tabel 3.18	Pasangan Lexicon-Fitur Untuk Tiap Kalimat Comparative.....	38
Tabel 3.19	Nilai CSF Hasil Pengamatan Kalimat Comparataive	39
Tabel 3.20	Nilai CSF Untuk Tiap Fitur Produk	39
Tabel 3.21	Pasangan Lexicon-Fitur Pada Kalimat Superlative.....	40
Tabel 3.22	Nilai CSF Hasil Pengamatan Kalimat Superlative.....	40
Tabel 3.23	Nilai CSF Akhir Untuk Tiap Fitur Produk.....	40
Tabel 3.24	Contoh Bobot dan Skor Tiap Fitur Produk	43
Tabel 4.1	Dokumen Review yang Digunakan Sebagai Dataset.....	45
Tabel 4.2	Tabel Uji Coverage Untuk Tiap Rule MDP.....	58
Tabel 4.3	Tabel Uji Akurasi Untuk Tiap Rule MDP	58
Tabel 4.4	Perbandingan Metode MDP dan DP	59
Tabel 4.5	Nilai Rate Untuk Tiap Produk Smartphone	60

Tabel 4.6	Nilai Rate Untuk Tiap Produk Camera Digital	60
Tabel 4.7	Nilai Rate Untuk Tiap Produk Televisi	60
Tabel 4.8	Tabel Akurasi Untuk Domain Smartphone (withKamus)	62
Tabel 4.9	Tabel Akurasi Untuk Domain Smartphone (withoutKamus)	62
Tabel 4.10	Akurasi Untuk Domain Smartphone (withKamus) Pasca Analisa	63
Tabel 4.11	Tabel Akurasi Untuk Domain Camera Digital (withKamus)	64
Tabel 4.12	Tabel Akurasi Untuk Domain Camera Digital (withoutKamus)	64
Tabel 4.13	Tabel Akurasi Untuk Domain Televisi (withKamus)	64
Tabel 4.14	Tabel Akurasi Untuk Domain Televisi (withoutKamus)	65
Tabel 4.15	Tabel Perbandingan Untuk Skor GSF dan GSF+CSF	65

DAFTAR LAMPIRAN

LAMPIRAN 1	Lembar Kuisisioner Penentuan Fitur Relevan	75
LAMPIRAN 2	Daftar Kamus Kata Fitur Relevan	79
LAMPIRAN 3	Antarmuka Aplikasi	83
LAMPIRAN 4	Hasil Uji-t Berpasangan Dua Sisi.....	85

[Halaman ini sengaja dikosongkan]



BAB 1 PENDAHULUAN

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dengan pesatnya perkembangan *e-commerce*, sebagian besar konsumen mengekspresikan opini mereka pada bermacam-macam entitas, seperti barang ataupun jasa. Ketersediaan teks-teks opini tersebut tidak hanya menyediakan informasi tambahan bagi konsumen yang lain, akan tetapi juga bermanfaat bagi para produsen untuk mengetahui tingkat kepuasan konsumen terhadap produk atau jasa yang mereka tawarkan. Ketersediaan teks opini dengan jumlah besar ini menarik perhatian beberapa orang peneliti untuk menggali informasi yang terkandung di dalamnya. Bidang ini selanjutnya lebih dikenal dengan istilah *opinion mining*.

Dalam *opinion mining*, salah satu masalah yang paling mendasar adalah pengekstrakan target opini. Pada suatu teks opini, target ini biasanya berupa frase/kata benda (Hu, 2004). Sebagai contoh dalam kalimat “*Canon S100 has a great lens*”, target opininya adalah “*lens*”. Dalam teks-teks opini tentang suatu produk yang tersebar di internet, target opini ini seringkali ditujukan bukan hanya untuk produk, tetapi juga untuk fitur (komponen) dari produk tersebut. Oleh karena itu, bidang ini seringkali disebut *feature based opinion mining*.

Permasalahan bagaimana mendapatkan fitur produk dari suatu teks opini cukup kompleks. Hal ini dikarenakan tidak semua kata benda yang terdapat dalam suatu teks opini adalah fitur produk. Harus ada suatu *identifier* yang dapat mengenali kata benda yang merupakan fitur produk. Hu menggunakan metode *supervised* untuk mengekstrak fitur produk dari suatu teks opini (Hu, 2004). Semua kata benda yang sering muncul dalam dokumen akan diekstrak menggunakan *association rule* dengan cara melihat hubungan antara kata tersebut dengan kata sifat dalam suatu kalimat. Kata sifat digunakan sebagai *identifier* dikarenakan kata ini seringkali digunakan oleh seseorang untuk mengekspresikan opininya. Kata benda yang berhasil terekstrak inilah yang kemudian dianggap sebagai fitur produk.

Dari penelitian tersebut, dapat disimpulkan bahwa untuk mengekstrak fitur produk, kata sifat digunakan sebagai salah satu indikator yang cukup kuat. Semakin dekat/kuat hubungan antara sebuah kata benda dengan suatu kata sifat, maka semakin besar peluang kata benda tersebut untuk diekstrak sebagai fitur produk. Karena pentingnya identifikasi kata sifat inilah, banyak peneliti yang memilih untuk menggunakan kamus kata sifat (*opinion lexicon*). Dengan memanfaatkan kamus ini, proses ekstraksi fitur produk menjadi lebih sederhana. Akan tetapi, penggunaan kamus kata tersebut tidak sepenuhnya efektif. Dikarenakan metode ini sangat bergantung pada kelengkapan kamus kata yang disediakan.

Untuk menangani permasalahan tersebut, Qiu mengusulkan algoritma *double propagation* yang merupakan metode *semi unsupervised* untuk mengekstrak fitur produk (Qiu, 2010). Metode ini tidak membutuhkan kamus kata sifat secara lengkap, karena metode ini dapat melengkapi kamus katanya secara otomatis. Caranya adalah dengan menemukan fitur menggunakan kamus kata sifat, kemudian memanfaatkan fitur yang telah terekstrak tadi untuk menemukan kata sifat lain yang terdapat dalam teks opini. Kata sifat baru tersebut secara otomatis akan ditambahkan dalam kamus kata. Proses ini berlangsung terus-menerus hingga tidak ada fitur dan kata sifat baru yang ditemukan.

Pengekstrakan fitur produk ini memunculkan ide dari sebagian peneliti untuk melakukan perbandingan produk secara otomatis dengan memanfaatkan teks opini. Hal ini didasarkan pada pengamatan bahwa seorang calon pembeli cenderung lebih suka membandingkan kelebihan dan kekurangan antara sebuah produk dengan produk lain sebelum melakukan pembelian. Liu membandingkan dua buah produk dengan mengamati tiap fitur dari kedua produk tersebut (Liu, 2005). Proses perbandingan ini diawali dengan mengekstrak fitur dari setiap produk yang akan dibandingkan. Kemudian, fitur yang telah terekstrak akan diberikan bobot berdasarkan jumlah komentar positif dan negatif untuk fitur tersebut. Bobot fitur tiap produk inilah yang dijadikan acuan untuk melakukan perbandingan.

Akan tetapi, sebagian besar penelitian tentang perbandingan produk dalam *opinion mining* hanya menilai kualitas produk dengan mengamatinya secara

sendiri-sendiri. Padahal, untuk melakukan suatu perbandingan, dua produk juga harus diamati secara bersamaan agar komentar yang diberikan lebih obyektif. Dalam suatu teks opini, seseorang seringkali mengungkapkan keunggulan suatu produk dibanding dengan produk yang lain menggunakan kalimat *comparative*, seperti pada kalimat “*Nokia 920 has a greater camera than iphone 5*”. Dalam kalimat tersebut, seorang komentator menyatakan bahwa fitur “*camera*” dari “*Nokia 920*” lebih baik daripada “*iphone 5*”. Penanganan kalimat *comparative* akan memberikan pengaruh yang cukup signifikan dalam proses *opinion mining*. Hal ini dikarenakan jumlah kalimat jenis ini sekitar 10% dari total jumlah opini terhadap suatu produk (Ganapathibhotla, 2008). Sejauh yang diketahui, kalimat *comparative* ini belum dimanfaatkan secara optimal oleh para peneliti untuk membandingkan suatu produk dengan produk yang lain.

Berdasarkan permasalahan tersebut, dalam penelitian ini diusulkan suatu metode pembobotan berbasis kalimat *comparative* dan fitur untuk menentukan nilai rekomendasi suatu produk berdasarkan teks opini.

1.2 Perumusan Masalah

Dalam penelitian ini, masalah-masalah yang akan diselesaikan dirumuskan sebagai berikut :

1. Bagaimana mengekstrak fitur produk dari suatu teks opini
2. Bagaimana mengidentifikasi bobot kalimat *comparative* dari suatu teks opini
3. Bagaimana mengidentifikasi fitur-fitur produk yang sejenis
4. Bagaimana menentukan nilai rekomendasi untuk suatu produk berdasarkan fitur yang sejenis serta bobot kalimat *comparative* dari teks opini

1.3 Batasan Masalah

Agar fokus permasalahan yang ditangani tidak melebar, maka perlu ada beberapa batasan dalam penelitian ini. Adapun batasan-batasannya adalah sebagai berikut :

1. Penelitian ini tidak menangani kalimat opini yang berupa sindiran atau perumpamaan

2. Dataset berupa dokumen *review* produk berbahasa Inggris yang didapatkan dari situs amazon.com

1.4 Tujuan Penelitian dan Manfaat Penelitian

Tujuan dari penelitian ini adalah membandingkan dua buah produk berdasarkan penggalian informasi terhadap bobot fitur dan bobot kalimat *comparative* pada teks opini yang diberikan oleh *customer* untuk produk tersebut.

Manfaat dari penggalian informasi pada teks opini ini diharapkan akan dapat membantu konsumen dalam memutuskan produk yang hendak dibeli tanpa harus membaca keseluruhan dokumen review untuk produk tersebut.

1.5 Kontribusi Penelitian

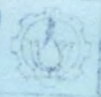
Kontribusi dari penelitian ini adalah :

1. Penambahan rule dalam metode *Double Propagation* yang digunakan saat penggalian informasi pada teks opini untuk mengekstrak dan memberikan bobot fitur produk.
2. Diusulkannya suatu metode pembobotan berbasis kalimat *comparative* dan fitur untuk menentukan nilai rekomendasi suatu produk berdasarkan teks opini.

MILIK PERPUSTAKAAN
ITS

BAB 2

KAJIAN PUSTAKA DAN DASAR TEORI



1963
ANATOLIY KULIK
DAN DOKTOR...

BAB 2

KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini akan menjelaskan teori-teori dasar yang berkaitan erat dengan topik bahasan dan tinjauan pustaka mengenai metode yang diusulkan.

2.1 POS Tagging

Part-of-Speech tagging merupakan sebuah pekerjaan dalam NLP (*Natural Language Processing*). Sebagian besar kegiatan yang dilakukan di bidang NLP seperti *Information Extraction*, *Question-Answering*, *Speech Recognition*, *Intelligent Tutoring System*, *Parser*, dan yang lainnya membutuhkan sistem POS Tagging ini untuk pemrosesan awalnya, termasuk *opinion mining*.

Part-of-speech tagging adalah sebuah sistem yang memberikan label kata secara otomatis pada suatu kalimat. Misalkan, ada kalimat "*I drink milk*". Sistem akan menerima input berupa kalimat tersebut, outputnya adalah:

I_PRP drink_VBZ milk_NN

dimana PRP adalah *pronoun* atau kata ganti, VBZ adalah *verb* atau kata kerja, dan NN adalah *noun* atau kata benda.

Ada 3 pendekatan pengklasifikasian kata yang dapat digunakan. Pendekatan pertama adalah pendekatan formal. Pada pendekatan ini, anatomi dari kata digunakan sebagai penentu klasifikasi dari kata. Sebagai contoh, kata yang berakhiran *-ing* biasanya dapat digolongkan langsung sebagai kata kerja (*verb*).

Pendekatan kedua adalah pendekatan sintaktik. Pendekatan ini menggunakan klasifikasi dari kata lain di dekat kata yang tidak teridentifikasi. Sebagai contoh, kata sifat (*adjective*) biasanya muncul tepat sebelum kata benda (*noun*) seperti *good camera*, *bad boy*, dan sebagainya.

Pendekatan terakhir adalah pendekatan konteks. Pada pendekatan ini, sebuah klasifikasi dipahami maknanya dan digunakan sebagai penentu penggolongan kata. Sebagai contoh, kata benda adalah kata yang merepresentasikan suatu objek. Pendekatan ini sangat sulit diformulasikan. Dan karenanya kurang begitu dipakai dalam pengklasifikasian kata (Jones , 1994).

2.2 Dependencies Parsing

Dependencies parsing adalah suatu proses memecah kalimat menjadi kata per kata untuk mencari relasi ketergantungan antar kata tersebut. Dalam papernya, Marneffe menuliskan bahwa relasi ketergantungan ini didapatkan dengan cara menganalisa *phrase structure parses* terlebih dahulu (Marneffe, 2006). Metode yang digunakan terdiri dari 2 fase, yaitu *dependency extraction* dan *dependency typing*. Fase *dependency extraction* sebenarnya cukup *simple*. Suatu kalimat diuraikan menggunakan *phrase structure grammar parser* agar membentuk suatu *tree*. Semua *Penn Treebank parser* dapat digunakan untuk proses tersebut (Marneffe menggunakan *Stanford parser*). *Head node* dari setiap *tree* kemudian diidentifikasi dan kata-kata lain akan bergantung pada kata tersebut. Proses pemilihan *head node* tersebut berbeda dengan proses pemilihan *head node* pada *phrase structure grammar*, dimana *dependency relation grammar* lebih memilih *verb* untuk dijadikan *head node*. Hal tersebut berdasarkan paper yang ditulis oleh Lucien Tesniere (Tesniere, 1959). Setelah itu, fase *dependency typing* dijalankan. Pada fase ini, setiap relasi dependensi yang telah terekstrak dilabeli agar mudah dipahami. Label yang disematkan pada relasi antar kata bermacam-macam bergantung pada type ketergantungannya dan siapa yang menyusunnya,

Stanford dalam aplikasinya yaitu Stanford Parser v.2.0.4 memiliki 53 jenis relasi ketergantungan yang dikenali. Berikut adalah beberapa contoh relasi yang berkaitan dengan penelitian ini.

- *amod (adjectival modifier)*

Relasi ini muncul ketika ada kata sifat (*adjective*) yang memodifikasi kata benda (*noun*) sehingga kata benda tersebut mengalami pergeseran makna atau rasa. Contoh : “*Sam eats red meat*” => *amod(meat, red)*

- *conj (conjunction)*

Adalah suatu relasi antara dua elemen yang terkoneksi oleh suatu kata konjungsi, seperti *and*, *or*, atau yang lain. Contoh : “*Bill is big and honest*” => *conj_and(big, honest)*

- *neg (negation modifier)*
Adalah suatu relasi antara kata negasi dengan kata yang dimodifikasi olehnya. Contoh : “*Bill is not a scientist*” => *neg(scientist, not)*
- *nsubj (nominal subject)*
Adalah sebuah frase kata benda yang merupakan subjek *syntactic* dari suatu klausa. Contoh : “*The baby is cute*” => *nsubj(cute, baby)*
- *det (determiner)*
Relasi *det* adalah relasi antara suatu kata benda dengan kata lain yang biasanya ada di depannya. Contoh : “*There is no zoom in this camera*” => *det(no, zoom)*
- *prep_with (preposition)*
Relasi ini sering muncul ketika ada dua buah kata yang dihubungkan oleh kata *with*. Contoh : “*I am happy with you*” => *prep_with(happy, you)*
- *advmod (adverb modifier)*
Relasi ini biasanya muncul ketika ada suatu *adverb* yang memodifikasi suatu kata kerja atau kata sifat (*adjective*). Contoh : “*He is very smart*” => *advmod(very, smart)*

2.3 Opinion Mining

Sentiment analysis/opinion mining merupakan sebuah cabang penelitian di domain *text mining*. Bidang ini adalah riset komputasional dari opini, sentimen dan emosi yang diekspresikan secara tekstual. Jika diberikan satu set dokumen teks *D* yang berisi opini (atau sentimen) mengenai suatu objek, maka *opinion mining* bertujuan untuk mengekstrak atribut dan komponen dari objek yang telah dikomentari pada setiap dokumen dan untuk menentukan apakah komentar tersebut positif, atau negatif (Liu, 2010). Pang menyebutkan bahwa *opinion mining* adalah bagian pekerjaan yang melakukan *review* yang berkaitan dengan perlakuan komputasional opini, sentimen, dan subjektifitas dari teks (Pang, 2008)

Istilah *opinion mining* muncul dalam *paper* Dave yang dipublikasikan dalam *proceeding* konferensi WWW pada tahun 2003 (Dave, 2003). Publikasi tersebut menjelaskan istilah *popularity* dalam komunitas sangat terkait dengan

pencarian *web* atau pencarian informasi. Menurut Dave, *opinion mining tool* yang ideal adalah alat yang memproses sekumpulan hasil pencarian untuk item tertentu yang menghasilkan suatu daftar atribut produk (kualitas, fitur dan lainnya) dan melakukan agregasi dari opini-opini tersebut.

Istilah sentimen digunakan dalam referensi untuk analisis otomatis dari teks evaluatif dalam melakukan analisis sentimen pasar. Istilah ini disebutkan dalam paper Nasukawa dan Yi (Yi, 2003). Sentimen juga didefinisikan sebagai perasaan yang disampaikan atau dimaksudkan untuk disampaikan, dengan kata-kata. Analisis sentimen itu sendiri difokuskan pada aplikasi spesifik yang melakukan klasifikasi *reviews* berdasarkan polaritasnya. Namun, sekarang secara luas istilah ini ditafsirkan sebagai pekerjaan dan perlakuan komputasional atas opini, sentimen dan subjektifitas dalam teks. Sehingga analisis sentimen dan *opinion mining* ditujukan pada bidang kajian yang sama, yakni sub analisis subjektifitas.

Liu mendefinisikan bahwa suatu kalimat opini merupakan kalimat yang mengekspresikan opini positif atau negatif secara eksplisit atau implisit (Liu, 2010). Liu juga mengatakan bahwa suatu kalimat opini dapat berupa kalimat subjektif atau kalimat objektif. Opini eksplisit merupakan opini yang secara eksplisit diekspresikan terhadap fitur atau objek dalam suatu kalimat subjektif. Sedangkan opini implisit merupakan opini terhadap fitur atau objek yang tersirat dalam suatu kalimat objektif. Misalnya kalimat '*The sound quality of this phone is amazing*' merupakan opini yang positif dan eksplisit. Sedangkan kalimat '*This earphone broken in two days*' merupakan opini yang negatif dan implisit. Liu mengatakan bahwa meskipun kalimat '*This earphone broken in two days*' menyampaikan fakta objektif, namun secara implisit kalimat ini mengindikasikan opini negatif terhadap '*earphone*'. Secara umum, kalimat objektif menyiratkan opini positif, negatif ataupun netral.

2.4 Feature-based Opinion Mining

Meskipun mengklasifikasikan opini ke dalam kelas positif dan negatif penting untuk dilakukan, akan tetapi terdapat informasi-informasi lain yang juga harus diketahui. Sebuah komentar positif terhadap suatu objek, tidak berarti

bahwa sang komentator menyukai segala hal tentang objek tersebut. Adakalanya ia mengatakan bahwa produk A baik, akan tetapi ada beberapa fitur dari produk tersebut yang kurang baik. Hal yang sama juga seringkali terjadi pada komentar-komentar negatif. Seperti dalam contoh kalimat *I like the camera of this smartphone, but not with its size*. Dalam kalimat tersebut, sang komentator menyatakan opini positif untuk fitur *camera* dari objek *smartphone*. Tapi di saat yang bersamaan, ia juga menyatakan opini negatif untuk atribut *size*.

Untuk mengetahui informasi-informasi tersebut, penganalisaan suatu dokumen opini harus dilakukan pada level fitur objeknya. Proses penganalisaan ini lebih dikenal dengan istilah *feature-based opinion mining*. Hal-hal yang harus dilakukan dalam proses penganalisaan tersebut adalah :

1. Mengidentifikasi fitur dari suatu objek yang sedang dikomentari, seperti pada contoh diatas.
2. Tentukan apakah fitur tersebut dikomentari positif, negatif atau netral.

Terdapat dua masalah fundamental dalam bidang *feature-based opinion mining* yaitu penyusunan *opinion lexicon* dan pengekstrakan target opini (Zhang 2010, Pang 2008).

Daftar kata opini (*opinion lexicon*) adalah daftar kata yang merepresentasikan suatu opini, seperti *bad, good, amazing, poor*, dan sebagainya. Kata ini seringkali digunakan oleh seseorang untuk menyatakan opini positif atau negatif. Penyusunan kata opini ke dalam *opinion lexicon* cukup sulit, karena suatu kata yang memiliki nilai positif untuk domain A belum tentu bernilai positif juga untuk domain B. Misal kata *long* dalam kalimat *This smartphone has long battery life* memiliki nilai positif. Sedangkan dalam kalimat *The execution time of this program is very long* bernilai negatif. Beberapa penelitian menggunakan bantuan kamus kata untuk penyusunan *opinion lexicon* (Hu, 2005). Akan tetapi metode ini dirasa kurang efektif karena hasil ekstraksi akan sangat bergantung pada kelengkapan kamus kata yang digunakan. *Wordnet* adalah salah satu kamus kata untuk dokumen berbahasa Inggris yang seringkali digunakan untuk menyusun *opinion lexicon* (Esuli, 2007). Namun, ada beberapa kata yang memiliki ketergantungan *domain* seperti kata *long* pada contoh sebelumnya. Sehingga mengandalkan bantuan kamus kata saja dirasa belum cukup. Beberapa peneliti

lain mencoba menggunakan pendekatan *unsupervised* dalam penyusunan *opinion lexicon*. Turney menggunakan nilai PMI (*Point-wise Mutual Information*) untuk mendapatkan kata yang paling mendekati kata *excellent* dan *poor* dalam penyusunan *opinion lexicon* (Turney, 2002). Kedua kata tersebut dipilih karena kata-kata itu menggambarkan kutub positif dan kutub negatif dalam suatu kalimat opini. Akan tetapi, metode ini membutuhkan waktu komputasi yang cukup lama karena perlunya akses ke *search engine* secara *online*.

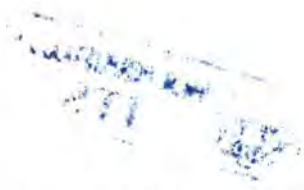
Sementara target opini ialah kepada apa (atau siapa) opini tersebut ditujukan. Target ini bisa berupa orang, organisasi, produk, atau fitur dari produk tersebut. Target opini perlu untuk diketahui karena dapat membiaskan maksud dari suatu opini. Misalkan ada suatu komentar untuk produk *Nokia 3630* sebagai berikut, *Yesterday, I bought Nokia 3630 in X store. The service of that store is really bad*. Komentar tersebut sebenarnya ditujukan untuk pelayanan dari toko X, bukan untuk *Nokia 3630*. Masalah inilah yang harus tertangani dengan baik.

2.5 Ekstraksi Fitur Produk Menggunakan Metode Double Propagation

Metode *double propagation* adalah metode *semi unsupervised* yang diusulkan oleh Qiu (Qiu, 2011). Pada dasarnya, metode ini akan mengekstrak kata opini (atau target opini) berulang kali menggunakan kata opini (atau target opini) yang telah diketahui atau telah terekstrak sebelumnya melalui identifikasi relasi *syntactic* nya. Metode ini disebut *semi unsupervised* karena masih membutuhkan bantuan dari *opinion lexicon* (kamus kata opini). Akan tetapi kamus ini tidak harus lengkap, karena dalam prosesnya kamus kata ini akan dilengkapi secara otomatis.

2.5.1 Identifikasi Relasi

Seperti disebutkan pada sub-bab sebelumnya, identifikasi relasi merupakan kunci utama dalam *double propagation* untuk mengekstrak kata opini dan target opini. Pada sub-bab ini, akan dijelaskan secara detail tentang identifikasi relasi yang dimaksud. Untuk mempermudah penjelasan, relasi antara kata opini dan target opini akan disimbolkan dengan *OT-Rel*, relasi antara kata opini dengan kata

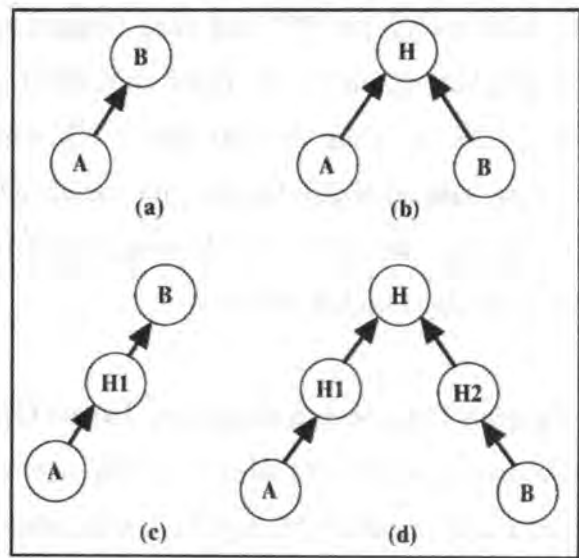


opini sebagai *OO-Rel*, dan relasi antara target opini dengan target opini sebagai *TT-Rel*.

Relasi yang dikenali dalam metode *double propagation* adalah relasi *syntactic* ketergantungan antar kata. Artinya, relasi antara 2 kata A dan B dapat dideskripsikan sebagai A bergantung pada B, atau B bergantung pada A. Terdapat 2 kategori dalam ketergantungan antar kata ini, yaitu ketergantungan langsung (*direct dependency*) dan ketergantungan tidak langsung (*indirect dependency*).

Ketergantungan langsung (KL) mengindikasikan bahwa kata A bergantung pada suatu kata B tanpa perantara, atau kata A dan kata B bergantung secara langsung pada kata C. Sedangkan relasi ketergantungan tidak langsung (KTL) terjadi jika kata A bergantung pada kata B melalui perantara kata C, atau kata A dan kata B sama-sama bergantung pada kata C melalui perantara kata D dan kata E. Lebih jelasnya dapat dilihat pada Gambar 2.1.

Gambar 2.1 menunjukkan perbedaan antara KL dan KTL. Gambar 2.1.(a) menunjukkan relasi ketergantungan langsung kata A terhadap kata B. Sedangkan Gambar 2.1.(b) menunjukkan bahwa kata A dan B sama-sama bergantung secara langsung terhadap kata H. Sementara Gambar 2.1.(c) menggambarkan situasi dimana kata A bergantung secara tidak langsung pada kata B melalui perantara kata H1. Dan Gambar 2.1.(d) menggambarkan situasi kata A dan kata B sama



Gambar 2.1. Relasi Ketergantungan Antar Kata

sama-sama bergantung secara tidak langsung pada kata H melalui perantara kata H1 dan H2.

Perlu diperhatikan bahwa baik KL maupun KTL hanya menggambarkan topologi dari semua kemungkinan relasi yang terjalin antara 2 buah kata. Karena relasi ini difungsikan untuk membantu penyusunan *opinion lexicon* dan pengekstrakan target opini, maka relasi yang diperhatikan hanya relasi yang melibatkan kedua kata tersebut. Dalam kasus ini, kata opini adalah kata *adjective*, dan target opini adalah *noun*. Hal ini didasarkan pada penelitian-penelitian sebelumnya tentang *feature-based opinion mining* (Hu, 2005).

Hu menggunakan *association rules* untuk mengekstrak fitur produk dari suatu kata opini. Tekniknya adalah dengan terlebih dahulu mencari kata benda (*noun*) yang muncul berulang kali pada dataset. Kalimat yang mengandung *noun* tersebut kemudian diduga sebagai kalimat opini. Setiap kata

Untuk mengenali kata *adjective* dan *noun* pada suatu kalimat, maka dibutuhkan aplikasi POS *Tagger*. Dalam metode *double propagation* ini, relasi antar kata didefinisikan sebagai matriks [POS(w_i), DT, R, POS(w_j)], dimana POS(w_i) adalah POS *tag* untuk kata w_i , DT adalah *Dependency Type* atau tipe relasi (KL atau KTL), R adalah relasi *syntactic* antar kata, sedangkan POS(w_j) adalah POS *tag* untuk kata w_j . Dalam papernya, Qiu mendefinisikan kata opini biasanya adalah kata *adjective*, oleh karena itu, POS *tag* yang diamati adalah *JJ*, *JJS*, dan *JJR* (Qiu, 2011). Sedangkan untuk target opini umumnya dinyatakan dalam bentuk *noun*, sehingga POS *tag* yang diamati adalah *NN* dan *NNS*. Sementara relasi *syntactic* (*R*) antara kata opini dan target opini adalah salah satu dari relasi *mod*, *pnmod*, *subj*, *s*, *obj*, *obj2*, dan *desc*. Untuk mengekstrak relasi *syntactic* ini, maka dokumen perlu di *parsing* terlebih dahulu.

2.5.2 Penyusunan *Opinion Lexicon* dan Ekstraksi Target Opini

Metode *double propagation* (DP) adalah metode *semi unsupervised*, oleh karena itu, metode ini masih membutuhkan kamus kata opini (*opinion lexicon*) yang terdiri dari kata-kata *adjective*. Akan tetapi, tidak seperti metode *supervised* yang membutuhkan kamus kata lengkap, DP hanya memerlukan sedikit kata opini

yang akan dijadikan sebagai umpan (*bootstrap*) untuk mengekstrak kata-kata yang lainnya.

DP menggunakan strategi *rule-based* dalam proses penyusunan *opinion lexicon* dan pengekstrakan target opini. Sebagai contoh, terdapat kalimat *Canon C3 takes great picture*, setelah diolah menggunakan aplikasi POS *tagger* dan *parser*, diketahui bahwa terdapat kata *adjective* yaitu *great* yang bergantung secara langsung terhadap kata *picture* yang merupakan *noun* melalui relasi *mod*. Sehingga jika disusun ke dalam bentuk matriks menjadi *[JJ,KL,mod,NN]*. Agar kata *picture* dapat dikenali sebagai suatu target, maka akan lebih mudah jika terdapat *rule* “setiap *adjective* yang bergantung secara langsung kepada suatu *noun* melalui relasi *mod*, maka *noun* tersebut adalah target opini”. Dengan membuat *rule* yang serupa, kata *great* juga bisa secara mudah dimasukkan dalam *opinion lexicon*.

Strategi tersebut dilakukan dengan memanfaatkan kata yang ada di dalam *opinion lexicon* sebagai umpan untuk mengekstrak target opini dan kata opini yang baru. Kemudian, kata-kata yang baru terekstrak tersebut dimanfaatkan untuk mengekstrak kata yang lain. Proses ini dilakukan berulang kali hingga tidak ada lagi kata baru yang ditemukan. Proses perulangan inilah yang disebut dengan *double propagation*.

2.5.2.1 Aturan Propagasi Berbasis Relasi

Terdapat 4 masalah utama dalam pengekstrakan kata opini dan target opini (dalam hal ini, target opini adalah fitur produk), yang harus ditangani oleh *double propagation* sebagai berikut :

- (1) Bagaimana mengekstrak target opini menggunakan kata opini
- (2) Bagaimana mengekstrak target opini menggunakan target opini
- (3) Bagaimana mengekstrak kata opini menggunakan target opini
- (4) Bagaimana mengekstrak kata opini menggunakan kata opini

Seperti dapat dilihat dalam keempat permasalahan tersebut, Qiu memfokuskan pengamatan pada kemunculan kata benda (*noun*), yang dianggap sebagai fitur produk, dan kata sifat (*adjective*), yang dianggap sebagai kata opini, pada suatu kalimat. Hal tersebut didasarkan pada observasi dataset serta hasil

penelitian Hu dalam papernya yang berjudul “*Mining and Summarizing Customer Review*” (Hu, 2004). Dalam paper tersebut, Hu mencoba untuk mengekstrak fitur produk dari suatu kalimat opini dengan menggunakan *association mining*. Hu mengemukakan teorema bahwa *noun* yang sering kali muncul dalam suatu dokumen opini terindikasi kuat sebagai fitur produk. Hu menghitung nilai *support* untuk setiap *noun* yang diekstrak dari dataset. *Noun* yang memiliki nilai *support* lebih dari 1% dianggap sebagai kandidat fitur produk. Untuk meyakinkan teoremanya tersebut, Hu juga memperhatikan kata sifat yang muncul di sekitar kandidat fitur produk. Hal tersebut berdasarkan penelitian sebelumnya yang mengatakan bahwa kemunculan kata sifat dalam suatu kalimat mengindikasikan kalimat tersebut adalah kalimat subjektif (Wiebe, 2000). Semakin dekat jarak antara kata sifat dan kandidat fitur produk, maka semakin kuat dugaan bahwa *noun* tersebut memang benar fitur produk. Alih-alih mengukur jarak antara kandidat fitur produk dan kata sifat dalam suatu kalimat, Qiu menggunakan relasi ketergantungan (*dependency relation*) untuk mengekstrak fitur produk dan kata opini dari suatu kalimat. Hal ini dilakukan karena sulit menentukan *threshold* suatu kata sifat dianggap dekat atau tidak dengan kandidat fitur produk dalam suatu kalimat.

Berdasarkan teorema tersebut, maka disusun aturan-aturan berbasis relasi seperti yang dapat dilihat pada Tabel 2.1. Dalam tabel tersebut, o adalah kata opini, sementara t adalah target opini. $\{O\}$ atau $\{T\}$ adalah himpunan kata opini atau target opini yang telah diketahui. H adalah suatu kata. $POS(O)$ atau $POS(T)$ adalah informasi POS tag untuk kata opini maupun target opini. Sedangkan $O-Dep$ atau $T-Dep$ adalah relasi ketergantungan antar kedua kata tersebut. $\{JJ\}$ dan $\{NN\}$ merupakan himpunan POS tag untuk kata opini dan target opini yang potensial. $\{MR\}$ adalah relasi ketergantungan *syntactic* antar kata yang dapat berupa *mod*, *pnmod*, *subj*, *s*, *obj*, *obj2*, dan *desc*. $\{CONJ\}$ adalah himpunan kata yang termasuk *conjunction*. $s.t$ adalah batasan yang harus dipenuhi. Sementara tanda panah menunjukkan arah relasi ketergantungan.

Dapat disimpulkan bahwa $R1$ diaplikasikan untuk mengekstrak target opini menggunakan kata opini yang diketahui sebelumnya. $R2$ untuk mengekstrak kata opini menggunakan target opini yang telah terekstrak

sebelumnya. $R3$ untuk mengekstrak target opini menggunakan target opini. Sedangkan $R4$ untuk mengekstrak kata opini menggunakan kata opini yang telah diketahui sebelumnya. Sebagai contoh terdapat kalimat *The phone has a good screen*. Setelah diolah dalam aplikasi POS *tagger* dan *parser* diketahui bahwa *good* adalah *JJ*, dan *screen* adalah *NN*. Keduanya berelasi dengan relasi *mod*. Jika *good* telah ada dalam *opinion lexicon*, maka aturan $R1$ bisa digunakan untuk mengekstrak target *screen*.

Tabel 2.1. Aturan Relasi dalam *Double Propagation*

RuleID	Observations	output	Examples
$R1_1$	$O \rightarrow O-Dep \rightarrow T$ s.t. $O \in \{O\}$, $O-Dep \in \{MR\}$, $POS(T) \in \{NN\}$	$t = T$	The phone has a <u>good</u> "screen". (<i>good</i> \rightarrow <i>mod</i> \rightarrow <i>screen</i>)
$R1_2$	$O \rightarrow O-Dep \rightarrow H \leftarrow T-Dep \leftarrow T$ s.t. $O \in \{O\}$, $O/T-Dep \in \{MR\}$, $POS(T) \in \{NN\}$	$t = T$	"iPod" is the <u>best</u> mp3 player. (<i>best</i> \rightarrow <i>mod</i> \rightarrow <i>player</i> \leftarrow <i>subj</i> \leftarrow <i>iPod</i>)
$R2_1$	$O \rightarrow O-Dep \rightarrow T$ s.t. $T \in \{T\}$, $O-Dep \in \{MR\}$, $POS(O) \in \{JJ\}$	$o = O$	same as $R1_1$ with <i>screen</i> as the known word and <i>good</i> as the extracted word
$R2_2$	$O \rightarrow O-Dep \rightarrow H \leftarrow T-Dep \leftarrow T$ s.t. $T \in \{T\}$, $O/T-Dep \in \{MR\}$, $POS(O) \in \{JJ\}$	$o = O$	same as $R1_2$ with <i>iPod</i> as the known word and <i>best</i> as the extract word
$R3_1$	$T_{i(j)} \rightarrow T_{i(j)}-Dep \rightarrow T_{j(i)}$ s.t. $T_{i(j)} \in \{T\}$, $T_{j(i)}-Dep \in \{CONJ\}$, $POS(T_{i(j)}) \in \{NN\}$	$t = T_{i(j)}$	Does the player play dvd with <u>audio</u> and "video"? (<i>video</i> \rightarrow <i>conj</i> \rightarrow <i>audio</i>)
$R3_2$	$T_i \rightarrow T_i-Dep \rightarrow H \leftarrow T_j-Dep \leftarrow T_j$ s.t. $T_i \in \{T\}$, $T_i-Dep == T_j-Dep$, $POS(T_i) \in \{NN\}$	$t = T_j$	Canon "G3" has a <u>great lens</u> . (<i>lens</i> \rightarrow <i>obj</i> \rightarrow <i>has</i> \leftarrow <i>subj</i> \leftarrow <i>G3</i>)
$R4_1$	$O_{i(j)} \rightarrow O_{i(j)}-Dep \rightarrow O_{j(i)}$ s.t. $O_{i(j)} \in \{O\}$, $O_{i(j)}-Dep \in \{CONJ\}$, $POS(O_{i(j)}) \in \{JJ\}$	$o = O_{i(j)}$	The camera is <u>amazing</u> and "easy" to use. (<i>easy</i> \rightarrow <i>conj</i> \rightarrow <i>amazing</i>)
$R4_2$	$O_i \rightarrow O_i-Dep \rightarrow H \leftarrow O_j-Dep \leftarrow O_j$ s.t. $O_i \in \{O\}$, $O_i-Dep == O_j-Dep$, $POS(O_i) \in \{JJ\}$	$o = O_j$	If you want to buy a <u>sexy</u> , "cool", accessory-available mp3 player, you can choose iPod. (<i>sexy</i> \rightarrow <i>mod</i> \rightarrow <i>player</i> \leftarrow <i>mod</i> \leftarrow <i>cool</i>)

2.5.2.2 Algoritma Propagasi

Gambar 2.2 menjelaskan proses algoritma secara lengkap. Dalam algoritma tersebut, *opinion lexicon* O dan review data R untuk suatu produk digunakan sebagai inputan. Sementara *opinion lexicon* yang lengkap O -Expanded dan fitur produk F merupakan outputnya. Algoritma ini akan terus bekerja mengekstrak kata opini dan fitur produk dari dokumen *review* yang disediakan hingga tidak ada lagi kata opini dan fitur produk yang tersisa. Aturan relasi $R1$, $R2$, $R3$ dan $R4$ dijadikan sebagai acuan dalam proses ekstraksi.

Sebagai contoh, diasumsikan terdapat 4 kalimat opini dalam himpunan review R , yaitu : (1) *Canon G3 takes great pictures* ; (2) *The picture is amazing* ; (3) *You may have to get more storage to store high quality pictures and recorded movies* ; (4) *The software is amazing*. Sementara dalam kamus *opinion lexicon* O , hanya terdapat kata *great*. Menggunakan baris ke 4 hingga 6 dari Gambar 2.2, didapatkan fitur produk *picture* berdasarkan aturan $R1_1$. Kemudian, dengan menggunakan fitur produk yang telah terekstrak tersebut, kata *amazing* juga dapat terekstrak sebagai kata opini pada baris ke 16 hingga 18 berdasarkan aturan $R2_2$ dan fitur *movies* terekstrak pada baris ke 13 hingga 15 berdasarkan aturan $R3_1$. Pada iterasi berikutnya, dengan kata *amazing* digunakan sebagai acuan, fitur *software* dapat terekstrak pada baris ke 4 hingga 6 berdasarkan aturan $R1_2$. Propagasi kemudian berhenti setelah memastikan semua fitur dan *opinion lexicon* yang ada dalam R telah berhasil terekstrak. Dari contoh tersebut dapat dilihat bahwa 3 fitur produk dan 1 *opinion lexicon* yang baru berhasil terekstrak hanya dengan memanfaatkan 1 *opinion lexicon* sebagai data input.

```

Input: Opinion Word Dictionary {O}, Review Data R
Output: All Possible Features {F}, The Expanded Opinion Lexicon {O-Expanded}
Function:
1. {O-Expanded} = {O}
2. {F} = ∅, {O} = ∅
3. for each parsed sentence in R
4.   if(Extracted features not in {F})
5.     Extract features {F} using R11 and R12 based on opinion words in {O-Expanded}
6.   endif
7.   if(Extracted opinion words not in {O-Expanded})
8.     Extract new opinion words {O} using R41 and R42 based on opinion words in {O-Expanded}
9.   endif
10. endfor
11. Set {F} = {F} + {Fi}, {O-Expanded} = {O-Expanded} + {Oi}
12. for each parsed sentence in R
13.   if(Extracted features not in {F})
14.     Extract features {F} using R31 and R32 based on features in {F}
15.   endif
16.   if(Extracted opinion words not in {O-Expanded})
17.     Extract opinion words {O'} using R21 and R22 based on features in {F}
18.   endif
19. end for
20. Set {Fi} = {Fi} + {F'}, {Oi} = {Oi} + {O'}
21. Set {F} = {F} + {F'}, {O-Expanded} = {O-Expanded} + {O'}
22. Repeat 2 till size({Fi}) = 0, size({Oi}) = 0

```

Gambar 2.2. Algoritma Propagasi

2.5.2.3 Penentuan Polaritas Kata Opini

Polaritas dari suatu kata opini memiliki peran penting dalam *opinion mining*. Karena polaritas tersebut yang menentukan apakah suatu target opini dikomentari baik atau buruk. Dalam *double propagation*, penentuan polaritas suatu kata opini dilakukan dengan memperhatikan kondisi kontekstual kata tersebut pada suatu kalimat opini. Hal ini didasarkan pada beberapa pengamatan berikut :

- (P1) Target yang sama akan memiliki polaritas yang sama dalam suatu dokumen opini
- (P2) Kata opini yang sama dalam satu *domain* akan memiliki polaritas yang sama pula

Berdasar pada pengamatan tersebut, kata opini yang baru terekstrak akan diberikan skor polaritas terlebih dahulu sebelum dimasukkan dalam *opinion lexicon*. Skor polaritas adalah +1 untuk kata opini positif, dan -1 untuk kata opini negatif.

Penentuan polaritas kata opini ini, pada dasarnya dilakukan untuk menentukan polaritas dari suatu target opini (fitur produk). Dalam *double propagation*, keduanya dapat dilakukan secara beriringan. Untuk itu, disusunlah aturan-aturan sebagai berikut :

- *Heterogeneous rule* : Dalam suatu dokumen opini, jika terdapat kata opini yang terekstrak oleh fitur produk, atau fitur produk yang terekstrak oleh suatu kata opini, diasumsikan bahwa keduanya memiliki polaritas yang sama. Sebagai contoh, dalam *opinion lexicon* terdapat kata “*great*” yang memiliki skor polaritas +1 (mengindikasikan bahwa kata ini positif). Jika kata tersebut digunakan untuk mengekstrak fitur “*screen*”, maka diasumsikan bahwa fitur tersebut juga memiliki skor polaritas +1, begitu pula sebaliknya. Hal ini dilakukan dengan turut memperhatikan ada atau tidaknya kata pembalik (*negasi*) seperti “*not*”, “*but*”, “*despite*”, “*neither*”, “*although*”, “*except*”, dan sejenisnya di dalam kalimat opini.
- *Homogeneous rule* : Untuk kata opini (atau fitur produk) A yang terekstrak oleh kata opini (atau fitur produk) B dalam domain yang sama,

maka diasumsikan polaritasnya juga sama, kecuali jika terdapat kata pembalik di antara keduanya.

- *Intra-review rule* : Ketika ada kata opini baru yang terekstrak oleh fitur produk dari dokumen opini yang lain, maka diasumsikan bahwa fitur tersebut tidak memiliki polaritas di dokumen saat ini. Hal ini dikarenakan masing-masing dokumen opini tidak saling berhubungan. Suatu fitur yang dikomentari positif oleh dokumen opini A, belum tentu juga dikomentari positif dalam dokumen opini B. Oleh karena itu, kata opini (fitur produk) yang terekstrak pun juga diasumsikan tidak memiliki polaritas (bernilai 0).

Penerapan ketiga aturan ini akan membuat suatu kata opini (atau fitur produk) memiliki lebih dari satu skor polaritas. Hal ini dikarenakan suatu fitur produk dapat dikomentari oleh beberapa kata opini, demikian pula sebaliknya. Oleh karena itu, skor-skor polaritas tersebut akan dijumlahkan. Jika hasil penjumlahan skor tersebut bernilai > 0 , menunjukkan bahwa kata opini (fitur produk) tersebut memiliki polaritas positif. Sebaliknya, jika hasil penjumlahannya adalah < 0 , maka kata opini (fitur produk) tersebut memiliki polaritas negatif. Sementara jika hasil penjumlahannya sama dengan 0, maka kata opini (fitur produk) tersebut memiliki polaritas netral.

2.6 Analisa Opini dalam Kalimat Comparative

Meskipun penelitian tentang *opinion mining* berkembang dengan pesat, sebagian besar penelitian ini hanya fokus pada pernyataan opini secara langsung atau opini yang ditujukan pada suatu entitas saja. Sedikit yang mencoba untuk meneliti kalimat perbandingan (*comparative*) dalam suatu teks opini, dimana kalimat jenis ini merepresentasikan tipe opini yang berbeda. Kalimat *comparative* memiliki tujuan yang sama seperti kalimat opini pada umumnya, yaitu untuk mengekspresikan rasa suka atau tidak suka terhadap suatu produk, akan tetapi dengan struktur kalimat yang sedikit berbeda. Sebagai contoh, kalimat “*The picture quality of Canon S100 is great*” merupakan kalimat opini yang diekspresikan secara langsung. Sementara kalimat “*The picture quality of Canon S100 is better than Nikon J20*” adalah kalimat *comparative*. Dapat dilihat bahwa kedua kalimat tersebut memiliki struktur yang berbeda. Kalimat *comparative* pada

dasarnya mengekspresikan rasa terhadap dua (atau lebih) produk dalam suatu kalimat yang sama. Produk yang dibandingkan pun harus memiliki fitur yang sama. Perbedaan lainnya adalah kata opini dalam suatu kalimat *comparative* memiliki bentuk yang berbeda dari kata opini yang terdapat pada kalimat opini pada umumnya. Penanganan terhadap kalimat jenis *comparative* dirasa cukup penting mengingat jumlah kalimat *comparative* yang terdapat pada suatu forum diskusi (yang berisi kalimat-kalimat opini) sekitar 10% dari total seluruh kalimat yang ada (Ganapathibhotla, 2008).

Jindal mengusulkan suatu metode untuk mengidentifikasi kalimat *comparative* pada suatu teks opini, dan mengekstrak entitas, kata pembanding serta fitur produk yang terdapat pada kalimat tersebut (Jindal, 2006). Seperti contoh dalam kalimat, “*Camera X has longer battery life than Camera Y*”, teknik yang diusulkan Jindal akan mengekstrak “*Camera X*” dan “*Camera Y*” sebagai entitas, “*longer*” sebagai kata pembanding, dan “*battery life*” sebagai fitur produk yang dibandingkan. Akan tetapi metode ini belum mampu mengetahui produk mana yang lebih diunggulkan oleh sang komentator. Dalam contoh kalimat tersebut, “*Camera X*” dengan jelas lebih diunggulkan mengacu pada fitur “*battery life*” dari kamera tersebut.

Ganapathibhotla mengusulkan perbaikan terhadap metode yang diusung oleh Jindal (Ganapathibhotla, 2008). Penelitian yang ia lakukan difokuskan untuk mengenali entitas apa yang lebih diunggulkan dalam suatu kalimat *comparative*. Dari pengamatan yang dilakukan, diketahui bahwa dalam suatu kalimat *comparative*, terdapat suatu kata pembanding seperti *better*, *worst*, *best*, *larger*, dan lain sebagainya. Kata-kata tersebut digunakan untuk menunjukkan ekspresi terhadap entitas apa yang diunggulkan. Lebih jauh lagi, kata-kata pembanding tersebut dapat dikategorikan menjadi 2, yaitu kata *comparative* (seperti *more*, *larger*, *better*, dan lainnya) serta kata *superlative* (seperti *best*, *worst*, *largest*, dan lainnya). Suatu entitas yang menjadi obyek perbandingan biasanya terletak di depan dan (atau) di belakang kata tersebut. Untuk kata *comparative* biasanya memiliki lebih dari satu entitas, sedangkan kata *superlative* biasanya hanya memiliki satu entitas saja. Untuk mengenali entitas apa yang lebih diunggulkan, maka perlu diketahui polaritas dari suatu kata pembanding. Karena kata

pembandingan ini umumnya memiliki kata dasar berupa kata sifat, maka *opinion lexicon* dapat dipergunakan untuk penentuan polaritas dari kata pembandingan tersebut. Berikutnya, dengan memanfaatkan polaritas kata pembandingan, dapat disimpulkan :

- 1) Jika terdapat kata *superlative* positif, maka entitas yang ada pada kalimat tersebut juga bernilai positif (lebih diunggulkan), demikian pula sebaliknya.
- 2) Jika terdapat kata *comparative* positif, maka entitas A yang muncul sebelum kata *comparative* tersebut bernilai positif yang artinya lebih diunggulkan daripada entitas B yang muncul setelah kata *comparative*. Sebaliknya, jika polaritas dari kata *comparative* adalah negatif, bermakna entitas B yang lebih diunggulkan daripada entitas A.

2.7 Analisa Opini untuk Perbandingan Produk

Sebagian peneliti melakukan perbandingan produk secara otomatis dengan memanfaatkan teks opini. Hal ini didasarkan pada pengamatan bahwa seorang calon pembeli cenderung lebih suka membandingkan kelebihan dan kekurangan antara sebuah produk dengan produk lain sebelum melakukan pembelian.

Liu membandingkan dua buah produk dengan mengamati tiap fitur dari kedua produk tersebut (Liu, 2005). Proses perbandingan ini diawali dengan mengekstrak fitur dari setiap produk yang akan dibandingkan. Kemudian, fitur yang telah terekstrak akan diberikan bobot berdasarkan jumlah komentar positif dan negatif untuk fitur tersebut. Bobot fitur tiap produk inilah yang dijadikan acuan untuk melakukan perbandingan.

Sun melakukan hal yang sedikit berbeda, dimana ia tidak hanya mengamati kalimat subjektif seperti yang dilakukan oleh Liu, tapi juga memperhatikan kalimat objektif untuk memberikan bobot pada suatu fitur produk (Sun, 2008). Kalimat objektif yang dimaksud adalah dokumen spesifikasi produk yang pada umumnya disediakan oleh produsen. Penanganan terhadap kalimat objektif ini dilakukan dengan membuat suatu pohon evolusi produk. Dalam pohon evolusi ini terdapat sejarah tentang produk tersebut, misalkan *Canon T100* merupakan pengembangan dari *Canon T50*, sedangkan *T50* adalah pengembangan dari *Canon S Series*. Dari informasi ini, maka dapat disimpulkan bahwa *Canon*

S50 (yang merupakan salah satu *varian* dari *Canon S Series*) memiliki spesifikasi yang lebih jelek daripada *Canon T100*. Dengan menggabungkan antara kalimat subjektif dan objektif ini, Sun mampu memberikan nilai rekomendasi untuk suatu produk. Akan tetapi, penggunaan kalimat objektif untuk memberikan nilai rekomendasi terhadap suatu produk dirasa kurang tepat. Hal ini dikarenakan tidak selamanya produk yang baru itu lebih disukai oleh konsumen dibandingkan dengan produk yang lama. Beberapa konsumen lebih menyukai produk yang lama dikarenakan sesuatu hal, seperti kemudahan penggunaannya, kelengkapan fitur yang ditawarkan, dan lain-lain.

Para peneliti tidak hanya melakukan perbandingan produk berdasarkan fiturnya. Zhang melakukan perbandingan dengan mengamati produk tersebut secara keseluruhan (Zhang, 2011). Ia memanfaatkan fitur "*Most Helpful Review*" yang disediakan oleh situs *amazon.com* untuk melihat review mana yang paling mencerminkan isi hati dari pembaca. Hal ini dilakukan oleh Zhang dengan alasan bahwa tidak semua konsumen menuliskan opininya pada kolom komentar. Beberapa konsumen cenderung untuk memilih komentar yang paling mencerminkan isi hati mereka dan memberikan skor untuk komentar tersebut. Skor yang dimiliki oleh tiap komentar inilah yang akan menentukan seberapa baik (atau buruk) kualitas suatu produk. Metode ini dirasa cukup efektif, akan tetapi jika yang diinginkan adalah perbandingan produk berdasarkan fitur, metode ini tidak bisa diterapkan karena sifatnya yang *general*.

[Halaman ini sengaja dikosongkan]

BAB 3

METODA PENELITIAN

3.1 Pengumpulan dan Analisis Data

Dataset yang akan digunakan dalam penelitian ini adalah dokumen review produk berbahasa Inggris yang didapatkan dari situs Amazon (www.amazon.com). Situs ini dipilih karena merupakan salah satu situs jual beli online terbesar dan memiliki customer yang tersebar hampir di seluruh dunia. Dokumen review produk didapatkan melalui metode *crawling* dengan memanfaatkan API yang sudah disediakan oleh Amazon. Dengan menggunakan API ini, selain dokumen review, juga bisa didapatkan informasi-informasi lain seperti nama produk, *rating* yang diberikan customer untuk tiap produk, id member yang memberikan komentar, skor *helpful feedbacks* dari tiap dokumen review, dan lain sebagainya. Produk yang akan diambil dokumen reviewnya adalah 10 produk untuk masing-masing *domain* yang berbeda, yaitu *smartphone*, TV dan kamera digital. Total dokumen *review* untuk seluruh produk tersebut adalah 4.848 dokumen. Dengan total kalimat sebanyak 17.826 buah.. Gambar 3.1 menunjukkan contoh *review* dalam situs amazon.com yang akan digunakan sebagai dataset.

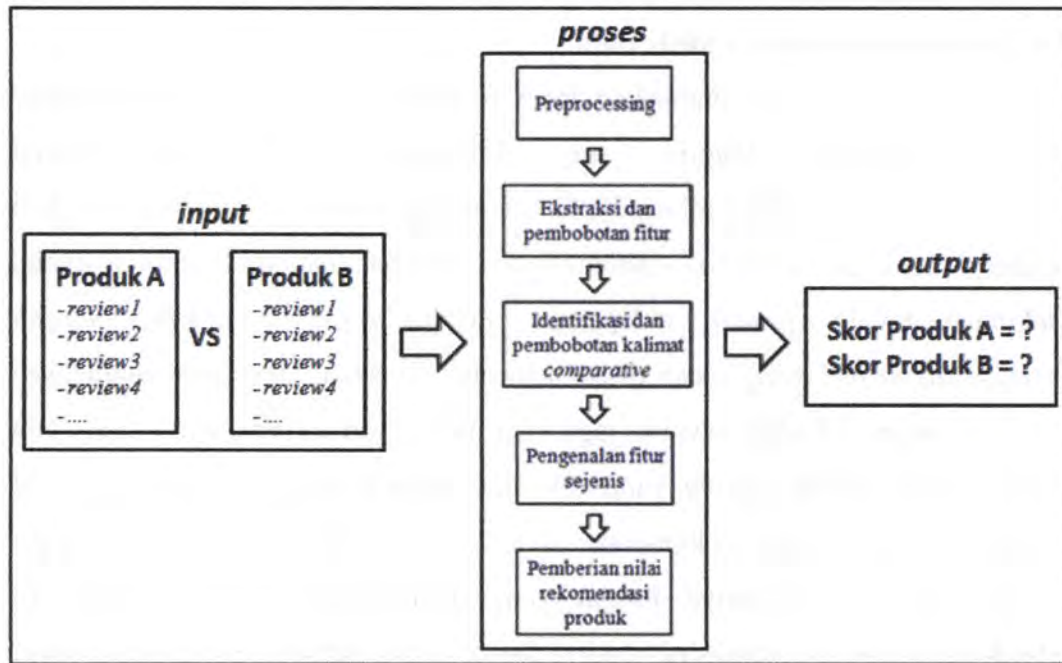


Gambar 3.1. Contoh Review dalam Situs Amazon.com

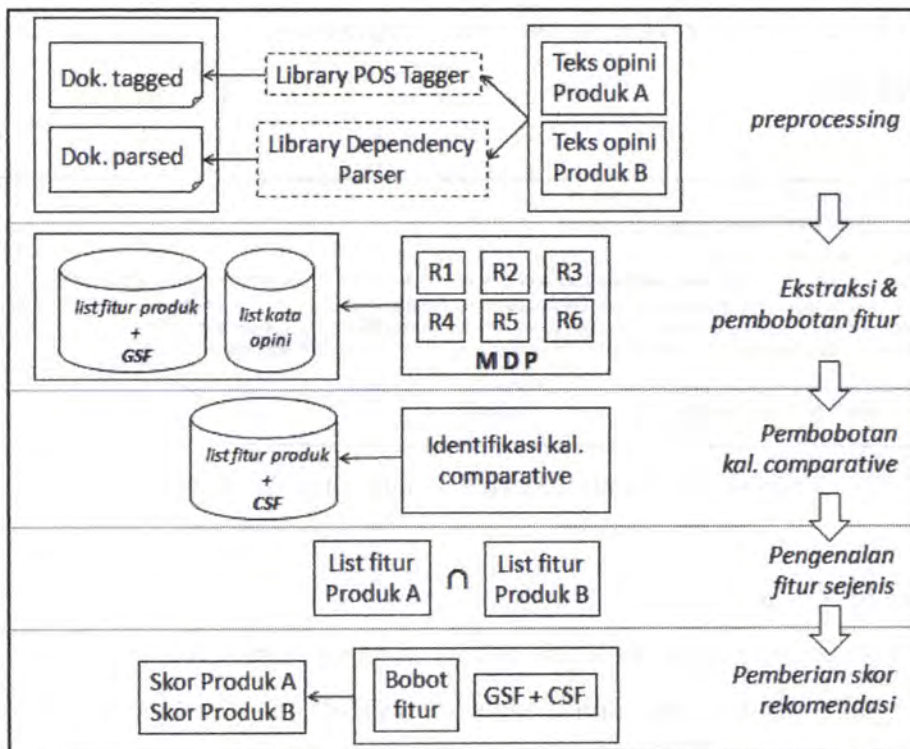
3.2 Desain Sistem

Sistem yang akan dibangun adalah sistem pemberi rekomendasi produk dengan memanfaatkan teks opini. Input dari sistem ini adalah semua dokumen *review* yang dimiliki oleh suatu produk. Sedangkan output yang dihasilkan adalah

skor rekomendasi yang dimiliki oleh setiap produk. Alur kerja sistem secara umum dapat dilihat pada Gambar 3.2.



Gambar 3.2. Tahapan Alur Kerja Sistem



Gambar 3.3. Diagram Proses Perbandingan Produk

Pada Gambar 3.3, dapat dilihat bahwa proses perbandingan produk terdiri dari 5 tahapan. Dimana masing-masing tahap merupakan modul tersendiri yang memanfaatkan output dari modul yang lain. Untuk memudahkan dalam pemahaman tiap tahapan alur kerja sistem, maka penjelasan disajikan dalam bentuk contoh kasus sebagai berikut.

Terdapat 3 buah dokumen *review* terhadap suatu produk kamera SLR yaitu *Canon G3*. Dokumen *review* tersebut antara lain :

- **Review A**

(A1) *Canon G3 takes great picture.* (A2) *The picture is very amazing.* (A3) *You may have to get more storage to store high quality pictures and movies.* (A4) *The display is also amazing.*

- **Review B**

(B1) *There is no zoom included in this camera.*

- **Review C**

(C1) *Canon G3 has an amazing screen.* (C2) *I think the screen of Canon G3 is better than Nikon S100.*

3.2.1 Preprocessing

Sebelum diproses, teks opini terlebih dahulu akan diolah menggunakan library *POS Tagger* dan *Dependency Parser*. Kedua library tersebut merupakan modul yang disediakan oleh stanford.edu untuk mengolah data teks, yang dapat diunduh secara gratis melalui internet dengan alamat <http://nlp.stanford.edu/software/>.

- *POS Tagging*

Proses ini dilakukan untuk mendapatkan informasi *tag* dari tiap kata yang ada pada dataset. Misal untuk kalimat (A1), setelah dilakukan proses *POS tagging*, hasilnya menjadi :

(A1') *Canon_NNP G3_NNP takes_VBZ great_JJ picture_NN . _.*

Dimana *NNP* adalah frase kata benda (*noun*), *VBZ* adalah kata kerja (*verb*) dan *JJ* adalah kata sifat (*adjective*).

- *Dependency Parsing*

Proses *dependency parsing* dilakukan untuk mendapatkan informasi relasi tiap kata dalam suatu kalimat. Misal untuk kalimat (A1), setelah dilakukan proses *parsing*, hasilnya adalah :

nm(G3-2, Canon-1)

nsubj(takes-3, G3-2)

root(ROOT-0, takes-3)

amod(picture-5, great-4)

doj(takes-3, picture-5)

Dari beberapa relasi tersebut tidak semuanya digunakan, tapi hanya beberapa saja yang sesuai dengan aturan MDP yang akan dijelaskan pada sub-bab berikutnya.

3.2.2 Ekstraksi dan Pembobotan Fitur

Proses ekstraksi dan pembobotan fitur mempunyai peran penting dalam merekomendasi suatu produk. Hal ini dikarenakan nilai rekomendasi akan diberikan berdasarkan fitur yang terdapat pada produk tersebut.

Metode yang digunakan untuk proses pengekstrakan dan pembobotan fitur ini adalah *Double Propagation (DP)*. Metode ini dipilih karena kemampuannya membuat kamus kata opini secara otomatis, sehingga tidak diperlukan kamus kata yang lengkap. Metode ini juga tidak memerlukan data latih sehingga tidak bergantung pada kelengkapan dataset. Akan tetapi, metode ini memiliki kelemahan dimana kurang cocok untuk menangani dokumen opini yang terlalu pendek, atau tidak memiliki kata opini dalam dokumen tersebut. Oleh karena itu, dalam penelitian ini, akan dilakukan sedikit modifikasi pada metode DP yakni dengan menambahkan aturan baru.

Algoritma *sequential covering* digunakan untuk menambahkan aturan baru pada metode DP. Cara kerjanya adalah dengan menganalisa akurasi dari tiap aturan yang dibuat, kemudian secara bertahap memperbaiki aturan tersebut atau menambahkan aturan baru untuk meningkatkan akurasinya. Algoritma ini sebenarnya adalah algoritma untuk menyusun *classification rule*. Jika kasus pengekstrakan fitur produk dianggap sebagai kasus pengklasifikasian, dimana

dataset akan dipisahkan menjadi class A (class kalimat yang mengandung fitur produk) dan class B (class kalimat yang tidak mengandung fitur produk), maka algoritma ini bisa digunakan untuk menganalisa aturan yang ada pada metode DP. Dari pengamatan yang dilakukan pada dataset, banyak terdapat kalimat yang terindikasi memiliki fitur produk tapi tidak memiliki kata opini. Padahal kata opini ini adalah salah satu indikator kuat yang digunakan oleh metode DP untuk mengatakan suatu kalimat mengandung fitur produk atau tidak. Oleh karena itu, metode DP akan memasukkan kalimat-kalimat tersebut ke dalam class B (seharusnya dimasukkan dalam class A). Dari hasil pengamatan tersebut, dapat disimpulkan bahwa aturan-aturan yang terdapat dalam metode DP belum sempurna untuk mengklasifikasikan seluruh kalimat yang ada pada dataset, sehingga masih memungkinkan untuk ditingkatkan akurasi dengan cara penambahan aturan baru. Dengan menggunakan algoritma *sequential covering* ini, nilai akurasi dari aturan yang baru akan dianalisa dengan menggunakan persamaan berikut :

$$akurasi(r) = \frac{p}{t} \quad (3.1)$$

dimana r adalah aturan baru yang dianalisa, t adalah seluruh data yang memenuhi kondisi *antecedent* aturan r , dan p adalah jumlah data di dalam t yang bernilai positif/benar. Suatu aturan dikatakan baik jika akurasi bernilai 1 atau mendekati nilai tersebut.

Selain itu, modifikasi lain yang dilakukan adalah penambahan bobot fitur ketika terdapat kata penguat opini yang muncul dalam kalimat (seperti *very*, *also*, dan lain-lain). Tabel 3.1 menunjukkan aturan ekstraksi fitur dan kata opini yang digunakan dalam metode *Modification Double Propagation (MDP)*. Dimana aturan baru yang ditambahkan adalah aturan R5 dan R6.

Tabel 3.1 Tabel Aturan Ekstraksi Kata Opini dan Fitur Produk dalam MDP

Relasi	ID	Parameter Penting	Aksi
$amod(noun, adj)$	R1 ₁	<i>noun</i> ada dalam daftar fitur produk	Ekstrak <i>adjective</i> sebagai kata opini dan masukkan dalam <i>opinion lexicon</i>
	R1 ₂	<i>adjective</i> ada dalam	Ekstrak <i>noun</i> sebagai fitur

		<i>opinion lexicon</i>	produk dan masukkan ke dalam daftar fitur produk
<i>nsubj(adj, noun)</i>	R2 ₁	<i>noun</i> ada dalam daftar fitur produk	Ekstrak <i>adjective</i> sebagai kata opini dan masukkan dalam <i>opinion lexicon</i>
	R2 ₂	<i>adjective</i> ada dalam <i>opinion lexicon</i>	Ekstrak <i>noun</i> sebagai fitur produk dan masukkan ke dalam daftar fitur produk
<i>conj_and(adj, adj)</i>	R3	salah satu <i>adjective</i> ada dalam <i>opinion lexicon</i>	Ekstrak <i>adjective</i> yang lain sebagai kata opini dan masukkan dalam <i>opinion lexicon</i>
<i>conj_and(noun, noun)</i>	R4	Salah satu <i>noun</i> ada dalam daftar fitur produk	Ekstrak <i>noun</i> yang lain sebagai fitur produk dan masukkan dalam daftar fitur produk
<i>det(no, noun)</i>	R5	Noun belum ada dalam daftar fitur produk	Ekstrak <i>noun</i> sebagai fitur produk dan masukkan ke dalam daftar fitur produk
<i>prep_with(adj, noun)</i>	R6	<i>adjective</i> ada dalam <i>opinion lexicon</i>	Ekstrak <i>noun</i> sebagai fitur produk dan masukkan ke dalam daftar fitur produk

Sedangkan untuk pemberian skor (pembobotan) kata opini dan fitur produk dilakukan bersamaan pada waktu proses ekstraksi berjalan dengan memanfaatkan aturan yang ditunjukkan oleh Tabel 3.2 serta persamaan (3.2) sampai (3.7).

Tabel 3.2 Tabel Pembobotan Fitur Produk dan Kata Opini dalam MDP

Relasi	ID	Parameter Penting	Aksi
<i>advmod(adj, adv)</i>	RS1	<i>adjective</i> ada dalam <i>opinion lexicon</i>	$S_v = 0,5$
<i>neg(adj, not)</i>	RS2	<i>adjective</i> ada dalam <i>opinion lexicon</i>	$S_n = -1$

1. Penentuan global skor fitur produk

$$GSF = GSF + LSF \quad (3.2)$$

2. Pembobotan fitur produk yang terekstrak oleh aturan R1₂, R2₂ dan R6

$$LSF = LSF + ((SO + (SO \times Sv)) \times Sn) \quad (3.3)$$

3. Pembobotan fitur produk yang terekstrak oleh aturan R4

$$LSF = LSF + pol_F_{eks} \quad (3.4)$$

4. Pembobotan fitur produk yang terekstrak oleh aturan R5

$$LSF = -1 \quad (3.5)$$

5. Pemberian skor pada kata opini yang terekstrak oleh aturan R1₁ dan R2₁

$$SO = pol_F \quad (3.6)$$

6. Pemberian skor pada kata opini yang terekstrak oleh aturan R3

$$SO = SO_{eks} \quad (3.7)$$

dimana,

GSF = global skor fitur ($GSF = 0$ untuk setiap fitur yang baru diekstrak)

LSF = lokal skor fitur ($LSF = 0$ untuk setiap awal review)

$$pol_F = \text{polaritas fitur} = \begin{cases} -1, & LSF < 0 \\ 0, & LSF = 0 \\ 1, & LSF > 0 \end{cases}$$

SO = skor kata opini

Sv = skor kata penguat opini (nilai Sv mengikuti aturan RS1)

Sn = skor kata negasi (nilai Sn mengikuti aturan RS2)

pol_F_{eks} = polaritas fitur yang menjadi ekstraktor

SO_{eks} = skor kata opini yang menjadi ekstraktor

Untuk menghemat waktu komputasi yang digunakan pada saat pengecekan aturan (*rule*), maka dokumen parser yang dihasilkan pada tahapan *preprocessing* difilter terlebih dahulu. Gambar 3.4 menjelaskan proses filter yang dilakukan dengan cara membuang relasi-relasi yang dianggap tidak penting sehingga yang tersisa hanya relasi-relasi yang memang sesuai dengan kondisi salah satu *rule*. Relasi-relasi yang dianggap penting tersebut adalah relasi *amod*, *nsubj*, *conj_and*, *det*, *neg*, dan *advmod*.

```

1 important_relation_list={amod, nsubj, conj_and, det, prep_with, neg, advmod};
2 fix_parser={};
3 for each relation in parser_document
4     if relation match with important_relation_list
5         fix_parser.add(relation);
6 end

```

Gambar 3.4 *Pseudocode* Penyaringan Relasi-relasi Penting

Di samping itu, penghematan waktu komputasi juga dilakukan dengan cara memberikan level prioritas pada tiap aturan dalam MDP. Level prioritas ini akan digunakan untuk mengatur letak (urutan) aturan pada saat implementasi metode MDP pada *source code*. Aturan yang memiliki level prioritas tertinggi akan diletakkan dalam urutan pertama dan diikuti oleh aturan-aturan lain yang memiliki level prioritas lebih rendah. Cara untuk menentukan level prioritas ini adalah dengan menghitung nilai *coverage* dari masing-masing aturan. Nilai *coverage* ini didapatkan dari persamaan berikut :

$$Coverage(r) = \frac{t}{n} \quad (3.8)$$

dimana r adalah aturan yang akan dihitung nilai *coverage*-nya, t adalah jumlah data yang memenuhi kondisi *antecedent* dari aturan r , dan n adalah jumlah seluruh data *sample*. Dari persamaan tersebut, dapat disimpulkan bahwa semakin banyak data yang kondisinya sesuai dengan *antecedent* dari aturan r , maka semakin besar nilai *coverage* yang dimiliki oleh aturan tersebut, dan semakin tinggi pula level prioritasnya.

Pada contoh kasus di atas, pertama yang harus disiapkan adalah kamus kata opini (*opinion lexicon*). Kamus ini berisi kata-kata sifat yang merepresentasikan opini beserta polaritasnya (positif atau negatif). Tabel 3.3 memperlihatkan contoh *opinion lexicon*.

Tabel 3.3. Kamus Kata Opini (*Opinion Lexicon*)

KATA	SKOR
Great	1

Metode MDP tidak membutuhkan *opinion lexicon* yang lengkap, karena dalam prosesnya, kamus kata tersebut akan dilengkapi secara otomatis. Dengan

menggunakan *opinion lexicon* dan aturan-aturan tersebut, maka kata opini beserta fitur produk dapat diekstrak dari dataset seperti berikut.

- (A1)

POS : *Canon_NNP G3_NNP takes_VBZ great_JJ picture_NN . _ .*

Parsing :

amod(picture-5, great-4)

Dapat dilihat bahwa dalam dokumen parser untuk kalimat (A1) hanya ada satu relasi antar kata yang tersisa yaitu relasi *amod(picture, great)*. Hal ini dikarenakan relasi lain dianggap tidak penting karena tidak akan memenuhi satupun aturan MDP, sehingga dapat dihilangkan. Setelah diamati, dalam (A1) terdapat kata *great* yang terdapat pada *opinion lexicon*, maka kata *picture* otomatis akan terekstrak berdasarkan aturan R1₂. Sedangkan skor untuk *picture* dihitung berdasarkan persamaan (3.3).

$$\begin{aligned} LSF_{picture} &= LSF_{picture} + \left((Pol_{O_{great}} + (Pol_{O_{great}} \times Sv)) \times Sn \right) \\ &= 0 + ((1 + (1 \cdot 0)) \cdot 1) \\ &= 1 \end{aligned}$$

Tabel 3.4. Daftar Fitur Produk Setelah Kalimat A1 Dievaluasi

FITUR	GSF	LSF
Picture	0	1

- (A2)

POS : *The_DT picture_NN is_VBZ very_RB amazing_JJ . _ .*

Parsing :

nsubj(amazing-5, picture-2)

advmod(amazing-5, very-4)

Dalam (A2) terdapat kata *picture*, sehingga dengan memanfaatkan aturan R2₁ didapatkan kata opini kedua yaitu *amazing*. Untuk mendapatkan skor dari kata *amazing*, persamaan (3.6) dapat digunakan.

Tabel 3.5. Daftar Kata Opini Setelah Kalimat A2 Dievaluasi

KATA	SKOR
Great	1
Amazing	1

Kemudian, karena fitur *picture* dikomentari lebih dari sekali, maka skornya juga harus diupdate menggunakan persamaan (3.3). Perhatikan juga bahwa terdapat telasi *advmod*, sehingga nilai dari *Sv* adalah 0,5

$$\begin{aligned}
 LSF_{picture} &= LSF_{picture} + \left((Pol_{O_{amazing}} + (Pol_{O_{amazing}} \times Sv)) \times Sn \right) \\
 &= 1 + ((1 + (1 \cdot 0,5)) \cdot 1) = 2,5
 \end{aligned}$$

Tabel 3.6. Daftar Fitur Produk Setelah Kalimat A2 Dievaluasi

FITUR	GSF	LSF
Picture	0	2,5

• (A3)

POS : You_PRP may_MD have_VB to_TO get_VB more_JJR storage_NN to_TO store_VB high_JJ quality_NN picture_NNS and_CC movie_NNS . .

Parsing :

amod(storage-7, more-6)

amod(picture-12, high-10)

conj_and(picture-12, movie-14)

Relasi *amod(storage, more)* dan *amod(picture, high)* untuk sementara diabaikan karena tidak memenuhi aturan R1 maupun R2. Berikutnya, dengan memanfaatkan kata *picture* yang terdapat dalam kalimat (A3), kata opini *high* juga dapat terekstrak berdasarkan aturan R1₁.

Tabel 3.7. Daftar Kata Opini Setelah Kalimat A3 Dievaluasi

KATA	SKOR
great	1
amazing	1
high	1

BAB 3

METODA PENELITIAN

$$\begin{aligned}
 LSF_{picture} &= LSF_{picture} + \left((Pol_{O_{high}} + (Pol_{O_{high}} \times Sv)) \times Sn \right) \\
 &= 2,5 + ((1 + (1 \cdot 0)) \cdot 1) \\
 &= 3,5
 \end{aligned}$$

Dalam (A3) juga terdapat fitur lain yaitu *movie*, kata ini juga dapat terekstrak dengan memanfaatkan kata *picture* berdasarkan aturan R3₂. Karenanya, untuk menghitung *LSF movie* digunakan persamaan (3.4).

$$\begin{aligned}
 LSF_{movie} &= LSF_{movie} + Pol_{F_{picture}} \\
 &= 0 + 1 \\
 &= 1
 \end{aligned}$$

Tabel 3.8. Daftar Fitur Produk Setelah Kalimat A3 Dievaluasi

FITUR	GSF	LSF[A]
picture	0	3,5
movie	0	1

• (A4)

POS : *The_DT display_NN is_VBZ also_RB amazing_JJ . .*

Parsing :

nsubj(amazing-20, display-17)

advmod(amazing-20, also-19)

Kemudian, dengan memanfaatkan kata *amazing* yang terdapat dalam kalimat (A4), fitur *display* juga dapat terekstrak berdasarkan aturan R1₂.

$$\begin{aligned}
 LSF_{display} &= LSF_{display} + \left((Pol_{O_{amazing}} + (Pol_{O_{amazing}} \times Sv)) \times Sn \right) \\
 &= 0 + ((1 + (1 \cdot 0,5)) \cdot 1) \\
 &= 1,5
 \end{aligned}$$

Tabel 3.9. Daftar Fitur Produk Setelah Kalimat A3 Dievaluasi

FITUR	GSF	LSF[A]
Picture	0	3,5
Movie	0	1
Display	0	1,5

Setelah semua kalimat dalam Review A telah selesai dianalisa, berikutnya adalah mengupdate skor GSF berdasarkan persamaan (3.2).

Tabel 3.10. Daftar Fitur Produk Setelah Review A Dievaluasi

FITUR	GSF	LSF[A]
Picture	3,5	3,5
Movie	1	0
Display	1,5	0

• **(B1)**

POS : *There_EX is_VBZ no_DT zoom_NN included_VBN in_IN this_DT camera_NN . .*

Parsing :

det(zoom-4, no-3)

Untuk kalimat (B1) yang tidak memiliki kata sifat, akan tetapi terindikasi sebagai kalimat opini, aturan R4 dapat digunakan untuk mendapatkan fitur *zoom*. Skor LSF dari fitur tersebut dapat dihitung dengan menggunakan persamaan (3.5).

Tabel 3.11. Daftar Fitur Produk Setelah Kalimat B1 Dievaluasi

FITUR	GSF	LSF[B]
Picture	3,5	0
Movie	1	0
Display	1,5	0
Zoom	0	-1

Karena review B hanya terdiri dari 1 kalimat, maka skor GSF dan LSF akan langsung diupdate.

Tabel 3.12. Daftar Fitur Produk Setelah Review B Dievaluasi

FITUR	GSF	LSF[B]
Picture	3,5	0
Movie	1	0
Display	1,5	0
Zoom	-1	0

- (C1)

POS : *Canon_NNP G3_NNP has_VBZ an_DT amazing_JJ screen_NN . .*

Parsing :

amod(screen-6, amazing-5)

Pada kalimat (C1), fitur *screen* diekstrak dengan menggunakan aturan R1₂ memanfaatkan kata *amazing* yang terdapat dalam kalimat tersebut.

$$\begin{aligned}
 LSF_{screen} &= LSF_{screen} + \left((Pol_{O_{amazing}} + (Pol_{O_{amazing}} \times Sv)) \times Sn \right) \\
 &= 0 + ((1 + (1 \cdot 0)) \cdot 1) \\
 &= 1
 \end{aligned}$$

Tabel 3.13. Daftar Fitur Produk Setelah Kalimat C1 Dievaluasi

FITUR	GSF	LSF[C]
picture	3,5	0
movie	1	0
display	1,5	0
zoom	-1	0
screen	0	1

- (C2)

POS : *I_PRP think_VBP the_DT screen_NN of_IN Canon_NNP G3_NNP is_VBZ better_JJR than_IN Nikon_NNP S100_NNP . .*

Parsing :

nsubj(better-9, screen-4)

Kemudian, dengan memanfaatkan fitur *screen* yang telah terekstrak sebelumnya, kata opini baru yaitu *better* dapat terekstrak berdasarkan aturan R2₁.

Tabel 3.14. Daftar Kata Opini Setelah Kalimat C2 Dievaluasi

KATA	SKOR
Great	1
Amazing	1
High	1
Better	1

Dan karena fitur *screen* dikomentari untuk yang kedua kalinya, maka nilai LSF nya jga harus diupdate.

$$\begin{aligned}
 LSF_{screen} &= LSF_{screen} + ((Pol_{O_{better}} + (Pol_{O_{better}} \times Sv)) \times Sn) \\
 &= 1 + ((1 + (1 \cdot 0)) \cdot 1) \\
 &= 2
 \end{aligned}$$

Berikutnya skor GSF untuk semua fitur diupdate berdasarkan persamaan (3.2).

Tabel 3.15. Daftar Fitur Produk Setelah Review C Dievaluasi

FITUR	GSF	LSF[C]
Picture	3,5	0
Movie	1	0
display	1,5	0
Zoom	-1	0
Screen	2	2

Dengan menggunakan metode *Modification Double Propagation*, 3 kata opini baru dan 5 fitur produk berhasil terekstrak hanya dengan memanfaatkan 1 buah kata opini.

3.2.3 Identifikasi dan Pembobotan Kalimat *Comparative*

Dalam penelitian ini, untuk mendapatkan nilai rekomendasi suatu produk, bobot kalimat *comparative* yang terdapat pada suatu teks opini juga ikut diperhatikan. Untuk mengidentifikasi keberadaan kalimat *comparative* dalam suatu teks opini, dilakukan dengan cara mencari adanya kata yang memiliki tag

JJR (*adjective comparative*) dan JJS (*adjective superlative*). Akan tetapi, tidak semua kalimat *comparative* yang teridentifikasi dapat digunakan, seperti kalimat *My bag is smaller than this camera*. Kalimat ini tidak bisa digunakan untuk mengetahui keunggulan suatu produk atas produk yang lain. Oleh karena itu, perlu ada suatu mekanisme untuk mendeteksi kalimat *comparative* mana yang harus diperhatikan dan kalimat mana yang bisa dihiraukan.

Karena penelitian ini bertujuan untuk memberikan nilai rekomendasi antara beberapa produk yang dibandingkan, maka kalimat *comparative* yang diperhatikan adalah kalimat yang didalamnya terdapat entitas produk yang akan dibandingkan. Oleh karena itu, diasumsikan daftar nama-nama produk yang terdapat pada dataset telah diketahui sebelumnya. Untuk lebih jelas, penjelasan tentang bagaimana mendapatkan nilai CSF (*Comparative Score Feature*) akan dijelaskan dalam bentuk contoh kasus berikut.

Misalkan produk yang dibandingkan adalah “*LG Optimus L7*” dan “*Sony Xperia U*”. Sedangkan produk yang diamati saat ini adalah “*LG Optimus L7*”. Dalam dataset terdapat tiga buah dokumen review berikut :

Review A

(A1) I confuse to buy this smartphone or Sony Xperia U. (A2) Xperia U has a lower prize, bigger screen and nicer design. (A3) But L7 has a greater processor, better camera and longer battery life than Xperia. (A4) Anyone, please help me...

Review B

(B1) I buy this smartphone yesterday. (B2) I think, it is worse than my friend's phone, Xperia U. (B3) And the thing that make me really disappointed is the fact that L7's prize is more expensive than xperia.

Review C

(C1) I like everything from this smartphone. (C2) Its screen is big, nice performance, cute design. (C3) I think this is the best phone that i ever had.

Melalui metode MDP, didapatkan daftar kata opini dan fitur produk seperti pada Tabel 3.16 dan 3.17 berikut.

Tabel 3.16. Contoh Daftar Kata Opini Yang Diekstrak Oleh MDP

Lexicon	Bobot
Good	1
Better	1
Bad	-1
Worse	-1
Best	1
Lower	1
Bigger	1
Nicer	1
Cuter	1
Greater	1
Longer	1
expensive	-1

Tabel 3.15. Contoh Daftar Fitur Produk Yang Diekstrak Oleh MDP

Fitur Produk	GSF
Phone	4
Camera	2.5
Screen	-1
Prize	-1.5
Design	2
Processor	1.5
Battery life	1

Untuk mendapatkan nilai CSF dari kalimat *comparative*, berikut langkah-langkahnya :

1. Cek kalimat yang mengandung sebagian atau seluruh kata “*Sony Xperia U*” dan kata **JJR/more**, simpan di list `ComparativeSenList[] = {A2,A3,B2,B3}`
2. Untuk tiap kalimat *comparative* yang ditemukan, lakukan langkah berikut :
 - Cari pasangan (lexicon+skor) – fitur produk dari masing2 kalimat

Tabel 3.18. Pasangan Lexicon-Fitur Untuk Tiap Kalimat *Comparative*

Kalimat	Pasangan lexicon-fitur	Skor lexicon
(A2)	Lower – prize	+1
	Bigger – screen	+1
	Nicer – design	+1

(A3)	Greater – processor	+1
	Better – camera	+1
	Longer – battery life	+1
(B2)	Worse – it (phone)	-1
(B3)	Expensive – prize	-1

- Amati letak frase “*Sony Xperia U*” dalam kalimat
 - IF letaknya di sebelah **kanan** lexicon AND skor lexicon = +1, THEN CSF = +1
 - ELSE IF letaknya di sebelah **kanan** lexicon AND skor lexicon = -1, THEN CSF = -1
 - ELSE IF letaknya di sebelah **kiri** lexicon AND skor lexicon = +1, THEN CSF = -1
 - ELSE IF letaknya di sebelah **kiri** lexicon AND skor lexicon = -1, THEN CSF = +1

Tabel 3.19. Nilai CSF Hasil Pengamatan Kalimat Comparative

Kalimat	Pasangan lexicon-fitur	Skor lexicon	Lokasi kata	CSF
(A2)	Lower – prize	+1	Kiri	-1
	Bigger – screen	+1	Kiri	-1
	Nicer – design	+1	kiri	-1
(A3)	Greater – processor	+1	Kanan	+1
	Better – camera	+1	Kanan	+1
	Longer – battery life	+1	Kanan	+1
(B2)	Worse – it (phone)	-1	Kanan	-1
(B3)	Expensive – prize	-1	Kanan	-1

- Update CSF yang dimiliki tiap fitur seperti pada Tabel 3.20

Tabel 3.20. Nilai CSF Untuk Tiap Fitur Produk

Fitur produk	GSF	CSF (vs Xperia U)
Phone	4	-1
Camera	2.5	1
Screen	-1	-1
Prize	-1.5	-2
Design	2	-1
Processor	1.5	1
Battery life	1	1

Sedangkan untuk mendapatkan skor CSF dari kalimat superlative, dilakukan dengan langkah-langkah sebagai berikut :

1. Cek kalimat yang mengandung kata **JJS/most**, simpan dalam list superlativeSenList[] = {C3}
2. Untuk tiap kalimat superlative yang ditemukan, lakukan langkah berikut :

- Cari pasangan (lexicon+skor) – fitur produk dari masing-masing kalimat

Tabel 3.21. Pasangan Lexicon-Fitur Pada Kalimat Superlative

Kalimat	Pasangan lexicon-fitur	Skor lexicon
(C3)	best – phone	+1

- Cari keberadaan kata “*Sony Xperia U*” dalam kalimat :
 - IF **ada** AND skor lexicon = +1, THEN CSF = -1
 - ELSE IF **ada** AND skor lexicon = -1, THEN CSF = +1
 - ELSE IF **tidak ada** AND skor lexicon = +1, THEN CSF = +1
 - ELSE IF **tidak ada** AND skor lexicon = -1, THEN CSF = -1

Tabel 3.22. Nilai CSF Hasil Pengamatan Kalimat Superlative

Kalimat	Pasangan lexicon-fitur	Skor lexicon	Ada kata ?	CSF
(A2)	Best – phone	+1	Tidak ada	+1

- Update CSF yang dimiliki tiap fitur seperti pada Tabel 3.23.

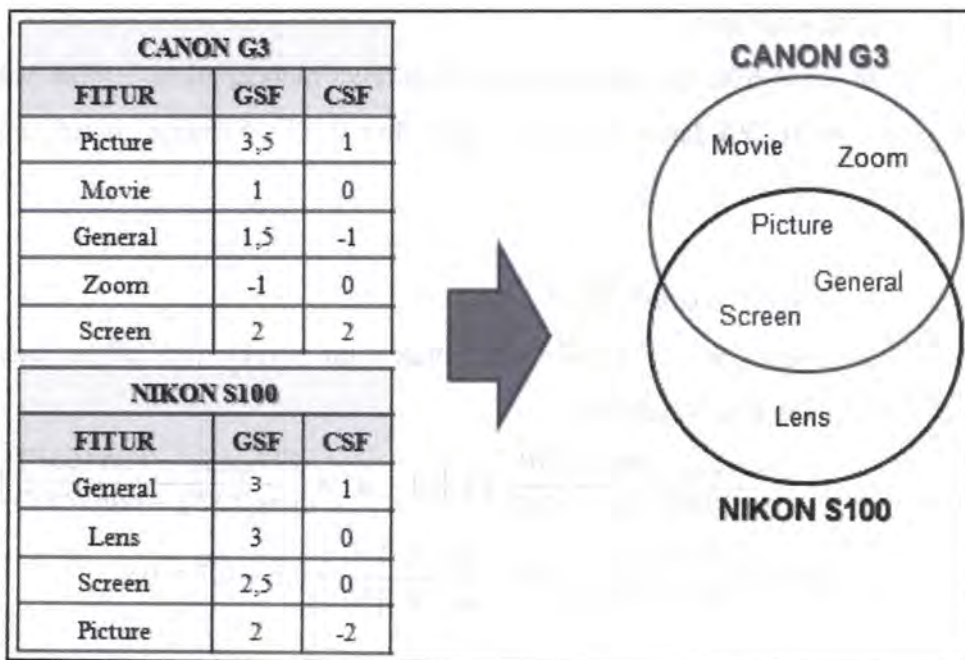
Tabel 3.23. Nilai CSF Akhir Untuk Tiap Fitur Produk

Fitur produk	GSF	CSF (vs Xperia U)
phone	4	0
camera	2.5	1
Screen	-1	-1
Prize	-1.5	-2
design	2	-1
processor	1.5	1
Battery life	1	1

3.2.4 Pengenalan Fitur Sejenis

Fitur yang telah didapatkan pada tahap sebelumnya kemudian dicari kata dasarnya. Semua fitur yang memiliki kata dasar yang sama akan dijumlahkan skor GSFnya. Berikutnya ketika produk A dibandingkan dengan produk B, fitur-fitur yang sama-sama dimiliki oleh kedua fitur tersebut akan dibandingkan, sedangkan fitur yang tidak memiliki pasangan tidak akan dibandingkan. Sementara untuk fitur yang merujuk ke produk yang sedang diamati (seperti Canon, G3, Canon G3 atau *camera*), akan dikelompokkan ke dalam satu fitur “*General*”.

Gambar 3.6 menunjukkan ilustrasi dua buah produk, yaitu Canon G3 dan Nikon S100 yang memiliki 3 fitur yang sama, yakni *picture*, *general*, dan *screen*. Maka jika kedua produk ini nantinya dibandingkan, ketiga fitur itulah yang akan menjadi fokus perbandingannya.



Gambar 3.6. Ilustrasi Fitur Produk yang Dianggap Beririsan

3.2.5 Pemberian Nilai Rekomendasi Suatu Produk

Inti dari penelitian ini adalah untuk mengetahui produk mana yang lebih unggul ketika ada dua produk yang dibandingkan. Untuk dapat memilih produk

mana yang lebih unggul, maka kedua produk harus diberikan suatu nilai. Untuk mendapatkan nilai ini, diperlukan 3 langkah sebagai berikut :

1. Hitung skor tiap fitur

Dalam tahap ini, skor GSF dan CSF digabungkan untuk mendapatkan skor akhir tiap fitur. Persamaan (3.9) adalah persamaan yang digunakan untuk menentukan nilai tersebut.

$$S_f = \left(\alpha \times \frac{GSF_f - \min(GSF)}{\max(GSF) - \min(GSF)} \right) + \left((1 - \alpha) \times \frac{CSF_f - \min(CSF)}{\max(CSF) - \min(CSF)} \right) \quad (3.9)$$

Dari persamaan (3.9) tersebut dapat dilihat bahwa untuk menghitung akhir fitur S_f untuk suatu produk, didapatkan dengan cara menjumlahkan skor GSF dan CSF dari fitur tersebut. Agar seimbang, kedua skor tersebut dinormalisasi terlebih dahulu sehingga mempunyai nilai pada range yang sama, yakni antara 0 dan 1. Sementara variabel α berisi suatu nilai yang digunakan untuk memberikan bobot lebih pada salah satu skor.

Dalam contoh kasus, tahap-tahap sebelumnya mendapatkan 2 nilai, yakni global skor fitur (GSF) dan *comparative* skor fitur (CSF). Sebagai contoh untuk fitur *screen*.

$$GSF_{screen} = 2$$

$$CSF_{screen}(CanonG3, NikonS100) = 2$$

Sehingga skor akhir produk *Canon G3* untuk fitur *screen* jika dibandingkan dengan produk *Nikon S100* adalah :

$$\begin{aligned} S_{screen} &= \left(\alpha \times \frac{GSF_{screen} - \min(GSF)}{\max(GSF) - \min(GSF)} \right) + \left((1 - \alpha) \times \frac{CSF_{screen} - \min(CSF)}{\max(CSF) - \min(CSF)} \right) \\ &= \left(0,3 \times \frac{2 - (-1)}{3,5 - (-1)} \right) + \left(0,7 \times \frac{2 - (-1)}{2 - (-1)} \right) = 0,2 + 0,7 = 0,9 \end{aligned}$$

2. Hitung bobot frekuensi tiap fitur

Setiap fitur yang telah terekstrak dihitung bobotnya dengan memanfaatkan data frekuensi terkomentarnya fitur tersebut dalam seluruh dataset kemudian dikalikan dengan probabilitas frekuensi dokumen review yang mengomentari fitur tersebut. Persamaan 3.10 menunjukkan formula untuk mendapatkan bobot tersebut.



BAB 4
HASIL DAN
PEMBAHASAN

$$bobot = (1 + \log(FF)) \times \log\left(\frac{DF}{N-DF}\right) \quad (3.10)$$

Persamaan tersebut sebenarnya mengadopsi metode *tf.prob_idf* yang pernah disebut dalam papernya Lan tentang perbandingan beberapa metode pembobotan dalam kategorisasi teks (Lan, 2004). Akan tetapi, metode pembobotan dengan mengalikan *tf* dan *idf* dirasa kurang cocok dengan topik penelitian ini. Dikarenakan dalam pengkategorisasian teks, yang ingin diekstrak adalah keyword, dimana keyword ini dapat dideteksi dengan mengamati term yang sering muncul dalam suatu dokumen, tapi jarang muncul di dokumen lain. Hal ini berbeda dengan metode indentifikasi fitur, dimana suatu term fitur produk dianggap penting ketika term tersebut sering muncul di suatu dokumen, dan sering pula muncul di dokumen yang lain. Oleh karena itu, dalam penelitian ini yang digunakan bukanlah *idf* melainkan *tf*.

3. Hitung nilai rekomendasi tiap produk

Nilai rekomendasi tiap produk dihitung dengan melihat berapa banyak fitur unggul yang dimiliki oleh produk tersebut. Artinya jika tidak ada satupun fitur unggul yang dimiliki oleh suatu produk, maka nilai rekomendasi dari produk tersebut pasti bernilai 0. Untuk mendapatkan nilai rekomendasi tersebut, berikut langkah-langkahnya :

1. Bandingkan skor fitur dari produk yang dibandingkan
2. Produk yang memiliki fitur dengan skor lebih tinggi diberikan nilai rekomendasi sesuai dengan bobot fitur tersebut
3. Lakukan langkah 1 dan 2 untuk seluruh fitur yang beririsan
4. Jumlahkan seluruh nilai rekomendasi yang didapatkan oleh setiap produk
5. Bandingkan nilai rekomendasi yang dimiliki oleh tiap fitur

Berikut adalah contoh menghitung nilai rekomendasi untuk dua buah produk yang dibandingkan.

Tabel 3.24. Contoh Bobot dan Skor Tiap Fitur Produk

Fitur	Bobot	Skor Produk 1	Skor Produk 2
Screen	3,5	0,74	0,26

Battery	2,3	0,35	0,65
Size	1,4	0,68	0,32

Dapat dilihat bahwa Produk 1 memiliki 2 fitur unggul yakni *screen* dan *size*. Sedangkan produk 2 hanya memiliki 1 fitur unggul, yakni *battery*. Oleh karena itu produk 1 akan memiliki nilai rekomendasi sebesar 4,9 dan produk 2 memiliki nilai rekomendasi sebesar 2,3. Nilai rekomendasi tersebut diperoleh dari hasil penjumlahan bobot setiap fitur unggul yang dimiliki produk tersebut. Karena nilai rekomendasi produk 1 lebih besar dari produk 2, maka produk 1 dianggap lebih baik dari pada produk 2.

3.3 Implementasi

Dalam tahap ini, diimplementasikan rancangan yang telah dibuat pada proses sebelumnya. Sistem akan dikembangkan untuk lingkungan desktop dengan menggunakan bahasa pemrograman java. Untuk *preprocessing*, *library* java dari *stanford parser* dan *stanford NLP* akan digunakan dengan terlebih dahulu meletakkan semua dataset dalam sebuah file txt.

3.4 Skenario Uji Coba

Untuk menguji keakuratan perbandingan produk yang dihasilkan oleh sistem, maka diperlukan bantuan pakar. Dalam kasus perbandingan produk, data pakar yang digunakan didapatkan dari situs *findthebest.com*. Situs tersebut menyediakan fitur ranking yang akan mengurutkan produk-produk pada domain tertentu berdasarkan data review yang diolah dari berbagai situs seperti CNET, PC Mag, Wire.com, dan PC World. Data dari pakar tersebut akan dibandingkan dengan data yang didapatkan dari system. Kemudian dihitung tingkat akurasi dengan melihat rasio banyaknya data yang sama antara system dan pakar dari keseluruhan data yang dibandingkan.

BAB 4

HASIL DAN PEMBAHASAN

4.1 Pengumpulan Dataset

Seperti dijelaskan pada bab sebelumnya, dataset yang dipergunakan dalam penelitian ini didapatkan dari situs amazon.com dengan metode *crawling*. Terdapat 2 tahapan dalam pengumpulan dataset ini. Tahap pertama adalah mendownload data review untuk setiap produk *smartphone*, *camera* dan TV. Tahap kedua adalah pembersihan tag-tag html sehingga didapatkan data yang bersih. Kemudian, pisahkan tiap review dan simpan dalam file yang berbeda. Sehingga pada akhirnya didapatkan suatu kumpulan dataset berbentuk file txt yang menyimpan satu buah review untuk tiap filenya. Tabel 4.1 menunjukkan jumlah dokumen review yang digunakan sebagai dataset.

Tabel 4.1. Dokumen Review yang Digunakan Sebagai Dataset

Domain	Jumlah produk	Jumlah Kalimat Perdokumen		Jumlah Dokumen Review
		Paling Sedikit	Paling Banyak	
Smartphone	10	1	5	1.610
Camera Digital	10	1	6	1.670
Televisi	10	1	8	1.568
Total Dokumen Review				4.848

4.2 Implementasi Sistem

Dalam sub bab ini akan dijelaskan implementasi yang dilakukan pada sistem dengan mengacu pada desain yang sudah dijelaskan pada bab sebelumnya.

4.2.1. Preprocessing

Dalam tahap ini, dataset yang terkumpul, diolah agar siap untuk dipergunakan pada tahapan selanjutnya.

1. POS Tagging

Proses ini bertujuan untuk mendapatkan informasi kedudukan kata dalam kalimat (kata benda, kata sifat, kata kerja, dan sebagainya). Untuk mendapatkan data POS tag dari dataset, digunakan *stanford-postagger* yang merupakan suatu library untuk menjalankan proses POS tagging. Library tersebut dapat diunduh secara gratis di internet (nlp.stanford.edu/software/tagger.shtml). Data POS tag yang dihasilkan kemudian disimpan dalam suatu file txt dengan nama "namafileasli-tagged.txt". Artinya, jika file asli yang diolah sebelumnya bernama "1.txt", maka dokumen hasil POS tag nya akan bernama "1-tagged.txt".

2. Dependency Parsing

Proses lainnya adalah *dependency parsing*. Tujuan dari proses ini adalah untuk mendapatkan relasi antar kata dalam suatu kalimat. Untuk menjalankan proses ini juga digunakan library dari stanford yang bernama *stanford-parser* (nlp.stanford.edu/software/lex-parser.shtml). Sama seperti library sebelumnya, library ini juga dapat diunduh secara gratis beserta dokumentasinya. Data hasil *parsing* kemudian disimpan dalam suatu file txt dengan nama "namafileasli-parsed.txt". Artinya, jika file asli yang diolah sebelumnya bernama "1.txt", maka dokumen hasil *parsing* nya akan bernama "1-parsed.txt".

Jika kedua proses tersebut berhasil dijalankan, maka dalam dataset akan terdapat 2 file tambahan yakni file tagged dan file parsed untuk tiap dokumen review. Dataset inilah yang akan dipergunakan untuk tahap-tahap berikutnya.

4.2.2. Ekstraksi dan Pembobotan Fitur Produk

Langkah pertama dalam pengekstrakan fitur produk adalah mendefinisikan kamus kata opini (*opinion lexicon*) terlebih dahulu. Dalam hal ini, kedua kamus tersebut telah didefinisikan sebelumnya, kemudian disimpan dalam file *positive-words.txt* (untuk menyimpan kata opini positif) dan file *negative-words.txt* (untuk menyimpan kata opini negatif).


```

1. //Mendefinisikan kamus kata opini positif
2. br = new BufferedReader(new FileReader("data/positive-words.txt"));
3. String line;
4. while ((line = br.readLine()) != null) {
5.     StaticVariable.lexicon.put(line, 1.0);
6. }
7. br.close();
8. //Mendefinisikan kamus kata opini negatif
9. bl = new BufferedReader(new FileReader("data/negative-words.txt"));
10. String line;
11. while ((line = bl.readLine()) != null) {
12.     StaticVariable.lexicon.put(line, -1.0);
13. }
14. bl.close();

```

Gambar 4.1. Potongan Kode Pendefinisikan Kamus Kata

Dalam Gambar 4.1, baris 1-7 digunakan untuk membaca kata-kata yang masuk dalam kata opini positif, sedangkan baris 8-14 digunakan untuk membaca kata-kata yang tergolong kata opini negatif. Sedangkan baris ke-5 dan ke-12 digunakan untuk memberi polaritas pada kata opini positif (ditandai dengan skor 1) dan kata opini negatif (ditandai dengan skor -1).

Tahap berikutnya adalah membaca dokumen tagger untuk mengambil semua *noun* dan *adjective* yang terdapat dalam dataset dan simpan dalam suatu list seperti yang ditunjukkan oleh Gambar 4.2 berikut pada baris 7-15.

```

1. br = new BufferedReader(new FileReader(path + "/" + nomer + "-tagged.txt"));
2. String line;
3. while ((line = br.readLine()) != null) {
4.     String kata2[] = line.split(" ");
5.     for (String kata : kata2) {
6.         String type_kata[] = kata.split("_");
7.         if (type_kata[1].startsWith("NN")) {
8.             if (!StaticVariable.noun.contains(type_kata[0].trim().toLowerCase())) {
9.                 StaticVariable.noun.add(type_kata[0].trim().toLowerCase());
10.            }
11.        } else if (type_kata[1].startsWith("JJ")) {
12.            if (!StaticVariable.adj.contains(type_kata[0].trim().toLowerCase())) {
13.                StaticVariable.adj.add(type_kata[0].trim().toLowerCase());
14.            }
15.        }
16.    }
17. }
18. br.close();

```

Gambar 4.2. Potongan Kode Pembacaan Dokumen Tagged

Hal ini dilakukan agar ketika proses MDP berjalan, system tinggal melihat kedalam list dan tidak perlu lagi membaca dokumen tagger hanya untuk memastikan apakah kata yang sedang dianalisa merupakan kata benda (*noun*) atau kata sifat (*adjective*), dimana proses tersebut akan memakan waktu komputasi yang cukup besar jika seringkali dilakukan.

Kemudian, dokumen parsed dari suatu dokumen review juga harus dibaca. Langkah ini dilakukan untuk mendapatkan semua kalimat dalam dokumen review serta menyimpan semua relasi antar kata dalam kalimat tersebut. Relasi-relasi inilah yang nanti akan dianalisa oleh metode MDP.

```
1. br = new BufferedReader(new FileReader(path + "/" + nomer + "-parsed.txt"));
2. String line;
3. while ((line = br.readLine()) != null) {
4.     if (line.startsWith("===")) {
5.         index++;
6.     }
7.     if (line.indexOf("(") > 0) {
8.         StaticVariable.tagkata.add(new Tagged(line, nomer, index));
9.         ParsedModel parsedModel = new ParsedModel();
10.        parsedModel.kata = line;
11.        parsedModel.nomer = nomer;
12.        parsedModel.index = index;
13.        parsedModel.path = path + "/" + nomer + ".txt";
14.        ls.add(parsedModel);
15.    }
16. }
17. br.close();
```

Gambar 4.3. Potongan Kode Pembacaan Dokumen Parsed

Dapat dilihat pada Gambar 4.3 di baris 10-14, bahwa selain relasi antar kata, urutan relasi serta lokasi relasi dalam suatu dokumen juga harus disimpan. Ini perlu dilakukan mengingat metode MDP akan melakukan proses analisa secara berulang. Oleh karena itu lokasi suatu relasi dalam dokumen juga perlu disimpan agar MDP dapat langsung menuju relasi tersebut jika ada sesuatu yang harus diupdate pada setiap iterasinya.

4.2.2.1. Pendefinisian Rule

Seperti yang telah dijelaskan dalam bab sebelumnya, metode MDP adalah metode rule based. Sehingga rule-rule yang akan dipergunakan harus

didefinisikan terlebih dahulu. Keenam rule dalam metode MDP diimplementasikan dalam kode seperti berikut.

- R1 (penanganan relasi *amod(nn,adj)*)

```
1. if (type.equalsIgnoreCase("amod")) {
2.     ReturnData returnData = chekNounAdj(kataPertama, kataKedua);
3.     if (returnData.hasil) {
4.         kataPertama = returnData.kata1;
5.         kataKedua = returnData.kata2;
6.         FiturProduk fiturProduk = new FiturProduk(kataPertama, model.nomer,
7.             model.path, model.index);
8.         if (fitur_produk.containsKey(fiturProduk)) {
9.             if (!StaticVariable.lexicon.containsKey(kataKedua)) {
10.                double nilaidef = polaritasProduk(fitur_produk.get(fiturProduk));
11.                StaticVariable.lexicon.put(kataKedua, nilaidef);
12.                double nilaisebelumnya = fitur_produk.get(fiturProduk);
13.                fitur_produk.put(fiturProduk, nilaisebelumnya
14.                    + polaritasProdukSvSn(kataKedua, model.nomer));
15.                addJmlFitur(fiturProduk.nama);
16.                proses = true;
17.                model.flag = true;
18.            }
19.        } else if (StaticVariable.lexicon.containsKey(kataKedua)) {
20.            if (!fitur_produk.containsKey(fiturProduk)) {
21.                fitur_produk.put(fiturProduk, polaritasProdukSvSn(kataKedua,
22.                    model.nomer));
23.                addJmlFitur(fiturProduk.nama);
24.                proses = true;
25.                model.flag = true;
26.            } else {
27.                double nilaisebelumnya = fitur_produk.get(fiturProduk);
28.                fitur_produk.put(fiturProduk, nilaisebelumnya
29.                    + polaritasProdukSvSn(kataKedua, model.nomer));
30.                addJmlFitur(fiturProduk.nama);
31.                proses = true;
32.                model.flag = true;
33.            }
34.        }
35.    }
36. }
```

Gambar 4.4. Potongan Kode Implementasi R1

Pada Gambar 4.4 di baris 20-33, dapat dilihat bahwa terdapat 2 kondisi untuk penanganan relasi *amod*. Kondisi pertama adalah ketika fitur produk atau kata opini sudah berhasil diekstrak pada iterasi sebelumnya, sehingga system hanya perlu mengubah bobot dari fitur berdasarkan persamaan yang telah dijelaskan pada bab sebelumnya. Kondisi kedua terjadi ketika fitur produk ataupun kata opini belum ada dalam list, yang menandakan bahwa kata tersebut belum pernah terekstrak sebelumnya. Untuk menagani kondisi ini, maka kata tersebut akan ditambahkan ke dalam list dan set bobotnya berdasarkan persamaan yang sudah didefinisikan sebelumnya.

- R2 dan R6 (penanganan relasi *nsubj(adj,nn)* dan *prep_with(adj,nn)*)

Karena relasi *nsubj* dan *prep_with* punya struktur kondisi dan penanganan yang hampir sama, maka kedua jenis relasi ini ditangani oleh satu buah kondisi seperti yang ditunjukkan oleh kode pada Gambar 4.5 di baris ke-1.

```

1. if ((type.equalsIgnoreCase("nsubj") || type.equalsIgnoreCase("prep_with"))) {
2.     ReturnData returnData = cekAdjNoun(kataPertama, kataKedua);
3.     if (returnData.hasil) {
4.         kataPertama = returnData.kata1;
5.         kataKedua = returnData.kata2;
6.         FiturProduk fiturProduk = new FiturProduk(kataKedua, model.nomer,
7.             model.path, model.index);
8.         if (fitur_produk.containsKey(fiturProduk)) {
9.             if (!StaticVariable.lexicon.containsKey(kataPertama)) {
10.                double nilaidef = polaritasProduk(fitur_produk.get(fiturProduk));
11.                StaticVariable.lexicon.put(kataPertama, nilaidef);
12.                double nilaisebelumnya = fitur_produk.get(fiturProduk);
13.                fitur_produk.put(fiturProduk, nilaisebelumnya
14.                    + polaritasProdukSvSn(kataPertama,
15.                        model.nomer));
16.                addJmlFitur(fiturProduk.nama);
17.                proses = true;
18.                model.flag = true;
19.            }
20.        } else if (StaticVariable.lexicon.containsKey(kataPertama)) {
21.            if (!fitur_produk.containsKey(fiturProduk)) {
22.                fitur_produk.put(fiturProduk, polaritasProdukSvSn(kataPertama,
23.                    model.nomer));
24.                addJmlFitur(fiturProduk.nama);
25.                proses = true;
26.                model.flag = true;
27.            } else {
28.                double nilaisebelumnya = fitur_produk.get(fiturProduk);
29.                fitur_produk.put(fiturProduk, nilaisebelumnya
30.                    + polaritasProdukSvSn(kataPertama,
31.                        model.nomer));
32.                addJmlFitur(fiturProduk.nama);
33.                proses = true;
34.                model.flag = true;
35.            }
36.        }
37.    }
38. }

```

Gambar 4.5. Potongan Kode Implementasi R2

- R3 (penanganan relasi *conj_and(adj,adj)*)

Terdapat dua jenis relasi *conj_and* yang ditangani. Gambar 4.6 adalah contoh script penanganan relasi *conj_and* untuk mengidentifikasi dan mengekstrak kata opini yang terdapat dalam suatu kalimat. Hal ini dapat dilihat dari fungsi cek relasi yang dipanggil yaitu fungsi *cekAdjAdj*, seperti yang terdapat pada baris ke-2 pada gambar tersebut.

```

1.  if (type.equalsIgnoreCase("conj_and")) {
2.      ReturnData returnData = chekAdjAdj(kataPertama, kataKedua);
3.      if (returnData.hasil) {
4.          StaticVariable.ruleConjAndAdj++;
5.          kataPertama = returnData.kata1;
6.          kataKedua = returnData.kata2;
7.          if (StaticVariable.lexicon.containsKey(kataPertama)) {
8.              if (!StaticVariable.lexicon.containsKey(kataKedua)) {
9.                  StaticVariable.lexicon.put(kataKedua,
10.                     StaticVariable.lexicon.get(kataPertama));
11.                  StaticVariable.parsedFilter.add(model);
12.                  proses = true;
13.                  model.flag = true;
14.              }
15.          } else if (StaticVariable.lexicon.containsKey(kataKedua)) {
16.              if (!StaticVariable.lexicon.containsKey(kataPertama)) {
17.                  StaticVariable.lexicon.put(kataPertama,
18.                     StaticVariable.lexicon.get(kataKedua));
19.                  StaticVariable.parsedFilter.add(model);
20.                  proses = true;
21.                  model.flag = true;
22.              }
23.          }
24.      }
25.  }

```

Gambar 4.6. Potongan Kode Implementasi R3

- R4 (penanganan relasi *conj_and(nn,nn)*)

Sedangkan relasi *conj_and* yang ditangani oleh rule ini adalah untuk mengidentifikasi dan mengekstrak fitur produk yang berupa *noun* dalam suatu kalimat opini. Hal ini dapat dilihat dari fungsi cek relasi yang dipanggil yaitu fungsi *chekNounNoun*, seperti yang terdapat pada baris ke-2 pada Gambar 4.7. Pada gambar tersebut juga dapat diamati bahwa terdapat dua kondisi yang menangani kemunculan kata benda (baris ke-10 dan baris ke-25). Maksud dari kedua kondisi tersebut adalah, jika kata benda pertama sudah terdapat pada daftar fitur produk, berarti kata benda tersebut sudah terdaftar sebagai fitur, sehingga kata kedua yang diekstrak untuk ditambahkan ke dalam daftar fitur produk (baris 10-24). Sebaliknya jika kata kedua yang sudah terdapat dalam daftar fitur produk, maka kata pertama lah yang akan diekstrak untuk ditambahkan ke dalam daftar fitur produk (baris 25-40).

```

1.  if (type.equalsIgnoreCase("conj_and")) {
2.      ReturnData returnData = cekNounNoun(kataPertama, kataKedua);
3.      if (returnData.hasil) {
4.          kataPertama = returnData.kata1;
5.          kataKedua = returnData.kata2;
6.          FiturProduk fiturProduk1 = new FiturProduk(kataPertama, model.nomer,
7.              model.path, model.index);
8.          FiturProduk fiturProduk2 = new FiturProduk(kataKedua, model.nomer,
9.              model.path, model.index);
10.         if (fitur_produk.containsKey(fiturProduk1)) {
11.             if (!fitur_produk.containsKey(fiturProduk2)) {
12.                 double nilaidef = polaritasProduk(fitur_produk.get(fiturProduk1));
13.                 fitur_produk.put(fiturProduk2, nilaidef);
14.                 addJmlFitur(fiturProduk2.nama);
15.                 proses = true;
16.                 model.flag = true;
17.             } else {
18.                 double nilaidef = polaritasProduk(fitur_produk.get(fiturProduk1));
19.                 double nilaisebelumnya = fitur_produk.get(fiturProduk2);
20.                 fitur_produk.put(fiturProduk2, nilaisebelumnya + nilaidef);
21.                 addJmlFitur(fiturProduk2.nama);
22.                 proses = true;
23.                 model.flag = true;
24.             }
25.         } else if (fitur_produk.containsKey(fiturProduk2)) {
26.             if (!fitur_produk.containsKey(fiturProduk1)) {
27.                 double nilaidef = polaritasProduk(fitur_produk.get(fiturProduk2));
28.                 fitur_produk.put(fiturProduk1, nilaidef);
29.                 addJmlFitur(fiturProduk1.nama);
30.                 proses = true;
31.                 model.flag = true;
32.             } else {
33.                 double nilaidef = polaritasProduk(fitur_produk.get(fiturProduk2));
34.                 double nilaisebelumnya = fitur_produk.get(fiturProduk1);
35.                 fitur_produk.put(fiturProduk1, nilaisebelumnya + nilaidef);
36.                 addJmlFitur(fiturProduk1.nama);
37.                 proses = true;
38.                 model.flag = true;
39.             }
40.         }
41.     }
42. }

```

Gambar 4.7. Potongan Kode Implementasi R4

- R5 (penanganan relasi *det(no,nn)*)

Rule ini adalah salah satu rule yang ditambahkan dalam metode MDP. Rule ini akan menangani fitur produk yang muncul dalam kalimat yang tidak memiliki kata opini. Indikator yang digunakan untuk mengekstrak fitur produk adalah keberadaan relasi *det* antara *noun* dan kata *no*.

Metode *prunning* yang dilakukan adalah dengan mencari frekuensi kemunculan fitur dalam keseluruhan dataset. Kemudian fitur-fitur tersebut diranking dari yang paling sering muncul sampai yang jarang muncul. Langkah terakhir adalah membuang fitur-fitur yang memiliki frekuensi kemunculan kurang dari *threshold* yang ditentukan. *Threshold* adalah suatu nilai yang digunakan untuk membatasi apakah suatu fitur dianggap penting, sehingga layak untuk diperhatikan atau tidak. Nilai *threshold* ini akan dibandingkan dengan jumlah frekuensi terkomentarnya suatu fitur. Misalkan nilai *threshold* diset sama dengan 80, maka fitur-fitur yang memiliki frekuensi terkomentari lebih dari 80 kali dianggap fitur yang penting. Dalam proses berikutnya, hanya fitur-fitur inilah yang diperhatikan oleh sistem, sedangkan fitur-fitur yang lain akan diabaikan. Hal ini dilakukan agar fitur-fitur yang tidak seberapa penting (jarang terkomentari) tidak menjadi *noise* dalam proses perbandingan produk, karena biasanya fitur-fitur relevan memiliki frekuensi terkomentari yang lebih tinggi dibanding dengan fitur yang tidak relevan. Penentuan *threshold* yang paling optimal akan diujikan pada sub-bab berikutnya.

4.2.3. Pengenalan dan Pembobotan Kalimat Comparative

Pengenalan kalimat *comparative* dilakukan dengan mengidentifikasi kemunculan kata *adjective comparative* dalam suatu kalimat. Hal ini bisa dilakukan dengan menganalisa kata yang memiliki tag JJR dalam dokumen tagged. Selain itu, pengenalan kalimat *comparative* juga dapat dilakukan dengan mengidentifikasikemunculan kata “*more*” dalam kalimat. Kedua jenis kata ini seringkali digunakan untuk menyusun suatu kalimat *comparative*. Untuk kata dengan tag JJR biasanya juga merupakan kata opini. Sedangkan untuk kata *more*, biasanya diikuti dengan kata *adjective* yang berfungsi sebagai kata opini. Kedua kondisi tersebut diimplementasikan dalam baris 3-10 pada Gambar 4.9.

```

1.  if (type.equalsIgnoreCase("det")) {
2.      ReturnData returnData = chekNoNoun(kataPertama, kataKedua);
3.      if (returnData != null && returnData.hasil) {
4.          kataPertama = returnData.kata1;
5.          kataKedua = returnData.kata2;
6.          FiturProduk fiturProduk = new FiturProduk(kataPertama, model.ncmer,
7.              model.path, model.index);
8.          if (!fitur_produk.containsKey(fiturProduk)) {
9.              if (kataPertama.equalsIgnoreCase("problem") ||
10.                 kataPertama.equalsIgnoreCase("complaint")) {
11.              } else {
12.                  fitur_produk.put(fiturProduk, -1.0);
13.                  addJmlFitur(fiturProduk.nama);
14.              }
15.              prosess = true;
16.              model.flag = true;
17.          } else {
18.              double nilaisebelumnya = fitur_produk.get(fiturProduk);
19.              if (kataPertama.equalsIgnoreCase("problem") ||
20.                 kataPertama.equalsIgnoreCase("complaint")) {
21.              } else {
22.                  fitur_produk.put(fiturProduk, nilaisebelumnya - 1.0);
23.                  addJmlFitur(fiturProduk.nama);
24.              }
25.              prosess = true;
26.              model.flag = true;
27.          }
28.      }
29.  }
30. }

```

Gambar 4.8. Potongan Kode Implementasi R5

Akan tetapi, tidak semua *noun* yang berpasangan dengan kata *no* adalah fitur produk, ada beberapa kata yang seringkali digunakan tapi bukanlah fitur produk, kata tersebut diantaranya adalah kata *problem* dan *complaint*. Oleh karena itu, kedua kata tersebut dimasukkan dalam kondisi pengecualian, seperti yang ditunjukkan oleh baris 9-10 serta baris 19-20 pada Gambar 4.8.

4.2.2.2. Prunning Fitur

Fitur yang berhasil diekstrak oleh metode MDP memiliki banyak *noise*. Hal ini terjadi karena kesalahan proses parsing ataupun karena kondisi dataset. Kesalahan parsing terjadi biasanya dikarenakan pemakaian kalimat yang tidak baku. Sedangkan kondisi dataset yang dapat menyebabkan terjadinya *noise* adalah penggunaan istilah-istilah tertentu yang digunakan untuk merujuk pada suatu fitur. Misal kata "*thing*" atau "*one*" yang seringkali digunakan untuk merujuk pada suatu fitur tertentu.


```

1. for (int x = 0; x < perkata.length; x++) {
2.     String katachek = cekKalimatType(perkata[x], "jJR", false);
3.     if (perkata[x].toLowerCase().startsWith("more")) {
4.         lkmCom.add(new lexicon_kata_mengandung(cekKalimatType(perkata[x + 1],
5.             null, true), kalimat.dokumen, kalimat.kalimatke, string1));
6.     }
7.     if (!katachek.equalsIgnoreCase(perkata[x])) {
8.         lkmCom.add(new lexicon_kata_mengandung(katachek, kalimat.dokumen,
9.             kalimat.kalimatke, string1));
10.    }
11. }

```

Gambar 4.9. Potongan Kode Identifikasi Kalimat Superlative

Berikutnya, untuk melakukan pembobotan pada kalimat tersebut, harus diketahui terlebih dahulu nama produk yang akan dibandingkan dengan produk yang sedang diamati. Dalam penelitian ini, produk lain yang coba dibandingkan dengan produk yang diamati diistilahkan dengan *enemy*. Kemudian, lokasi dari kata *enemy* tersebut diamati apakah terdapat di sebelah kiri (Gambar 4.10 baris 22-28) ataukah di sebelah kanan (Gambar 4.10 baris 29-36) dari kata opini yang telah ditemukan. Letak kata ini akan menentukan bobot dari kalimat tersebut.

```

21. if (typeKalimat == TypeKalimat.Comparative) {
22.     if (indexProduk < indexKata1) {
23.         kirikanan = "kiri";
24.         if (enemy == true) {
25.             positiveNegative = -2;
26.         } else {
27.             positiveNegative = 2;
28.         }
29.     } else {
30.         kirikanan = "kanan";
31.         if (enemy == true) {
32.             positiveNegative = 2;
33.         } else {
34.             positiveNegative = -2;
35.         }
36.     }
37. }

```

Gambar 4.10. Potongan Kode Penanganan Kalimat Comparative

Selain kalimat *comparative*, terdapat jenis kalimat lain yang memiliki fungsi hampir sama dengan kalimat *comparative*, yaitu jenis kalimat *superlative*. Hampir sama dengan metode untuk mengidentifikasi kalimat *comparative*, untuk mengidentifikasi kalimat *superlative* kata yang dijadikan sebagai indikator adalah kata *adjective superlative* atau kata dengan tag JJS dan kata *most*. Kedua indikator

tersebut didefinisikan sebagai 2 kondisi yang berbeda. Dimana kondisi *most* didefinisikan oleh baris 43-46, dan kondisi JJS didefinisikan oleh baris 47-50 pada Gambar 4.11.

```
41. for (int x = 0; x < perkata.length; x++) {
42.     String katacek = cekKalimatType(perkata[x], "jjs", false);
43.     if (perkata[x].toLowerCase().startsWith("most")) {
44.         lkmSup.add(new lexicon_kata_mengandung(cekKalimatType(perkata[x + 1],
45.             null, true), kalimat.dokumen, kalimat.kalimatke));
46.     }
47.     if (!katacek.equalsIgnoreCase(perkata[x])) {
48.         lkmSup.add(new lexicon_kata_mengandung(katacek, kalimat.dokumen,
49.             kalimat.kalimatke));
50.     }
51. }
```

Gambar 4.11. Potongan Kode Identifikasi Kalimat Superlative

Sedangkan untuk memberikan bobot pada kalimat jenis *superlative* ini, letak produk yang berfungsi sebagai *enemy* tidak terlalu diperlukan. Hal ini dikarenakan kalimat jenis ini biasanya hanya mengandung satu produk saja. Sehingga yang perlu diperhatikan hanya ada atau tidaknya produk lain yang muncul dalam kalimat tersebut. Logikanya, jika kata opini yang berhasil dideteksi adalah jenis kata opini positif, dan muncul nama dari produk lain (selain produk yang sedang diamati) dalam kalimat tersebut, maka dapat disimpulkan bahwa bobot dari kalimat ini adalah negatif terhadap produk yang sedang diamati, dan positif terhadap produk lainnya, dan begitu pula sebaliknya. Baris 55-57 pada gambar 4.12 menangani kondisi jika produk *enemy* muncul dalam kalimat, sedangkan baris 58-60 menangani kondisi sebaliknya.

```
55. if (indexProduk > 0) {
56.     kirikanan = "ada";
57.     positiveNegative = -1;
58. } else if (indexProduk < 0) {
59.     kirikanan = "tidak ada";
60.     positiveNegative = 1;
61. }
```

Gambar 4.12. Potongan Kode Penanganan Kalimat Superlative

4.2.4. Pengenalan Fitur Sejenis

Agar dapat dibandingkan secara adil, maka kedua produk harus memiliki fitur dengan jumlah dan nama yang sama. Oleh karena itu, dalam penelitian ini,

fitur yang dibandingkan hanya fitur-fitur yang beririsan atau dimiliki oleh kedua produk.

```
1. for (Map.Entry<String, nilai_gsf_csf> entry : tempProd1.entrySet()) {
2.     String key = entry.getKey();
3.     nilai_gsf_csf value = entry.getValue();
4.     if (prod2.containsKey(key)) {
5.         nilai_gsf_csf valueAsli = prod2.get(key);
6.         prod2.put(key, new nilai_gsf_csf(valueAsli.gsf,
7.             valueAsli.csf - value.csf));
8.     }
9. }
```

Gambar 4.12. Potongan Kode Pengenalan Fitur Sejenis

Baris 4-7 pada Gambar 4.12 adalah baris kode untuk penanganan penyesuaian skor CSF pada tiap fitur yang beririsan. Hal ini dikarenakan skor CSF difungsikan untuk menghilangkan independensi penilaian untuk tiap produk. Oleh karena itu, perlu ada penyesuaian sedikit di bagian ini. Ketika ada fitur dari produk 1 yang memiliki skor CSF +3, maka seharusnya fitur yang sama di produk 2 harus memiliki skor -3. Karena ketika seseorang mengatakan produk A lebih baik dari produk B, pada dasarnya dia memberi skor +1 untuk produk A, dan di saat yang sama memberi skor -1 untuk produk B.

4.2.5. Pemberian Nilai Rekomendasi Untuk Tiap Produk

Cara memberi nilai rekomendasi untuk tiap produk telah dijelaskan pada bab sebelumnya. Dalam implementasinya, terdapat dua mekanisme pemberian nilai rekomendasi. Yakni dengan memanfaatkan kamus kata dan tanpa menggunakan kamus kata. Perbedaan dari keduanya adalah jika menggunakan kamus kata, maka fitur-fitur yang tidak terdapat dalam kamus kata akan diabaikan dalam proses penghitungan nilai rekomendasi. Sedangkan jika kamus kata tidak dimanfaatkan, maka semua fitur yang beririsan akan diperhatikan dalam proses penentuan nilai rekomendasi.

Kamus kata yang digunakan disini berisi daftar fitur relevan dari suatu domain produk. Daftar fitur ini didapatkan melalui penyebaran kuisioner pada 30 responden dari beragam latar belakang. Responden diminta untuk memilih fitur yang dianggap relevan dari serangkaian fitur yang berhasil diekstrak oleh metode

MDP. Fitur yang dipilih oleh 51% responden, dianggap sebagai fitur yang relevan, dan dimasukkan dalam kamus kata.

Penggunaan kamus kata ini hanya untuk memastikan apakah fitur-fitur yang dianggap tidak relevan memberikan gangguan yang signifikan terhadap hasil perbandingan yang dilakukan oleh system.

4.3. Hasil Uji Coba

Terdapat 2 uji coba yang akan dilakukan pada sistem, yaitu uji metode *Modification Double Propagation* (MDP) yang digunakan sebagai ekstraktor fitur dan uji akurasi nilai rekomendasi untuk tiap pasangan produk yang dibandingkan.

4.3.1. Uji Metode MDP

Untuk menguji apakah tiap rule yang ada dalam metode MDP sudah memenuhi ekspektasi yang diharapkan, maka perlu diuji kemampuan *coverage* dan akurasi dari tiap rule. Cara untuk menguji kedua nilai tersebut telah dijelaskan pada bab sebelumnya. Tabel 4.2 dan 4.3 menunjukkan hasil uji coba yang telah dilakukan pada tiap rule. Dari tabel tersebut dapat diamati bahwa R5 dan R6, yang merupakan rule baru yang ditambahkan pada metode DP, memiliki nilai akurasi yang sama dengan rule yang lain.

Tabel 4.2. Tabel Uji Coverage Untuk Tiap Rule MDP

Domain	Coverage Rule					
	R1	R2	R3	R4	R5	R6
Smartphone	0,501	0,148	0,065	0,216	0,043	0,026
Camera Digital	0,523	0,132	0,079	0,218	0,022	0,025
Televisi	0,507	0,167	0,070	0,192	0,036	0,028

Tabel 4.3. Tabel Uji Akurasi Untuk Tiap Rule MDP

Domain	Akurasi Rule					
	R1	R2	R3	R4	R5	R6
Smartphone	0,097	0,148	0,735	0,224	0,145	0,263
Camera Digital	0,079	0,098	0,615	0,156	0,159	0,212
Televisi	0,096	0,099	0,655	0,145	0,149	0,234

didapatkan oleh sistem yang tanpa menggunakan kamus kata. Padahal secara logika, seharusnya sistem yang tidak menggunakan kamus kata akan memiliki banyak noise, sehingga sulit untuk bisa mengalahkan sistem yang menggunakan kamus kata. Setelah dilakukan proses analisa, ternyata permasalahan muncul dikarenakan ada beberapa fitur relevan yang tidak masuk ke dalam kamus kata relevan. Hal ini bisa dimaklumi karena penyusunan kamus kata dilakukan dengan cara kuisisioner. Dimana seluruh responden berasal dari Indonesia, sedangkan dataset adalah data berbahasa Inggris. Mungkin karena perbedaan budaya, atau responden tidak membaca penggunaan kata tersebut dalam kalimat opini yang utuh, sehingga mereka tidak memilih kata tersebut sebagai kata relevan. Padahal kata-kata tersebut jika dimasukkan ke kamus kata, akan meningkatkan akurasi dari system dengan sangat signifikan. Setelah dianalisa, ada 2 kata yang harus dimasukkan dalam kamus kata relevan agar bisa meningkatkan akurasi sistem yang menggunakan kamus kata, yakni kata *time* dan *product*. Setelah kedua kata ini dimasukkan, maka nilai akurasi tertinggi didapatkn oleh system yang menggunakan kamus kata seperti yang dapat dilihat pada tabel 4.10.

Tabel 4.10. Akurasi Untuk Domain Smartphone (withKamus) pasca analisa

T	Alpha										
	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
10	0,89	0,60	0,58	0,53	0,53	0,56	0,56	0,49	0,51	0,56	0,42
20	0,89	0,69	0,64	0,64	0,64	0,64	0,64	0,64	0,64	0,64	0,56
30	0,89	0,73	0,71	0,71	0,69	0,69	0,69	0,69	0,69	0,67	0,60
40	0,89	0,82	0,84	0,84	0,80	0,73	0,69	0,69	0,67	0,67	0,69
50	0,89	0,82	0,82	0,82	0,80	0,78	0,76	0,76	0,76	0,71	0,69
60	0,89	0,80	0,78	0,80	0,78	0,73	0,73	0,76	0,76	0,73	0,67
70	0,89	0,91	0,93	0,91	0,91	0,87	0,84	0,82	0,82	0,80	0,78
80	0,89	0,89	0,93	0,93	0,91	0,87	0,84	0,82	0,82	0,80	0,78
90	0,89	0,91	0,96	0,96	0,91	0,89	0,89	0,87	0,82	0,80	0,78
100	0,89	0,91	0,96	0,96	0,91	0,89	0,87	0,84	0,78	0,73	0,71

Sedangkan untuk kedua domain yang lain, yakni domain camera digital dan televisi, dapat dilihat pada Tabel 4.11 sampai 4.14.

Tabel 4.11. Tabel Akurasi Untuk Domain Camera Digital (withKamus)

T	Alpha										
	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
10	0,36	0,36	0,36	0,33	0,33	0,31	0,31	0,38	0,40	0,40	0,25
20	0,36	0,36	0,33	0,31	0,31	0,29	0,27	0,36	0,38	0,38	0,21
30	0,40	0,40	0,40	0,38	0,38	0,38	0,38	0,44	0,42	0,42	0,32
40	0,51	0,51	0,51	0,47	0,44	0,38	0,44	0,51	0,51	0,58	0,43
50	0,44	0,44	0,44	0,44	0,42	0,40	0,36	0,42	0,47	0,44	0,40
60	0,51	0,51	0,51	0,44	0,42	0,44	0,42	0,44	0,44	0,42	0,42
70	0,53	0,53	0,56	0,58	0,53	0,51	0,56	0,56	0,53	0,53	0,56
80	0,56	0,56	0,58	0,60	0,58	0,51	0,56	0,56	0,56	0,53	0,51
90	0,58	0,58	0,58	0,62	0,60	0,60	0,64	0,60	0,58	0,58	0,56
100	0,58	0,58	0,58	0,62	0,60	0,60	0,64	0,60	0,58	0,58	0,56

Tabel 4.12. Tabel Akurasi Untuk Domain Camera Digital (withoutKamus)

T	Alpha										
	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
10	0,49	0,49	0,51	0,51	0,51	0,51	0,51	0,51	0,51	0,51	0,51
20	0,47	0,47	0,47	0,49	0,49	0,49	0,49	0,49	0,49	0,49	0,46
30	0,47	0,47	0,47	0,51	0,53	0,53	0,56	0,56	0,53	0,56	0,53
40	0,44	0,44	0,47	0,47	0,49	0,47	0,51	0,53	0,53	0,53	0,45
50	0,47	0,47	0,47	0,44	0,44	0,49	0,51	0,56	0,53	0,51	0,39
60	0,42	0,42	0,42	0,40	0,42	0,47	0,51	0,49	0,49	0,47	0,38
70	0,47	0,47	0,47	0,42	0,44	0,44	0,49	0,49	0,47	0,44	0,37
80	0,49	0,49	0,49	0,51	0,51	0,51	0,56	0,56	0,53	0,53	0,49
90	0,56	0,66	0,56	0,60	0,62	0,60	0,60	0,60	0,58	0,58	0,56
100	0,56	0,56	0,56	0,60	0,62	0,60	0,60	0,60	0,58	0,58	0,56

Tabel 4.13. Tabel Akurasi Untuk Domain Televisi (withKamus)

T	Alpha										
	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
10	0,51	0,56	0,56	0,53	0,49	0,49	0,51	0,51	0,51	0,51	0,48
20	0,51	0,53	0,53	0,51	0,49	0,49	0,51	0,51	0,51	0,51	0,48
30	0,51	0,53	0,53	0,51	0,51	0,53	0,53	0,51	0,51	0,51	0,49
40	0,53	0,53	0,56	0,53	0,53	0,53	0,53	0,53	0,53	0,53	0,49
50	0,58	0,51	0,53	0,53	0,56	0,49	0,51	0,53	0,58	0,58	0,49
60	0,60	0,62	0,64	0,64	0,62	0,60	0,58	0,64	0,60	0,60	0,53
70	0,62	0,71	0,69	0,64	0,64	0,60	0,60	0,64	0,62	0,62	0,56
80	0,76	0,60	0,60	0,69	0,67	0,67	0,67	0,64	0,76	0,78	0,56

90	0,60	0,76	0,78	0,69	0,67	0,67	0,67	0,69	0,60	0,60	0,56
100	0,60	0,78	0,80	0,62	0,67	0,67	0,67	0,62	0,60	0,60	0,58

Tabel 4.14. Tabel Akurasi Untuk Domain Televisi (withoutKamus)

T	Alpha										
	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
10	0,51	0,51	0,51	0,51	0,51	0,51	0,51	0,51	0,51	0,51	0,48
20	0,53	0,53	0,53	0,51	0,53	0,51	0,51	0,51	0,51	0,51	0,48
30	0,51	0,51	0,51	0,49	0,51	0,49	0,49	0,49	0,49	0,49	0,48
40	0,53	0,53	0,53	0,51	0,53	0,49	0,49	0,49	0,49	0,49	0,48
50	0,44	0,44	0,44	0,44	0,44	0,44	0,44	0,44	0,44	0,44	0,44
60	0,49	0,56	0,56	0,49	0,49	0,49	0,47	0,47	0,47	0,47	0,49
70	0,53	0,56	0,56	0,49	0,53	0,49	0,49	0,49	0,51	0,51	0,48
80	0,55	0,67	0,67	0,58	0,60	0,56	0,58	0,56	0,56	0,56	0,51
90	0,56	0,62	0,64	0,56	0,58	0,51	0,53	0,49	0,51	0,51	0,48
100	0,58	0,78	0,78	0,67	0,67	0,67	0,62	0,67	0,60	0,60	0,56

Dari hasil ujicoba akurasi yang diujikan pada ketiga domain tersebut dapat diambil kesimpulan bahwa penambahan skor CSF ternyata mampu meningkatkan akurasi jika dibandingkan dengan sistem yang hanya menggunakan skor GSF seperti yang pernah diusulkan oleh Liu (Liu, 2005). Untuk lebih jelasnya bisa dilihat pada Tabel 4.15.

Tabel 4.15. Tabel Perbandingan Untuk Skor GSF dan GSF+CSF

Domain	Akurasi Tertinggi	
	GSF	GSF+CSF
Smartphone	0,71	0,96
Camera Digital	0,56	0,64
Televisi	0,58	0,80

Untuk menguji apakah terdapat perbedaan yang signifikan antara akurasi yang didapatkan oleh sistem yang menggunakan CSF+GSF dan sistem yang hanya menggunakan GSF, maka dilakukan uji-*t*. Uji-*t* yang digunakan adalah uji-*t* berpasangan dua sisi dengan taraf signifikansi 0.05. Nilai *alpha* yang dipilih untuk melakukan uji-*t* ini adalah *alpha* yang menghasilkan akurasi tertinggi untuk tiap domain produk. Agar sistem yang diuji tidak terbiaskan oleh *noise*, maka sistem

yang diuji hanya sistem yang menggunakan kamus kata dalam proses perbandingannya. Proses uji-*t* ini dapat dilihat pada Lampiran 6.

4.4. Pembahasan

Dari hasil uji coba yang dilakukan pada sub bab sebelumnya, dapat dilihat bahwa rule-rule yang didefinisikan pada metode *Modification Double Propagation* (MDP) telah mampu mengekstrak fitur produk dengan tingkat akurasi yang cukup baik (lihat Tabel 4.3). Khusus untuk 2 rule baru yang ditambahkan pada penelitian ini, yakni R5 dan R6, meskipun nilai *coverage* dari kedua rule tersebut relatif kecil (lihat Tabel 4.2), tapi tingkat akurasinya menyamai rule-rule yang lain. Hal ini menunjukkan bahwa dua rule tersebut dapat menjadi solusi untuk mengekstrak fitur-fitur yang jarang dikomentari tapi relevan dengan domain produk yang sedang dianalisa. Bahkan untuk R6, yaitu rule yang menangani relasi *prep_with(adj,noun)*, rule tersebut dapat mencegah sistem untuk mengekstrak fitur yang salah. Sebagai contoh pada kalimat berikut :

I am happy with the screen of this smartphone.

Dalam kalimat tersebut, terdapat kata *adjective* yang langsung bisa dengan mudah diidentifikasi sebagai kata opini, yakni kata *happy*. Dalam dokumen *parsed*, kata ini akan memiliki relasi *nsubj* dengan kata *I*. Oleh karena itu, kata *I* akan dianggap sebagai fitur oleh relasi R2. Padahal dalam kalimat tersebut, fitur yang seharusnya terekstrak adalah *screen*. Struktur kalimat seperti ini dapat menjadi jebakan pada metode *Double Propagation* yang konvensional. Padahal jika diamati lebih lanjut, terdapat relasi lain yang menghubungkan kata *happy* tersebut dengan kata *screen*, yakni relasi *prep_with*. Hal inilah yang ditangani oleh rule R6.

Tabel 4.4 menunjukkan kelebihan lain dari metode *Modification Double Propagation* (MDP) dibandingkan dengan metode DP yang konvensional. Meskipun rata-rata waktu komputasi yang digunakan oleh metode MDP lebih tinggi, tapi selisihnya tidaklah besar. Hanya berkisar antara 0 sampai 10 detik. Sementara rata-rata nilai *precision* dari metode MDP lebih tinggi daripada metode DP. Angka 0,45 bukan berarti bahwa metode MDP gagal mengekstrak sebagian besar fitur relevan. Melainkan jumlah fitur relevan yang terdapat di kamus kata memang jauh lebih sedikit dibandingkan fitur yang berhasil diekstrak secara

keseluruhan. Hal ini dibuktikan dengan pengujian nilai *recall* dari metode MDP yang mencapai 0,81, lebih tinggi dibandingkan metode DP yang hanya mencapai 0,71. Total kemunculan fitur relevan yang berhasil diidentifikasi oleh metode MDP pun lebih tinggi dibandingkan metode DP dengan selisih hampir 100 kali kemunculan. Artinya, jika menggunakan metode DP yang konvensional, maka ada 100 fitur relevan yang muncul di dataset tapi tidak dapat diidentifikasi oleh sistem. Hal ini menunjukkan bahwa penambahan rule pada metode MDP mampu meningkatkan kinerja dari sistem dalam mengekstrak fitur produk.

Nilai CSF yang diusulkan dalam penelitian ini juga terbukti mampu meningkatkan akurasi dari sistem. Hal ini dapat dilihat pada Tabel 4.8 sampai 4.14. Dimana untuk tiap nilai *threshold*, akurasi tertinggi didapatkan pada saat *alpha* tidak sama dengan 1. Seperti dijelaskan pada bab sebelumnya, bahwa nilai *alpha* ini adalah bobot yang diberikan untuk nilai GSF, sedangkan bobot untuk CSF adalah $1 - \alpha$. Artinya, semakin besar nilai *alpha*, semakin kecil pula nilai CSF memberikan pengaruh dalam proses perbandingan. Sehingga saat nilai *alpha* diset sama dengan 1, artinya nilai CSF tidak diperhatikan atau diabaikan, dan sistem hanya menggunakan nilai GSF saja dalam melakukan proses perbandingan. Ketika akurasi tertinggi justru didapatkan saat nilai *alpha* tidak sama dengan 1, artinya nilai CSF ikut berperan dalam proses perbandingan yang dilakukan oleh sistem. Hal ini membuktikan bahwa teorema awal yang diajukan dalam penelitian ini tentang kalimat *comparative* ikut memberikan peran dalam proses perbandingan suatu produk terbukti kebenarannya. Hasil pengujian yang dilakukan dengan menggunakan uji-*t* untuk melihat seberapa signifikan perubahan nilai akurasi pada sistem yang menggunakan CSF jika dibandingkan dengan sistem yang hanya menggunakan GSF juga menunjukkan hasil bahwa perubahannya signifikan.

Dari Tabel 4.8 sampai 4.14 juga dapat dilihat bahwa besarnya nilai akurasi tidak hanya dipengaruhi oleh nilai *alpha*, tetapi juga nilai *threshold*. Besar nilai *threshold* ini akan menentukan berapa banyak fitur yang turut diamati dalam proses perbandingan. Semakin besar nilai *threshold*, maka semakin sedikit jumlah fitur yang diproses selama perbandingan. Demikian pula sebaliknya. Dari keberadaan nilai *threshold* ini pun dapat diamati bahwa semakin banyak fitur yang

turut diamati dalam suatu perbandingan, maka nilai akurasinya cenderung menurun. Hal ini diakibatkan semakin banyaknya fitur-fitur yang tidak relevan (atau fitur relevan tapi tidak penting) turut diperhatikan dalam proses perbandingan. Fitur-fitur ini dapat menjadi *noise* yang dapat membiaskan hasil perbandingan.

Pengujian akurasi yang dilakukan pada domain camera digital dan televisi menunjukkan hasil yang tidak terlalu bagus untuk semua kondisi. Dimana akurasi tertinggi hanya berkisar di range 0,6 sampai 0,8. Hal ini disebabkan ketidakterediaan dataset yang bagus untuk kedua domain tersebut. Seperti dijelaskan pada bab sebelumnya bahwa dataset didapatkan dari situs amazon.com, sedangkan data pakar didapatkan dari situs findthebest.com. Masalah muncul ketika tidak semua produk yang dijual di amazon, memiliki nilai rate di situs findthebest.com. Solusinya adalah membalik alur, yakni melihat produk apa saja yang memiliki rate di findthebest, baru kemudian mencari produk yang dimaksud di situs amazon.com. Akan tetapi, untuk domain tv dan camera digital, range harga dari produk-produk yang memiliki rate di findthebest terlampau besar. Jika untuk domain smartphone range harganya berkisar antara 1 sampai 2 juta, range harga untuk domain tv dan camera digital berkisar antara 3 sampai 8 juta rupiah. Perbedaan range harga yang terlalu jauh inilah yang membuat system kesulitan untuk mendeteksi produk mana yang lebih unggul. Karena independensi antar produk juga terlampau tinggi.

Sementara pengujian akurasi dengan menggunakan 2 skenario yang berbeda, yakni menggunakan kamus kata relevan dan tanpa menggunakan kamus kata, turut memberikan informasi bahwa algoritma pembobotan yang diberikan untuk tiap fitur mampu meminimalisir gangguan (*noise*) yang diakibatkan oleh munculnya fitur yang tidak relevan. Dengan adanya bobot yang diberikan pada tiap fitur, maka secara otomatis, fitur-fitur yang jarang muncul (biasanya bukan fitur yang relevan) akan mendapatkan bobot yang kecil pula. Sehingga tidak akan memberikan pengaruh yang terlalu besar dalam proses perbandingan. Akan tetapi, jika *noise* ini sangat banyak, seperti yang terjadi saat *threshold* kurang dari 30, maka keberadaan *noise* ini juga akan mengganggu dan akan menurunkan akurasi dari system.

Dari hasil pengamatan yang dilakukan pada hasil ujicoba dan pengamatan pada Tabel 4.4 sampai 4.7, dapat disimpulkan bahwa tidak selamanya produk dengan harga yang lebih mahal akan selalu unggul ketika dibandingkan dengan produk yang harganya lebih murah. Baik pakar maupun sistem setuju bahwa tingginya harga suatu produk, tidak menjamin bahwa kualitas barang tersebut juga tinggi. Bahkan adakalanya calon pembeli tidak puas dengan tingginya harga yang ditetapkan oleh produsen. Hal ini ditunjang oleh pengamatan pada 5 fitur yang paling sering dikomentari untuk setiap produk yang terdapat pada Lampiran 2, dimana fitur "price" adalah salah satu fitur yang paling sering dikomentari pada domain kamera. Sehingga, dapat diambil kesimpulan bahwa fitur harga akan ikut menentukan hasil perbandingan hanya jika fitur tersebut dikomentari oleh banyak customer.

Dalam penelitian ini, terdapat beberapa hal yang perlu diperhatikan. Pertama, sistem belum mampu menentukan berapa nilai *threshold* dan *alpha* yang terbaik untuk semua data, karena hal tersebut sangat bergantung pada dataset. Kedua, sistem juga belum mampu menangani fitur-fitur yang ditulis dengan nama yang berbeda atau menggunakan kata ganti, misal kata *thing* dalam penelitian ini dianggap tidak relevan. Meskipun diketahui bahwa kata tersebut biasanya digunakan untuk merujuk pada suatu fitur lain, tapi sistem belum bisa mengetahui fitur mana yang dirujuk oleh kata tersebut. Ketiga, sistem juga belum mampu menangani kalimat opini yang bersifat obyektif, misal kalimat "*there is a scratch on my screen*". Kalimat tersebut sebenarnya merupakan bentuk ungkapan kekecewaan terhadap fitur layar karena munculnya goresan, tapi sistem belum mampu mendeteksi kalimat seperti ini sebagai kalimat opini karena keterbatasan rule yang dibuat. Dimana tidak selamanya struktur kalimat seperti itu mengindikasikan kalimat opini. Seperti kalimat "*There is money in my pocket*" yang bukan merupakan kalimat opini. Kekurangan-kekurangan tersebut akan menjadi topik bahasan yang menarik untuk diteliti pada penelitian berikutnya.

Akan tetapi, terlepas dari kekurangan yang ada, sistem menunjukkan hasil yang lebih baik jika dibandingkan dengan penelitian-penelitian yang ada sebelumnya. Dari Tabel 4.4 dapat dilihat bahwa metode MDP yang diusulkan dalam penelitian kali ini mampu mengalahkan metode DP konvensional yang

diusulkan oleh Qiu dalam beberapa hal saat ekstraksi fitur. Skor CSF yang diusulkan dalam penelitian ini juga terbukti mampu meningkatkan hasil akurasi dari sistem jika digabungkan dengan skor GSF. Dibandingkan dengan metode yang diusulkan oleh Liu yang hanya menggunakan GSF saja, terbukti nilai akurasi yang didapatkan oleh sistem jauh lebih unggul. Bahkan system mampu mengungguli metode yang diusulkan oleh Liu hampir di semua *threshold* yang diujikan.



BAB 5 KESIMPULAN DAN SARAN

BARU
KEMERDEKAAN
DAN SABAN

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari hasil uji coba yang dilakukan pada bab sebelumnya, maka dapat diambil kesimpulan sebagai berikut :

- Penambahan rule dalam metode *Modification Double Propagation* terbukti dapat meningkatkan nilai *precision* sebesar 0,45 dan *recall* sebesar 0,81 daripada metode *Double Propagation* konvensional yang sebelumnya diusulkan oleh Qiu.
- Penambahan skor CSF dapat meningkatkan akurasi proses perbandingan suatu produk hingga mencapai 93% tanpa menggunakan kamus kata relevan dan 96% jika menggunakan kamus kata relevan.
- Nilai akurasi dari sistem sangat bergantung pada nilai *threshold* dan bobot kalimat *comparative (alpha)* yang didefinisikan oleh user. Pada pengujian yang dilakukan pada tiga buah domain, yakni smartphone, camera digital dan televisi, pada umumnya nilai akurasi terbaik didapatkan ketika nilai *threshold* berkisar antara 80 hingga 100. Akan tetapi dari hasil uji coba tersebut, sulit diketahui berapa nilai *alpha* yang akan memberikan akurasi terbaik dikarenakan hasil yang berbeda-beda di tiap domainnya. Sehingga disimpulkan bahwa nilai *alpha* ini akan sangat bergantung pada dataset.

5.2. Saran

Untuk meningkatkan kinerja dari system, maka untuk penelitian selanjutnya ada beberapa hal yang harus dibenahi atau ditangani, diantaranya :

- Mengotomatisasi pencarian nilai *threshold* dan *alpha* yang paling optimal
- Penanganan terhadap kata-kata yang seringkali digunakan sebagai kata ganti untuk merujuk pada fitur tertentu, seperti *thing*, *one*, dan lain-lain
- Penanganan terhadap keberadaan kalimat opini yang berbentuk kalimat objektif

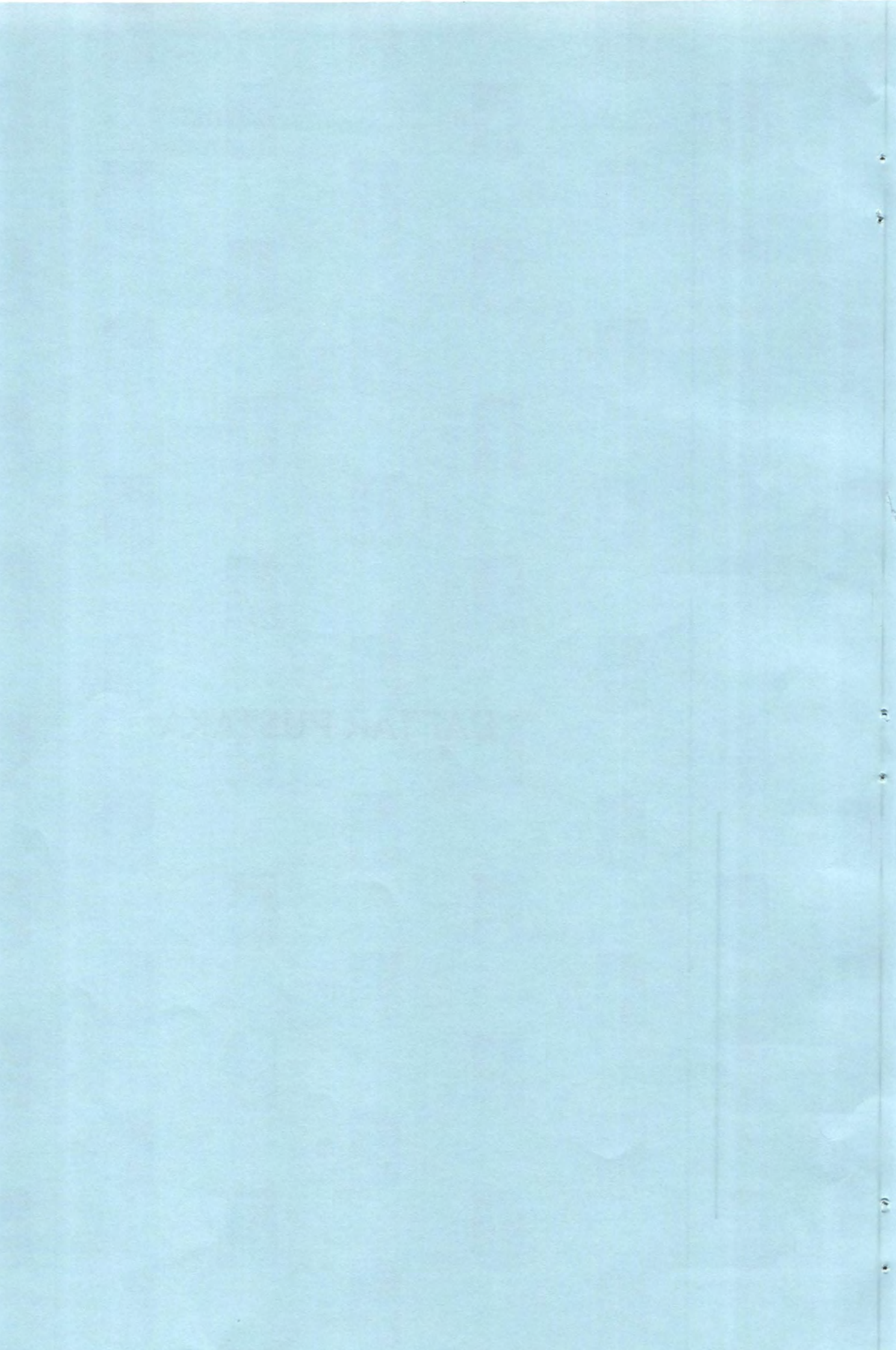


112
112

Halaman ini sengaja dikosongkan

Faint, illegible text covering the majority of the page, likely bleed-through from the reverse side of the document.

DAFTAR PUSTAKA



DAFTAR PUSTAKA

- Boiy, E., Hens, P., Deschacht, K., dan Moens, M. F. (2007) "Automatic sentiment analysis in on-line text." *Proceedings of the 11th International Conference on Electronic Publishing*, hal. 349-360
- Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Ursu, C., Dimitrov, M., ... dan Aswani, N. (2002). Developing language processing components with GATE (a user guide). *University of Sheffield, Sheffield UK*, 5.
- Dave, K., Lawrence, S., dan Pennock, D. M. (2003) "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," *Proceedings of WWW*, hal. 519–528
- De Marneffe, M. C., MacCartney, B., & Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC* (Vol. 6, pp. 449-454)
- Esuli, A., & Sebastiani, F. (2007). Pageranking wordnet synsets: An application to opinion mining. In *Annual meeting-association for computational linguistics* Vol. 45, No. 1, hal. 424
- Ganapathibhotla, M., dan Liu, B. (2008). Mining opinions in comparative sentences. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, hal. 241-248
- Hu, M., dan Liu, B. (2004) "Mining and summarizing customer reviews," *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, hal. 168-177
- Jindal, N., dan Liu, B. (2006). "Identifying comparative sentences in text documents," *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*.
- Jones, B. (1994). Can punctuation help parsing. In *15 th International Conference on Computational Linguistics, Kyoto, Japan*.
- Lan, M., Sung, S., Low, H., Tan, C. (2004). A Comparative Study on Term Weighting Schemes for Text Categorization. In *Proceedings of the 18th international conference on Computational Linguistics*

- Liu, B. (2010). Sentiment analysis: A multi-faceted problem. *the IEEE Intelligent Systems*.
- Liu, B., Hu, M., dan Cheng, J. (2005). Opinion observer: analyzing and comparing opinions on the Web. In *Proceedings of the 14th international conference on World Wide Web*. ACM, hal. 342-351
- Pang, B., dan Lee, L. (2008) "Opinion mining and sentiment analysis." *Foundations and Trends in Information Retrieval* 2(1-2), hal. 1-135.
- Tan, S., Cheng, X., Wang, Y., dan Xu, H. (2009) "Adapting naive bayes to domain adaptation for sentiment analysis." *Advances in Information Retrieval*, hal. 337-349
- Tesnière, L., & Fourquet, J. (1959). *Eléments de syntaxe structurale* (Vol. 1965). Paris: Klincksieck.
- Turney P. D. (2002), "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews," *Proceedings of the Association for Computational Linguistics (ACL)*, hal. 417-424
- Qiu, G., Liu, B., Bu, J., dan Chen, C. (2011). Opinion word expansion and target extraction through double propagation. *Computational linguistics*, 37(1), hal. 9-27
- Wiebe, J., Wilson, T., Bruce, R., Bell, M., & Martin, M. (2004). Learning subjective language. *Computational linguistics*, 30(3), 277-308.
- Yi, J., Nasukawa, T., Bunescu, R., dan Niblack, W. (2003). Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, hal. 427-434
- Zhang, K., Cheng, Y., Liao, W. K., dan Choudhary, A. (2011). Mining millions of reviews: a technique to rank products based on importance of reviews. In *10th international conference on entertainment computing*.
- Zhang, L., Liu, B., Lim, S. H., dan O'Brien-Strain, E. (2010). Extracting and ranking product features in opinion documents. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, hal. 1462-1470

LAMPIRAN 1

Lembar Kuisiener Penentuan Fitur Relevan

IDENTITAS RESPONDEN

Nama :
 Jenis Kelamin : Pria / Wanita **(coret yang tidak perlu)*
 Usia :
 Pendidikan terakhir : SD / SMP / SMA / S1 / S2 / S3 **(coret yang tidak perlu)*

KUISIONER PEMILIHAN FITUR PRODUK YANG RELEVAN

Setiap produk memiliki beberapa fitur (komponen) yang seringkali dikomentari oleh customer. Misal untuk produk smartphone, customer seringkali mengomentari fitur screen, size, camera, keypad, price, dll. Sedangkan untuk produk Komputer, biasanya yang sering dikomentari selain harganya adalah ukurannya, processornya, memorinya, VGanya, dan sebagainya. Fitur-fitur tersebut dikategorikan sebagai fitur yang relevan untuk produk tersebut.

Berikut ini adalah daftar kata yang berhasil diekstrak oleh system otomatisasi pengekstrak fitur produk. Anda diharapkan memberi tanda “R” pada kolom “R” untuk setiap fitur yang Anda anggap merupakan fitur yang relevan untuk 3 buah produk, yaitu televisi, smartphone, dan camera digital.

TELEVISI

FITUR	R	FITUR	R	FITUR	R	FITUR	R
Quality		unit		3d		Way	
picture quality		store		Area		Model	
Picture		player		tv features		Service	
Light		item		Voice		Brand	
Year		value		Website		Lcd	
Issue		version		Video		Use	
Thing		money		sony tv		Deal	
Set		menu		inch tv		Setup	
Screen		job		Component		Netflix	
Price		rate		Monitor		Sony	
Time		system		Cable		Clarity	
Product		performance		Reason		Everything	
Purchase		line		Everyone		Samsung	
Sound		something		Star		screen tv	
Choice		installation		Tech		Audio	
Movie		highly		Panel		Size	
Amazon		work		Network		Complaint	
Experience		box		Site		Bar	
Shipping		week		Difference		Condition	
One		good		Regret		Setting	

Day	control	Anything	theater
Buy	connection	Pro	mention
Review	output	Warranty	replacement tv
Color	reviewer	Pixel	room
Month	option	Remote	help
Display	amount	Image	support
Viera	app	Idea	improvement
Problem	speaker	Gift	while
Hdtv	right	Volume	function
Feature	firmware	Internet	addition
Signal	rating	Button	black
Bargain	easy	samsung tv	programming
Power	input	Dvd	order

Apakah ada fitur yang menurut Anda relevan untuk produk Televisi tapi tidak ada dalam list ini? Jika ada, sebutkan !

.....

SMARTPHONE

FITUR	R	FITUR	R	FITUR	R	FITUR	R
smartphone		Network		Charge		memory	
product		Model		cell phone		market	
problem		Mobile		Case		manufacturer	
phone		Keyboard		Card		htc	
version		Item		Amazon		hour	
Time		Iphone		Audio		have	
Thing		Idea		Zoom		gonna	
Price		Design		Resolution		file	
One		Complaint		led flash		fan	
Device		Cell		Light		experience	
Way		Buy		Hardware		display	
Size		Button		Headset		prize	
Screen		Warranty		Headphone		photo	
Quality		Volume		Game		packaging	
Day		Update		Application		other	
Battery		Delivery		touch screen		option	
App		customer service		Tool		music	
Work		Curve		Text		money	
Purchase		Contact		System		month	
Issue		Connection		Sound		cellphone	
everything		Company		sim card		android	
Deal		Shipping		Choice		upgrade	
condition		Setting		Carrier		stuff	
User		Research		Camera		speed	
People		Reason		Brand		software	
Look		Provider		Box		service	
Feature		Reception		battery life		seller	
Review		Use		Year			
Picture		Performance		Processor			

Apakah ada fitur yang menurut Anda relevan untuk produk Smartphone tapi tidak ada dalam list ini? Jika ada, sebutkan !

.....

CAMERA DIGITAL

FITUR	R	FITUR	R	FITUR	R	FITUR	R
Quality		shot mode		Range		Model	
Year		memory card		Function		Work	
Purchase		Stabilizer		Slr		Friend	
Picture		Charge		Level		Side	
Feature		Shutter		Stuff		complaint	
Thing		Lcd		Digital		Shoot	
Price		Exposure		Detail		Owner	
One		Balance		Card		Pic	
Time		picture quality		Upgrade		Control	
Review		Problem		Effect		condition	
Point		battery life		Clarity		Canon	
Product		issue		lcd screen		shoot cameras	
Shot		screen		Photograph		User	
Zoom		result		Panasonic		Button	
Job		service		View		Easy	
Photo		flash		Body		Part	
Image		brand		Experience		Amount	
performance		resolution		Release		dslr camera	
Light		use		menu system		Lense	
photographer		speed		Drawback		Version	
Video		month		Angle		Nikon	
reason		dslr		battery charger		Right	
Setting		Place		Charger		exception	
image quality		Hand		slr camera		description	
deal		Mode		photography		Luck	
color		Menu		Out		Thought	
value		quality camera		File		improvement	
option		Everything		Company		telephoto	
choice		Start		Word		Grip	
size		Fun		Manual		While	
item		Trip		Shooting		First	
weight		Functionality		Software		Battery	
kodak		Memory		Fast		Type	
setup		light conditions		Auto		Lighting	

Apakah ada fitur yang menurut Anda relevan untuk produk camera digital tapi tidak ada dalam list ini? Jika ada, sebutkan !

.....

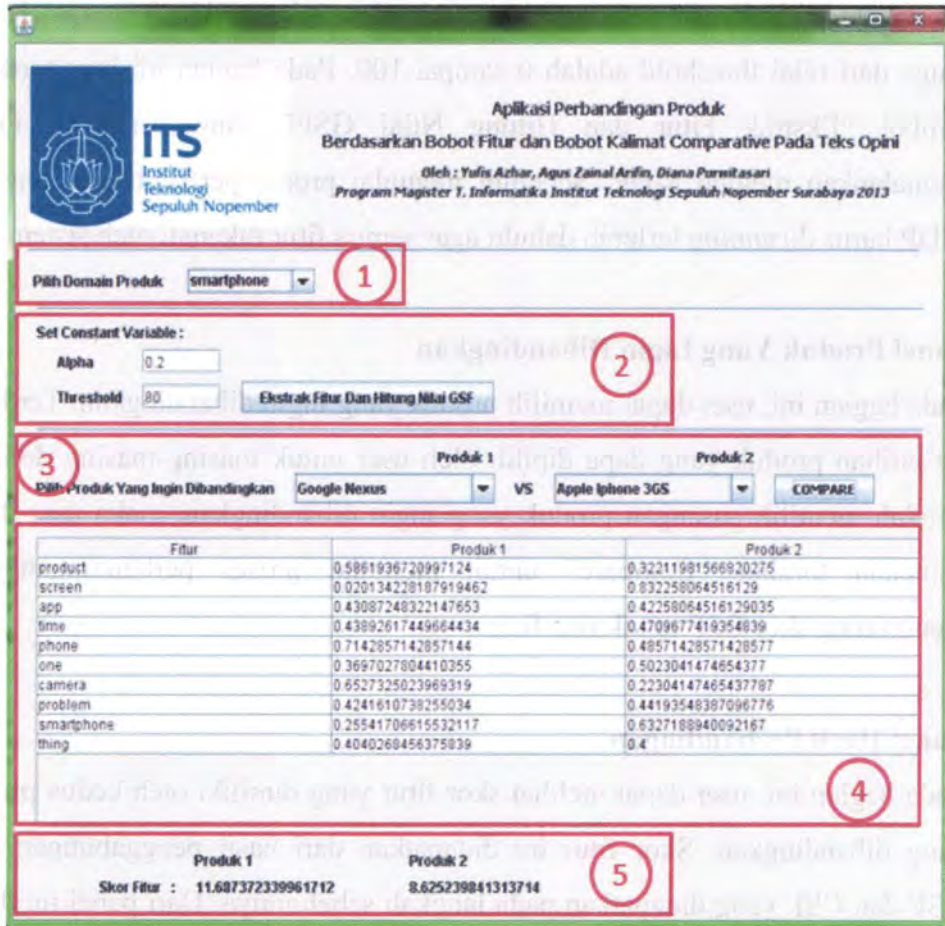
Halaman ini sengaja dikosongkan

No	Nama	Alamat	Telepon	Waktu
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

LAMPIRAN 3

Antarmuka Aplikasi

Dalam Lampiran 3 ini akan diperlihatkan tampilan antarmuka (*interface*) dari aplikasi perbandingan produk yang dibuat dalam penelitian ini.



Aplikasi Perbandingan Produk
Berdasarkan Bobot Fitur dan Bobot Kalimat Comparative Pada Teks Opini
Oleh : Yufis Azhar, Agus Zainal Arifin, Diana Purmitasari
Program Magister 1. Informatika Institut Teknologi Sepuluh Nopember Surabaya 2013

Pilih Domain Produk: 1

Set Constant Variable:
Alpha: 2
Threshold: Ekstrak Fitur Dan Hitung Nilai GSF

Dih. Produk Yang Ingin Dibandingkan: VS COMPARE 3

Fitur	Produk 1	Produk 2
product	0.5861936720997124	0.32211981566820275
screen	0.020134228187919462	0.832258064516129
app	0.43087248322147653	0.42258064516129035
time	0.43892617449664434	0.4709677419354839
phone	0.7142857142857144	0.48571428571428577
one	0.3697027804410355	0.5023041474654377
camera	0.6527325023969319	0.22304147465437787
problem	0.4241610738255034	0.44193548387096776
smartphone	0.25541706615532117	0.6327188940092167
thing	0.4040260456375839	0.4

Produk 1 Skor Fitur : 11.68737239961712 5
Produk 2 Skor Fitur : 8.625239841313714

Gambar 1. Tampilan Antarmuka Aplikasi

Pada gambar antarmuka, terdapat 5 panel yang digunakan untuk berinteraksi dengan user. Adapun penjelasan dari masing-masing panel tersebut adalah sebagai berikut :

1. Panel Domain Produk

Pada bagian ini, user dapat memilih domain produk yang ingin dibandingkan. Terdapat 3 pilihan domain yang dapat dipilih oleh user, yaitu *smartphone*, *camera*, dan *tv*.

2. Panel *Constant Variable*

Pada bagian ini, user dapat menentukan nilai *alpha* dan nilai *threshold* yang digunakan selama proses perbandingan. Kedua nilai ini awalnya kosong, dan akan berisi nilai yang direkomendasikan oleh sistem ketika user memilih salah satu domain pada panel 1. Akan tetapi user tetap dapat mengubah kedua nilai ini sesuka hati, dengan catatan range dari nilai *alpha* adalah 0 sampai 1, dan range dari nilai *threshold* adalah 0 sampai 100. Pada bagian ini juga terdapat tombol “Ekstrak Fitur dan Hitung Nilai GSF” yang berfungsi untuk menjalankan metode MDP. Sebelum memulai proses perbandingan, metode MDP harus *dirunning* terlebih dahulu agar semua fitur dikenali oleh sistem.

3. Panel Produk Yang Ingin Dibandingkan

Pada bagian ini, user dapat memilih produk yang ingin dibandingkan. Terdapat 10 pilihan produk yang dapat dipilih oleh user untuk masing-masing domain. Setelah memilih pasangan produk yang ingin dibandingkan, maka user dapat menekan tombol “Compare” untuk memulai proses perbandingan dan menghitung skor CSF untuk tiap fitur.

4. Panel Hasil Perbandingan

Pada bagian ini, user dapat melihat skor fitur yang dimiliki oleh kedua produk yang dibandingkan. Skor fitur ini didapatkan dari hasil penggabungan skor GSF dan CSF yang didapatkan pada langkah sebelumnya. Dari panel ini dapat terlihat produk mana yang lebih unggul berdasarkan fitur tertentu.

5. Panel Skor Rekomendasi Produk

Dalam bagian ini, skor rekomendasi dari tiap produk ditampilkan. Skor ini dihitung dengan melihat seberapa banyak dan seberapa penting fitur unggul yang dimiliki oleh suatu produk. Dari panel ini, produk yang memiliki skor lebih besar berarti produk itulah yang lebih diunggulkan oleh sistem.

A. Uji-*t* Untuk Domain Smartphone

Tabel 1. Perbandingan Nilai Akurasi Untuk Domain Smartphone Antara GSF ($\alpha=1$) dan GSF+CSF ($\alpha=0,2$) Dengan Menggunakan Kamus Kata

Threshold	GSF	GSF+CSF	Selisih
10	0,42	0,58	-0,1600
20	0,56	0,64	-0,0800
30	0,6	0,71	-0,1100
40	0,69	0,84	-0,1500
50	0,69	0,82	-0,1300
60	0,67	0,78	-0,1100
70	0,78	0,93	-0,1500
80	0,78	0,93	-0,1500
90	0,78	0,96	-0,1800
100	0,71	0,96	-0,2500

Tabel 2. Hasil Uji-*t* Pada Perbandingan Nilai Akurasi Untuk Domain Smartphone Antara GSF ($\alpha=1$) dan GSF+CSF ($\alpha=0,2$) Dengan Menggunakan Kamus Kata

Uji- <i>t</i> Berpasangan	
jumlah data	10
df (degree freedom)	9
Taraf keyakinan (α)	0,05
t Tabel(α,df)	2,262
Mean GSF	0,668
Mean GSF+CSF	0,815
selisih Mean	-0,147
Standar Deviasi Selisih	0,046
T hitung	-10,010
Jawaban Hipotesis	H0 Ditolak
Perbedaan	Ada Perbedaan Signifikan

Dapat dilihat pada Tabel 2, hasil uji-*t* menunjukkan bahwa *t* Tabel lebih besar dari *T* hitung, maka disimpulkan terdapat perbedaan yang signifikan antara hasil perbandingan nilai akurasi yang menggunakan metode pembobotan dengan menggunakan GSF+CSF dibandingkan dengan GSF. Nilai *mean* dari metode GSF+CSF juga lebih besar dibandingkan metode GSF. Oleh karena

itu, untuk domain smartphone, dapat disimpulkan bahwa penambahan skor CSF dapat meningkatkan hasil akurasi dari sistem.

B. Uji-t Untuk Domain Camera Digital

Tabel 3. Perbandingan Nilai Akurasi Untuk Domain Camera Antara GSF ($\alpha=1$) dan GSF+CSF ($\alpha=0,6$) Dengan Menggunakan Kamus Kata

Threshold	GSF	GSF+CSF	Selisih
10	0,25	0,31	-0,0600
20	0,21	0,27	-0,0600
30	0,32	0,38	-0,0600
40	0,43	0,44	-0,0100
50	0,4	0,44	-0,0400
60	0,42	0,42	0,0000
70	0,56	0,56	0,0000
80	0,51	0,56	-0,0500
90	0,56	0,64	-0,0800
100	0,56	0,64	-0,0800

Tabel 4. Hasil Uji-t Pada Perbandingan Nilai Akurasi Untuk Domain Smartphone Antara GSF ($\alpha=1$) dan GSF+CSF ($\alpha=0,6$) Dengan Menggunakan Kamus Kata

Uji-t Berpasangan	
jumlah data	10
df (degree freedom)	9
Taraf keyakinan (α)	0,05
t Tabel(α,df)	2,262
Mean GSF	0,422
Mean GSF+CSF	0,466
selisih Mean	-0,044
Standar Deviasi Selisih	0,031
T hitung	-4,544
Jawaban Hipotesis	H0 Ditolak
Perbedaan	Ada Perbedaan Signifikan

Dapat dilihat pada Tabel 4, hasil uji-t menunjukkan bahwa t Tabel lebih besar dari T hitung, maka disimpulkan terdapat perbedaan yang signifikan antara hasil perbandingan nilai akurasi yang menggunakan metode

LAMPIRAN 4

Uji-t Berpasangan Dua Sisi

Uji-*t* berpasangan (*Paired-Samples t Test*) digunakan untuk membandingkan selisih dua *mean* dari dua sampel yang berpasangan dengan asumsi data berdistribusi normal. Pada penelitian ini uji-*t* berpasangan yang digunakan adalah uji-*t* berpasangan dua sisi (*two-sided* atau *two-tailed test*). Uji-*t* berpasangan dua sisi digunakan untuk membandingkan dua buah distribusi data apakah memiliki perbedaan yang signifikan atau tidak berbeda signifikan. Pada penelitian ini data-data yang akan menjalani uji-*t* telah terdistribusi normal menurut pengujian *Kolmogorov Smirnov*.

PERUMUSAN HIPOTESIS :

Secara Matematis

$$h_0 : \mu_{data1} = \mu_{data2} \text{ atau } h_0 : \mu_D = 0$$

$$h_1 : \mu_{data1} \neq \mu_{data2} \text{ atau } h_1 : \mu_D \neq 0$$

dimana:

μ_{data1} adalah rata-rata dari distribusi *data1*

μ_{data2} adalah rata-rata dari distribusi *data2*

$$\mu_D = \mu_{data1} - \mu_{data2}$$

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:

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

d_i = selisih pasangan data yaitu $d_i = data1_i - data2_i$, $i = 1, 2, 3, \dots, n$ (jumlah data)

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

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_0 ditolak maka ada perbedaan signifikan.
2. Jika $-t_{hitung} < -t_{tabel}$ atau $t_{hitung} < t_{tabel}$ atau $|t_{hitung}| < |t_{tabel}|$, maka keputusan:
 h_0 diterima maka tidak ada perbedaan signifikan.

Adapun langkah-langkah pada uji-t yaitu:

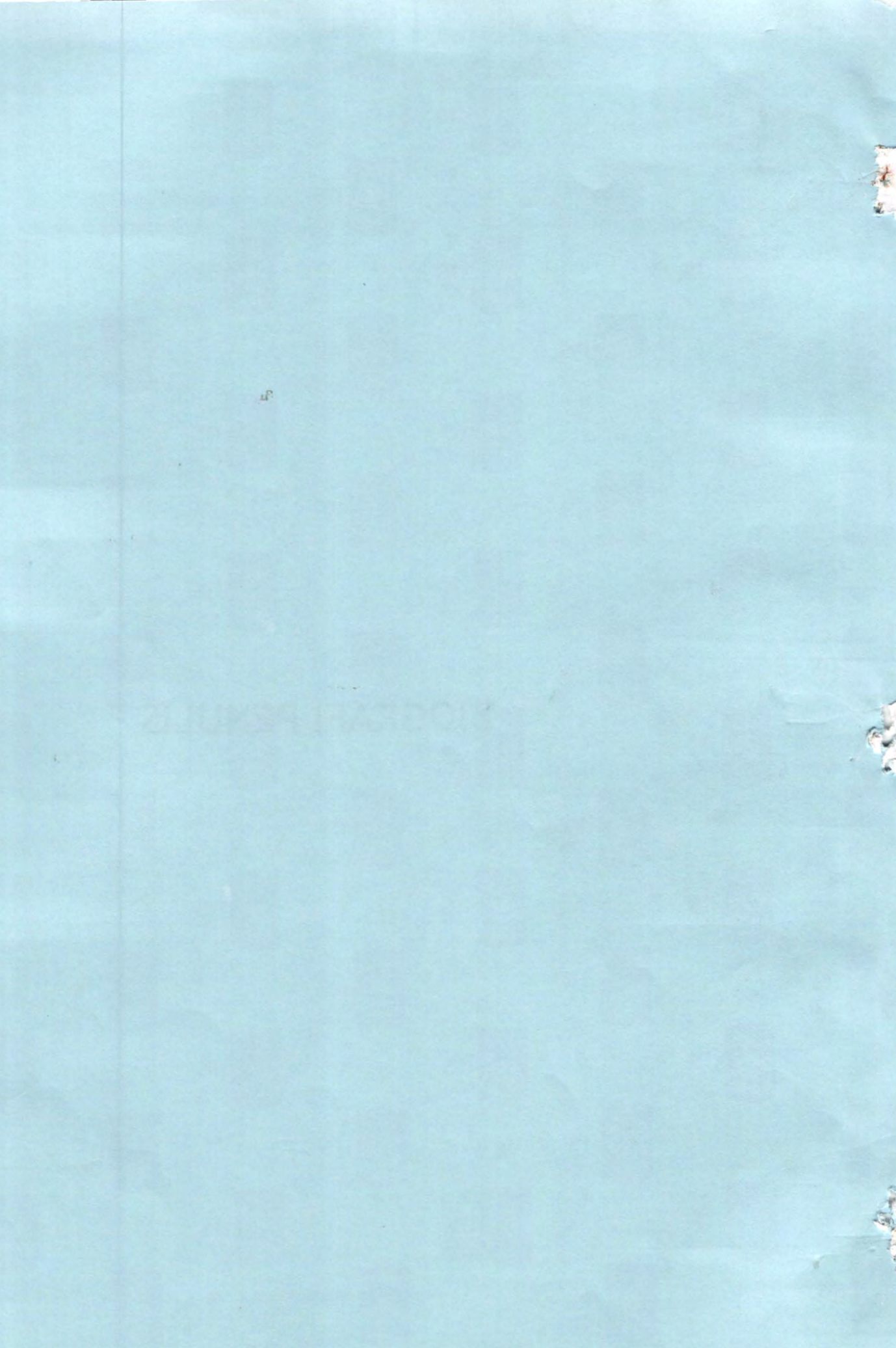
1. Masukkan tiap nilai *data1* dan *data2* ke dalam tabel sesuai dengan urutan data yang sebenarnya. Kemudian cari selisih setiap pasangan data untuk masing-masing kolom.
2. Hitung nilai \bar{d} dan s_d untuk mendapatkan nilai t_{hitung} .
3. Setelah mendapatkan nilai t_{hitung} , maka bandingkan dengan nilai t_{tabel} yang dihasilkan dengan membaca Tabel *t* pada Lampiran 5. $t_{tabel} = t_{\alpha, df}$. Variabel α adalah nilai taraf signifikansi (pada penelitian ini $\alpha = 0.05$) dan df adalah derajat kebebasan (*degree freedom*) dihitung dengan rumus $df = n - 1$.
4. Tarik kesimpulan sesuai dengan aturan pada **Pengambilan Keputusan Hipotesis Uji-t berpasangan**. Jika h_0 ditolak maka ada perbedaan signifikan pada distribusi data dari kedua sampel sedangkan jika menerima h_0 maka tidak ada perbedaan signifikan pada kedua distribusi data.

Pada penelitian ini digunakan nilai α (taraf signifikansi) = 0.05 dan derajat bebas = $n - 1 = 10$. Sehingga nilai t_{tabel} berdasarkan nilai yang diambil dari Tabel *t* yaitu $t_{0.05, 10} = 0.228$.

Berikut adalah hasil uji *t* berpasangan dua sisi terhadap 3 buah pasangan matriks akurasi hasil perbandingan pada setiap domain produk dengan menggunakan pembobotan GSF dan GSF+CSF, dimana:

$h_0: \mu_{data1} = \mu_{data2}$ atau $h_0: \mu_D = 0$ (tidak ada perbedaan signifikan)

$h_1: \mu_{data1} \neq \mu_{data2}$ atau $h_1: \mu_D \neq 0$ (terdapat perbedaan signifikan)



BIOGRAFI PENULIS

Dapat dilihat pada Tabel 6, hasil uji-t hasil uji-t menunjukkan bahwa t Tabel lebih besar dari T hitung, maka disimpulkan terdapat perbedaan yang signifikan antara hasil perbandingan nilai akurasi yang menggunakan metode pembobotan dengan menggunakan GSF+CSF dibandingkan dengan GSF. Nilai *mean* dari metode GSF+CSF juga lebih besar dibandingkan metode GSF. Oleh karena itu, untuk domain televisi, dapat disimpulkan bahwa penambahan skor CSF dapat meningkatkan hasil akurasi dari sistem.

Metode	4-7-1971	1971	1972
GSF	0,00	0,00	0,00
GSF+CSF	0,00	0,00	0,00
GSF	0,00	0,00	0,00
GSF+CSF	0,00	0,00	0,00
GSF	0,00	0,00	0,00
GSF+CSF	0,00	0,00	0,00
GSF	0,00	0,00	0,00
GSF+CSF	0,00	0,00	0,00
GSF	0,00	0,00	0,00
GSF+CSF	0,00	0,00	0,00

Metode	4-7-1971	1971	1972
GSF	0,00	0,00	0,00
GSF+CSF	0,00	0,00	0,00
GSF	0,00	0,00	0,00
GSF+CSF	0,00	0,00	0,00
GSF	0,00	0,00	0,00
GSF+CSF	0,00	0,00	0,00
GSF	0,00	0,00	0,00
GSF+CSF	0,00	0,00	0,00
GSF	0,00	0,00	0,00
GSF+CSF	0,00	0,00	0,00

pembobotan dengan menggunakan GSF+CSF dibandingkan dengan GSF. Nilai *mean* dari metode GSF+CSF juga lebih besar dibandingkan metode GSF. Karena itu, untuk domain camera digital, dapat disimpulkan bahwa penambahan skor CSF dapat meningkatkan hasil akurasi dari sistem.

C. Uji-*t* Untuk Domain Televisi

Tabel 5. Perbandingan Nilai Akurasi Untuk Domain Camera Antara GSF ($\alpha=1$) dan GSF+CSF ($\alpha=0,2$) Dengan Menggunakan Kamus Kata

Threshold	GSF	GSF+CSF	Selisih
10	0,48	0,56	-0,0800
20	0,48	0,53	-0,0500
30	0,49	0,53	-0,0400
40	0,49	0,56	-0,0700
50	0,49	0,53	-0,0400
60	0,53	0,64	-0,1100
70	0,56	0,69	-0,1300
80	0,56	0,6	-0,0400
90	0,56	0,78	-0,2200
100	0,58	0,8	-0,2200

Tabel 6. Hasil Uji-*t* Pada Perbandingan Nilai Akurasi Untuk Domain Smartphone Antara GSF ($\alpha=1$) dan GSF+CSF ($\alpha=0,2$) Dengan Menggunakan Kamus Kata

Uji- <i>t</i> Berpasangan	
jumlah data	10
df (degree freedom)	9
Taraf keyakinan (α)	0,05
t Tabel(α,df)	2,262
Mean GSF	0,522
Mean GSF+CSF	0,622
selisih Mean	-0,100
Standar Deviasi Selisih	0,070
T hitung	-4,502
Jawaban Hipotesis	H0 Ditolak
Perbedaan	Ada Perbedaan Signifikan

BIOGRAFI PENULIS



Yufis Azhar adalah anak ketiga dari tiga bersaudara. Penulis dilahirkan dalam lingkungan keluarga sederhana dari pasangan Abdul Wahab dan Nur Hidayati. Penulis memulai pendidikan dasar di MI Bahauddin Taman Sidoarjo, dan melanjutkan ke jenjang sekolah pertama di SMP Muhammadiyah 4 Surabaya. Pada tahun 2002 penulis melanjutkan ke jenjang sekolah menengah atas di SMA Muhammadiyah 3 Surabaya dan dilanjutkan ke jenjang sarjana pada tahun 2005 di Jurusan Teknik Informatika Universitas Muhammadiyah Malang dimana penulis merupakan salah satu dari mahasiswa angkata pertama di jurusan ini. Pada tahun 2011, penulis melanjutkan ke jenjang S2 di Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya. Pada jenjang ini penulis mengambil bidang keahlian komputasi cerdas dan visualisasi. Penulis saat ini masih tercatat dan aktif sebagai salah satu staff pengajar di Jurusan Teknik Informatika Universitas Muhammadiyah Malang sejak tahun 2009 dengan bidang keahlian Kecerdasan Buatan dan Information Retrieval.

Email : yufis05@gmail.com, yufis.azhar@yahoo.com, yufis@umm.ac.id

BIOGRAFI PENULIS

Yulis Axhan adalah anak ketiga dari tiga bersaudara. Penulis tertarik dalam lingkungan keluarga sederhana dan program *World Wide Web* dan lain-lain. Hingga Penulis menulis *Perjalanan ke MIM* (Balutidin Tambo Siboga dan sebagainya) ke jenjang sekolah pertama di SMP Muhammadiyah 4 Surabaya pada tahun 2002 penulis melanjutkan ke jenjang sekolah menengah atas di SMA Muhammadiyah 3 Surabaya dan melanjutkan ke jenjang sarjana pada tahun 2005 di Jurusan Teknik Informatika Universitas Muhammadiyah Malang dimana penulis merupakan salah satu dari mahasiswa angkatan pertama di jurusan ini. Pada tahun 2011 penulis melanjutkan ke jenjang S2 di Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya. Pada jenjang ini penulis mengabdikan bidang keahliannya cerdas dan visualisasi. Penulis saat ini masih terencana dan aktif sebagai salah satu staf pengajar di Jurusan Teknik Informatika Universitas Muhammadiyah Malang sejak tahun 2009 dengan bidang keahlian Keahlian Binaan dan Penelitian.



Retrieved

from <http://www.iaesjournal.com/issue/index.php/iaesjournal/article/view/121>